



Discovering Research Hypotheses in Social Science Using Knowledge Graph Embeddings

Rosaline de Haan¹, Ilaria Tiddi^{2(✉)}, and Wouter Beek¹

¹ Triply, Amsterdam, The Netherlands
{rosaline.de.haan,wouter}@triply.cc

² Vrije Universiteit Amsterdam, Amsterdam, The Netherlands
i.tiddi@vu.nl

Abstract. In an era of ever-increasing scientific publications available, scientists struggle to keep pace with the literature, interpret research results and identify new research hypotheses to falsify. This is particularly in fields such as the social sciences, where automated support for scientific discovery is still widely unavailable and unimplemented. In this work, we introduce an automated system that supports social scientists in identifying new research hypotheses. With the idea that knowledge graphs help modeling domain-specific information, and that machine learning can be used to identify the most relevant facts therein, we frame the problem of hypothesis discovery as a link prediction task, where the ComplEx model is used to predict new relationships between entities of a knowledge graph representing scientific papers and their experimental details. The final output consists in fully formulated hypotheses including the newly discovered triples (hypothesis statement), along with supporting statements from the knowledge graph (hypothesis evidence and hypothesis history). A quantitative and qualitative evaluation is carried using experts in the field. Encouraging results show that a simple combination of machine learning and knowledge graph methods can serve as a basis for automated scientific discovery.

Keywords: Scientific discovery · Knowledge graphs · Link prediction · Social science

1 Introduction

Scientific research usually starts with asking a question, followed by doing background research, and then formulating a testable hypothesis. Doing background research to properly substantiate a hypothesis can be a difficult and time-consuming task for scientists. It is estimated that over 3 million scientific articles are published annually, a number that keeps growing of 4% each year [25]. The fast rate at which new publications appear, as well as the inefficient way in which scientific information is communicated (e.g. PDF documents), calls for

more efficient data analysis and synthesis, in a way that scientists formulating new research hypotheses can be supported rather than overloaded.

The task of significantly speeding up the steps in the scientific process is generally called automated scientific discovery [15]. The latest years have seen Artificial Intelligence approaches for automated scientific discovery in various scientific fields, either relying on symbolic knowledge representation or machine-driven methods. Knowledge graphs such as the Gene Ontology¹ and the ontology collection of the Open Biological and Biomedical Ontology Foundry² have been used to encode domain-specific information, such as representing biological systems from the molecular to the organism level. Machine Learning and particularly link prediction methods, that help predicting which missing edges in a graph are most likely to exist, have also been used, e.g. to support medical scientists by showing them new associations between drugs and diseases [15, 19].

There is currently not much automated support for social scientists when it comes to getting new insights from scientific information. This is partly due to the more qualitative and uncertain nature of social science data, making it hard to represent, and consequently less machine-interpretable [3]. One effort in this direction is the COoperation Databank (CODA), where an international team of social scientists published a structured, open-access repository of research on human cooperation using social dilemmas. The dataset represents about 3,000 research publications with their experimental settings, variables of observation, and quantitative results. Given the large amount of structured information available, and the success of predictive methods seen in other disciplines, it is natural to think that a hybrid method could be designed, to automatically suggest social scientists new hypotheses to be tested.

Here, we study the problem of automatic hypothesis discovery in the field of social sciences. Following approaches in the biomedical field, we propose to frame our problem as a link prediction task, and particularly to exploit the structured representation of the domain to learn research hypotheses in the form of unseen triples over a knowledge graph describing research papers and their experimental settings. Using knowledge graph embeddings, we predict the likelihood of new possible relationships between entities, consisting in the variables studied social science research. These relationships are then used to provide the experts with new research hypotheses structured in a *statement* (the newly predicted associations), *evidence* and *history* (both triples existing in the graph). We quantitatively and qualitatively assess this approach using experts in the field, which helps us evaluating the accuracy and meaningfulness of the discovered hypotheses. Our novelty is not a the prediction algorithm for automated hypothesis discovery, but rather the hybrid method based on link prediction over knowledge graph data. More specifically, we show: (i) how a thorough structured representation of scientific knowledge helps the automatic discovery of research hypotheses, (ii) how our hybrid method can support experts in formulating new research hypotheses and (iii) a practical application in the field of social science.

¹ <http://geneontology.org/>.

² <http://www.obofoundry.org/>.

2 Related Work

Our work relates to three areas, namely (i) existing methods for representing and mining scientific knowledge, (ii) approaches for automated hypothesis discovery in science and (iii) knowledge graph embedding methods and applications.

Representing and Mining Scientific Knowledge. Several methods have been developed to represent scientific knowledge and foster interoperability and reproducibility. Micro- and nanopublications [4,8] have been introduced in the last decade as standardised formats for the publication of minimal scientific claims, i.e. minipublications. Such models allow to describe evidence and nuanced scientific assertions expressing a relationship between two predicates (e.g. a gene relates to a disease), together with provenance information describing both the methods used to derive the assertion and publication metadata. The DISK hypothesis ontology [7] was introduced to capture the evolution of research hypotheses in the neuroscience field, including the provenance and revisions. More precisely, a DISK hypothesis consists of structured assertions (hypothesis statement), some numerical confidence level (hypothesis qualifier), the information of the analysis that were carried out (hypothesis evidence), and prior hypotheses revised to generate the current one (hypothesis history). In the field of medical science, the different elements to be included in a hypothesis can be described with the PICO ontology³, describing Patients, the Condition or disease of interest and its alternative (Intervention), and the Outcome of the study.

Repositories for storing scientific publications at large scale in the form of knowledge graphs include both domain-specific initiatives (e.g. the AI-KG [5] for Computer Science and the Cooperation Databank [22] for the social sciences), and domain-independent projects such as the Open Research Knowledge Graph (ORKG) project⁴. These initiatives focus on representing research outputs in terms of their content, i.e. describing approach, evaluation methods, results etc., rather than publication context such as year, authors and publication venues. This type of novel representations allows to automatise not only the search for new research, but also to compare it at large scale.

Some work has focused on developing systems that aid with mining claims in the existing literature. The AKminer (Academic Knowledge Miner) system [9] was introduced to automatically mine useful concepts and relationships from scientific literature and visually present them in the form of a knowledge graph. Similarly, [17] uses text-mining to automatically extract claims and contributions from scientific literature and enrich them through entity linking methods. Supervised distant learning was used by [14,24] to extract PICO sentences from clinical trial reports and support evidence-based medicine.

Machine-Supported Hypothesis Discovery. Automated hypothesis discovery using intelligent systems has been interest of study for a long time. Earliest work include the ARROWSMITH discovery support system [21] to help scientists in finding complementary literature for their studies and formulate a

³ <https://linkeddata.cochrane.org/pico-ontology>.

⁴ <https://www.orkg.org/orkg/>.

testable hypothesis based on the two sets, and the work of [1], which used various machine learning techniques to discover patterns, co-occurrences and correlations in biological data. These approaches inspired the work of [20], which relies on a scientific text collection to discover hypotheses, via Medical Subject Headings (MeSH)-term based text-mining.

Biomedical literature was also used by [10] to develop a link discovery method based on classification, where concepts are learnt and used as a basis for hypothesis generation. An Inductive Matrix Completion method was presented by [12], where the discovered gene-disease associations were supported by different types of evidence learnt as latent factors. The Knowledge Integration Toolkit (KnIT) [11] used methods such as matrix factorization and graph diffusion to reason over a network of scientific publications in the biomedical field to generate new and testable hypotheses. The work of [15] shows how scientific insights can be generated using machine support also in the field of astronomy and geosciences. Their model allows to create multiple variants of hypothesised phenomena and their corresponding physical properties; these are matched in the existing empirical data, and scientists can both refine them and use them to justify a stated research hypothesis. The DISK ontology was also used in the field of neuroscience for automated hypothesis assessment [6].

Knowledge Graph Embeddings for Link Prediction. Machine learning methods for knowledge graph completion (or link prediction) use inductive techniques, mostly based on knowledge graph embeddings or rule/axiom mining, to locally and logically predict the likelihood of certain link between two nodes to exist [13]. Currently, the tensor decomposition ComplEx method [23] has proven to be the most stable in terms of performance and scalability [2]. Link prediction methods have been previously used for hypothesis discovery. Authors of [16] first create a knowledge graph from biomedical data and then convert it to a lower dimensional space using graph embeddings. The learnt embeddings are then used to train a recurrent neural network model to predict new drug therapies against diseases. A similar approach is the one of [19] to generate hypotheses on re-purposing drugs for rare diseases; the method relies on graph embeddings learnt over a large knowledge graph including information from the literature of pharmacology, genetics and pathology.

3 Background and Motivating Scenario

The COoperation DATAbank Knowledge Graph. The COoperation DATAbank consists in ~3,000 studies from the social and behavioural sciences published in 3 languages and annotated with more than 300 cooperation-related features, including characteristics of the sample participating in the study (e.g. sample size, average age of sample, percentage of males, country of participants), characteristics of the experimental paradigm (structure of the social dilemma, incentives, repeated trial data), and quantitative results of the experiment (e.g. mean levels of cooperation, variance in cooperation, and effect sizes). The dataset was designed to be fully compliant with the F.A.I.R. principles, and has been

published as an openly available knowledge graph⁵ to allow domain experts to perform their analyses in minutes, instead of many months of painstaking work [22].

Before continuing with the knowledge graph structure, we need to familiarise the reader with the basic concepts of experimental science. Studies using this methodology may observe a relation between two (one independent, one dependent) variables, which can be quantified as an effect size (representing the quantitative result). The goal of the single experiments carried within in a study is to test whether the dependent variable (DV) changes for when modifying the value of the Independent Variable (IV), which indicates there is a relationship between the two variables. In the case of the Databank, one could imagine an experiment aimed at studying the impact (effect size) of a person’s social values (independent variable) over her willingness to cooperate (dependent variable). With this in mind, the CODA knowledge graph includes publications consisting of a `cdo:Paper` class that links to an arbitrary set of `cdo:Study`, i.e. experiments performed in different settings and with different goals. Additional metadata about the paper such as publication date, authors etc. are included as properties of a `cdo:DOI` class. Each `cdo:Study` links to one or more conditions tested, represented by the class `cdo:Observation`, that are in turn modelled as comparisons of one or two different `cdo:Treatment`.

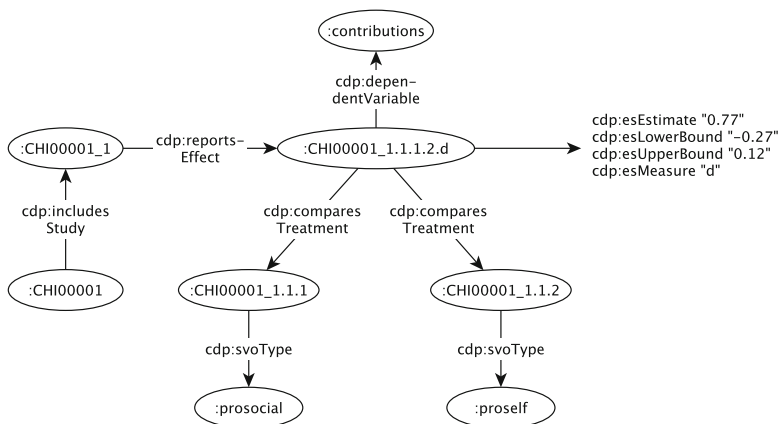


Fig. 1. Example of an observation comparing prosocial vs. prosself behaviour.

In a practical example, Fig. 1 shows the paper `:CHI00001` including the study `:CHI00001_1`, which in turns reports the observation `:CHI00001_1.1.1.2.d` comparing treatment `:CHI00001_1.1.1` and `:CHI00001_1.1.2` (we call them T1 and T2 for simplicity). Treatments consist in the experimental settings that the experimenter modifies with the goal of testing how and if the cooperation between

⁵ <http://data.cooperationdatabank.org/>.

participants of a game varies significantly. In our example, the experimenter manipulated the property `cdp:svoType` which, recalling what stated above, consists then in the independent variable observed. This is confirmed by the fact that T1 and T2 have a different value for the property (`:prosocial` and `:proself` respectively). Similar to `cdp:svoType`, any RDF property whose domain is the class `cdo:Treatment` is organised in a domain-specific taxonomy of independent variables, representing information relative to cooperation in social dilemmas. Finally, in order to represent how and how much the cooperation varies during an observation/experiment, we use the class `cdo:DependentVariable` for the DV and the datatype property `cdp:esEstimate` for the effect size measurement, e.g. `CHI00001.1.1.1.2.d` measures the DV `:contributions` and its effect size has a value of 0.77⁶. The positive effect size reported by the experimental observation means that T1 scored higher on cooperation than T2, indicating that participants with a pro-social value orientation showed a more cooperative behaviour than participants who had a pro-self value orientation.

Challenge, Solution and Novelty. In the scenario above, it is natural to see how the CODA knowledge graph intrinsically represents research hypotheses that were tested in the human cooperation literature. In other words, one can consider each `cdo:Observation` subgraph as a research hypothesis that aims at testing whether there exists a relation between the `cdo:IndependentVariable` and `cdo:DependentVariable`. The effect size value of each observation then tells us the strength of such relation, identified by the experiment performed to validate the hypothesis. The research question we ask is therefore: is it possible to learn new, plausible observations starting from the representations recorded in CODA? More generally, how to support domain experts in producing new research hypotheses through a more automated method? The solution we propose is to frame the problem of learning research hypotheses as a link prediction task, where we exploit the existing `cdo:Observation` subgraph structures to learn new unseen triples involving a `cdo:IndependentVariable` and `cdo:DependentVariable`. Our hypothesis is that entities and relationships neighbouring the predicted links can help completing the new research hypotheses. Our work main novelty is applying a hybrid method to automatically support social scientists for the first time, following similar approaches in the biomedical field. We train a knowledge graph embedding model to predict the likelihood of a new possible association between an IV and a DV. We then develop a system that suggests new possible research hypotheses including both triples existing in CODA and new predicted triples according to a predefined structure. Accuracy and meaningfulness of the discovered hypotheses are assessed quantitatively and qualitatively in a user-study based on the domain expertise of social scientists from the field.

⁶ CODA contains two types of effect size measures, i.e. the correlation coefficient ρ and the standardized mean difference d , which can be easily converted to one another. For simplicity, we will only refer to Cohen's d values from now on.

4 Approach

The proposed approach includes three steps: a pre-processing phase for data selection and generation of the model input (Sect. 4.1), a learning phase including parameter tuning, model training, and link prediction (Sect. 4.2), and a last phase for the automated generation of hypotheses (Sect. 4.3).

4.1 Pre-processing

The first step is to choose the right amount of CODA information to retrieve, and create an input for the embedding model to be able to predict new triples.

Observation Selection. First, we define a set of criteria to select the CODA observations, namely:

1. instances of the class `cdo:Observation`;
2. observations reporting using Cohen’s d as effect size measure;
3. observations comparing two treatments;
4. observations linking to an instance of a `cdo:DependentVariable`.

The SPARQL query used to get the observations can be found online⁷, and results in 4,721 observations, the study, paper and DOI that reported them, the effect size with confidence levels, the experimental design, and sample size and standard deviation per treatment pair.

A further refinement is performed by analysing the independent variables of each observation. We identify the properties-values for which the two treatments compared by an observation differ on, e.g. `cdp:svoType/:prosocial` vs. `cdp:svoType/:proself` in the example of the previous section. To prevent noise and reduce complexity, we dropped observations that had no differing predicates (errors attributed to the large sparsity of the data and to human annotation), or that might differ for more than one property. This left 2,444 observations to train the model, coming from 632 papers and 858 studies, and including 128 unique IVs and 2 unique DVs.

Data Permutation. Since KG embedding methods are generally not capable of learning continuous variables, we learn effect sizes as categorical instead of continuous information. This is also motivated by the fact that Cohen’s d is in fact a measure that can be interpreted categorically [18]. To this end, we created a new RDF property `cdp:esType` and a set of 5 instances of the class `cdo:ESType` that a `cdo:Observation` might point to, representing the 5 bins mapping the continuous effect size values to Cohen’s categories⁸. Table 1 shows the ranges for each bin/instance, and their respective effect size types.

⁷ <https://data.cooperationdatabank.org/coda/-/queries/link-prediction-selection-query>.

⁸ Due to the relatively small sets, medium and large effects were grouped together.

Table 1. Effect size ranges, their interpretation and the respective instance created.

Effect size range	Intepretation	Instance
–Infinity, –0.5	Large/medium negative correlation	:largeMediumNegativeES
–0.5, –0.2	Small negative correlation	:smallNegativeES
–0.2, 0.2	No correlation	:nullFinding
0.2, 0.5	Small positive correlation	:smallPositiveES
0.5, infinity	Large/medium positive correlation	:largeMediumPositiveES

As also explained in Sect. 3, an effect size is an indication of the size of the correlation between an independent and a dependent variable, measured based on the different IV values that two treatments take during an experimentation. This means that, in order to predict a new correlation between IV and DV, one would have to predict multiple triples, i.e. at least one per treatment (and their respective IV values). In order to simplify the task, we summarise the factor that influences the effect size into a single node, by considering IV values pairs as single hypotheses. We therefore combine all possible values for a given IV property into pairs, assigning a hypothesis number to each pair, and create a new node that is linked to the original T1/T2 values through the property `cdp:hypothesis`. The new nodes, shown e.g. in Table 2, are then used for the hypothesis generation. For continuous properties reporting many different values in the object position, four different ranges were automatically created to prevent the generation of an excessive amount of hypotheses. Similarly, pairs with the same IV values in a different order were considered as the same hypothesis (e.g. T1 = proself/T2 = prosocial and T1 = prosocial/T2 = proself were both linked to `:SV0typeH2`), but the effect size node of the observation was switched (positive to negative, or vice versa) to maintain the direction of the correlation coherent.

Table 2. Hypothesis nodes based on combinations of IV values for T1 and T2.

IV	T1 value	T2 value	Hypothesis node
SVO type	Individualist	Prosocial	:SV0type_H1
SVO type	Prosocial	Proself	:SV0type_H2
SVO type	Individualist	Altruist	:SV0type_H3

We then link the created hypothesis nodes to the dependent variable nodes using three new predicates, related to the type of correlation that is observed: `cdp:hasPositiveEffect0n`, `cdp:hasNoEffect0n`, `cdp:hasNegativeEffect0n`. These properties are based on the statistical significance of the observation, computed using the 95% confidence interval for the effect size. A confidence interval measures the imprecision of the computed effect size in an experiment. When the interval does not include 0, it can be inferred that the association is statistically

significant ($p < 0.05$). In other words, the confidence interval tells us how trustworthy is the observation we are analysing, in terms of effect size, population estimate, and direction of the effect. Depending on the confidence interval, we use `:hasNoEffectOn` if the effect size is not significant, while `:hasNegativeEffectOn` and `:hasPositiveEffectOn` are used with observations indicating a significant negative and positive correlation between IV and DV, respectively. When no confidence interval was given in the data, we derive the direction of the correlation using the rule of thumb as reported of [18]: observations with an effect size below -0.2 got a negative effect property, observations that reported effect sizes above 0.2 got a positive effect property, and observations with an effect size between -0.2 and 0.2 got a no effect property. This led us to a total of 751 positive effect triples, 1,017 no effect triples and 676 negative effect triples.

Dataset Creation. The last step of the pre-processing task consists in the conversion into learnable subgraphs, i.e. sets of triples. To do this, we use part of the information already in the data, namely observation ID, the independent and dependent variables, the IV values for the two treatments, and combine them with the computed effect size type, the hypothesis number, and relationship to the dependent variable. A construct query⁹ was used to generate subgraphs as depicted in Fig. 2 for 2,444 observations. This led to a dataset of 29,339 triples, that served as input for the link prediction model.

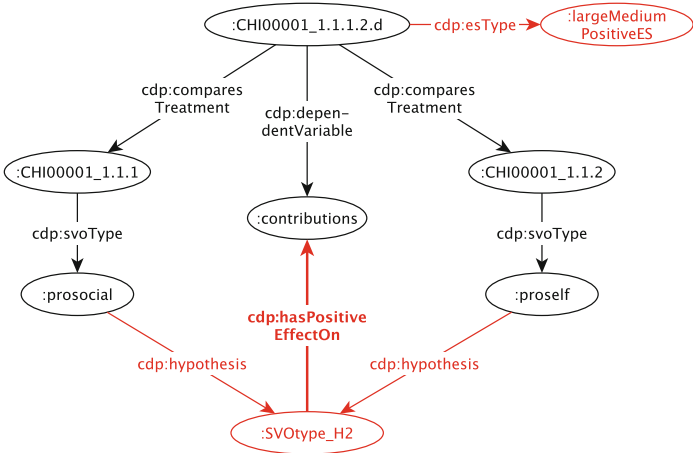


Fig. 2. New graph for the observation CHI00001_1.1.1.2.d used as input for the link prediction task. In red, the nodes and edges created. In bold, the link to be predicted. (Color figure online)

⁹ <https://data.cooperationdatabank.org/coda/-/queries/Rosaline-Construct-Link-Prediction>.

Table 3. Final parameter configuration.

Parameter	Value
Batches_count	555
Epochs	100
k (dimensionality)	200
η (# neg. samples generated per each pos.)	15
Loss	multiclass_nll
Embedding_model_params	{‘negative_corruption_entities’: ‘all’}
LP regulariser params	{‘p’:1, ‘lambda’:1e-5}
Xavier initialiser params	{‘uniform’: False}
Adam optimizer params	{‘lr’: 0.0005}

4.2 Learning and Predicting Triples

Training and Testing. We use the created dataset to learn a model predicting unseen triples to be used in new hypotheses. Strictly speaking, the prediction consists in identifying triples including a hypothesis number, an effect size predicate and a dependent variable, e.g. $\langle \text{SV0type_H2 cdp:hasPositiveEffectOn :contributions} \rangle$. To do this, all triples reporting a negative or a positive effect were gathered. We decided not to make predictions for the no-effect triples, as experts might be less interested in non-interesting relations between variables to frame their hypotheses. Investigating this for future work could be interesting. From the total 1,427 effect triples, the 243 unique hypotheses in subject position, the 2 unique predicates and the 2 unique dependent variables in object position were used to learn how to generate new combinations. This yielded to $243 * 2 * 2 = 972$ total triples, of which 412 were already in the dataset and marked as “seen”, while the other 560 were denoted as “unseen”.

We used the ComplEx model to learn the likelihood of each triple. We first split the dataset into a training set of 24,539 triples, a test set of 2,400 triples, and a validation set of 2,400 triples. A corruption strategy is then used to generate negative statements. Parameter tuning was finally performed to explore impact on the model performance, see Table 3 for the final configuration. Standard metrics such as mean reciprocal rank, hits@N and mean rank were used to evaluate the trained model.

Link Prediction. The learnt model was used to compute ranks and scores for unseen triples. Ranks indicate the position at which the test set triple was found when performing link prediction, while scores are the returned raw scores generated by the model. Probabilities of unseen triples are also calculated by calibrating the model. We set a positive base rate of 0.5 (50%) to indicate the ratio between positive vs. negative triples. After calibration, a probability for each

unseen triple was predicted. We then obtained their ranks, score and probabilities for the 560 unseen triples, to be later used during the hypotheses generation step. A sample of these is in Table 4 below.

Table 4. Prediction example of unseen triples.

Statement	Rank	Score	Prob.
:iteratedStrategy_H6 cdp:hasPositiveEffectOn :cooperation	1	7.38	0.98
:iteratedStrategy_H9 cdp:hasPositiveEffectOn :cooperation	2	7.32	0.98
			...
:uncertaintyTarget_H1 cdp:hasPositiveEffectOn :cooperation	3816	0.10	0.19
:exitOption_H1 cdp:hasNegativeEffectOn :contributions	4659	-0.03	0.17

4.3 Hypotheses Generation

The final step is to automatically generate human-interpretable hypotheses, based on the unseen triples predicted by the model. Each statement from Table 4 was converted into a readable text using a prefixed structure following the DISK ontology. A *hypothesis statement* was created by disassembling the triples into respectively the independent variable (the predicted subject), the type of effect (the predicted predicate) and dependent variable (the predicted object). The *hypothesis evidence* was created by querying the CODA knowledge graph for labels of both IVs and DVs, and by converting the effect type property into decapitalised words with spacing. We also retrieve the description of both the IV class and the relevant IV values. The *hypothesis history* was built by retrieving the DOIs of papers that studied that combination of IV values. An example of a generated hypothesis is shown below.

Hypothesis Statement

Partner’s group membership has negative effect on contributions

Hypothesis Evidence

Dependent Variable (DV): <https://data.cooperationdatabank.org/id/dependentvariable/contributions>

Independent Variable (IV): <https://data.cooperationdatabank.org/vocab/prop/targetMembership>

Whether the participant is interacting with a partner identified as ingroup, outgroup, or stranger.

The IV values to compare in the treatments (T1, T2) are :

Treatment	IV value	Description
T1	ingroup	Partner(s) is a member of the participant’s group
T2	ingroup_and_outgroup	When an experimental treatment explicitly provides information that a partner or group belongs to both an ingroup and an outgroup

Hypothesis History

<http://dx.doi.org/10.1016/j.joep.2013.06.005>

<http://dx.doi.org/10.1177/0146167205282149>

<http://dx.doi.org/10.1016/j.ijintrel.2011.02.017>

Implementation. The current approach was implemented using Python 3.7.7. The ComplEx model was implemented using the Ampligraph¹⁰ library. All the code and results can be found on GitHub¹¹. The queries were made using the SPARQL API service of the CODA knowledge graph, hosted by TriplyDB¹².

5 Evaluation

We first quantitatively and qualitatively evaluate the model performance through known metrics and inspection of the independent variable embeddings. We then evaluate the generated hypotheses through domain experts.

5.1 Model Performance

We compare our model to the TransE, ComplEx and DistMult models trained with their default parameters. The resulting performance metrics are shown in Table 5. We used three different types of metrics as indication of how well the models were capable of predicting the triples in the test set: mean reciprocal rank (MRR), hits@N, and mean rank (MR). Reciprocal rank measures the correctness of a ranked triple, and mean RR is defined as $MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i}$, where Q

¹⁰ <https://github.com/Accenture/AmpliGraph>.

¹¹ https://github.com/roosyay/CoDa_Hypotheses.

¹² <https://coda.triply.cc/>.

is the number of triples and $rank_i$ the rank of the i th triple predicted by the model. Hits@N indicates how many triples are ranked in the top N positions when ranked against corruptions, i.e.: $Hits@N = \frac{1}{|Q|} \sum_{(s,p,o) \in Q} ind(rank(s,p,o) \leq N)$ where Q is the triples in the test set, (s,p,o) is a triple $\in Q$, and $ind(\cdot)$ is an indicator function returning 1 if the positive triple is in the top N triples, 0 otherwise. We use three values for N , namely 1, 3 and 10. Finally, the MR score is the sum of the true ranks divided by the total amount of ranks, defined as $MR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} rank_{(s,p,o)_i}$. Note that the MR score is not robust to outliers, and is therefore only taken into account together with the other metrics. Overall, these scores indicate a reasonable performance of ComplEx, and confirm our idea that link prediction methods in general can be used for hypothesis discovery. Some room for improvement is left for future work, namely applying our method to other datasets, and fine-tuning additional machine learning techniques.

Table 5. Model comparison.

	MRR	Hits@10	Hits@3	Hits@1	MR
ComplEx	0.60	0.66	0.62	0.56	736.77
DistMult	0.48	0.62	0.57	0.38	683.31
TransE	0.30	0.46	0.35	0.20	330.42
Tuned ComplEx	0.68	0.75	0.69	0.64	279.91

5.2 Qualitative Analysis

To get insight into how the model effectively learnt the data, we created a visualisation of the main independent variables (see Fig. 3). To do this, the 400-dimensional embeddings of 128 unique independent variables were retrieved from the trained model and transformed into an array of (128, 400). We used a UMAP reduction to reduce the 400 dimensions to 2 only, allowing then to display the embeddings in a 2-dimensional space. In order to find the optimal number of clusters in this space, we used an elbow method measuring the Within-Cluster Sums of Squares (WCSS) without finding any significant distinction. We therefore used a silhouette analysis, revealing that 23 clusters was the best balance between the number of clusters and a relatively high silhouette score (silhouette score = 0.49). Clusters were obtained using scikit-learn’s KMeans ($K = 23$) and the visualisation was obtained using the Matplotlib package.

As shown in Fig. 3, most clusters are groups of variables that are `rdfs:subClassOf` the same class. For example, in clusters 3, 11 and 15, all the variables related to respectively punishment, emotion and leadership are clustered together. Clusters such as 2 and 13 seem to have less cohesion, as no overarching topics can be found that group IVs together. This can be due by a larger variety of the studies that analysed these IV, and potentially a lack of more data. Both clusters also include variables related to reward, showing that studies with reward-related IV are more heterogeneous and were not grouped together.

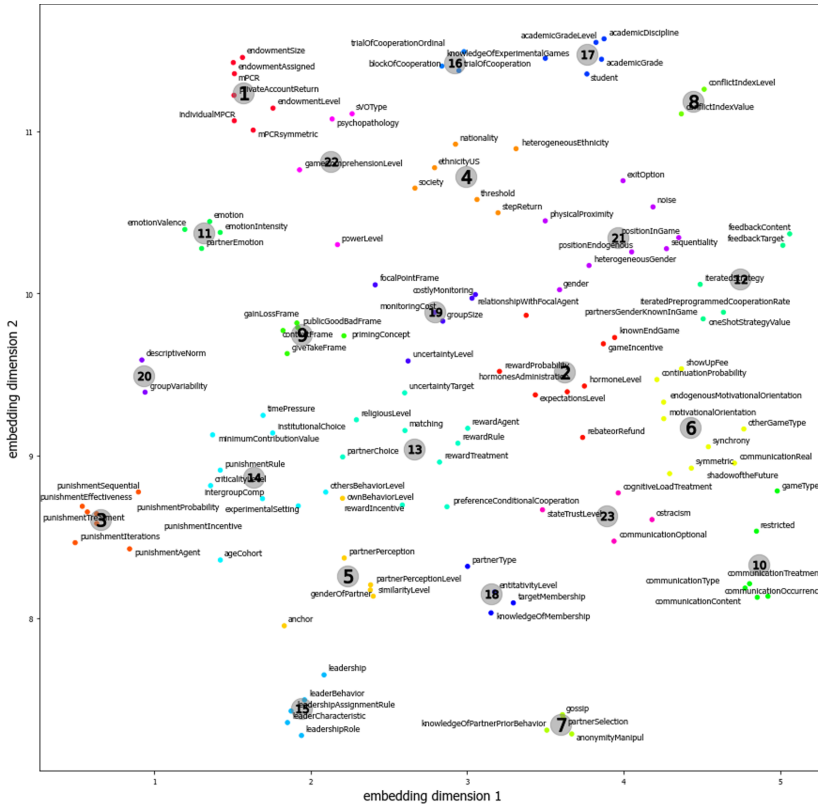


Fig. 3. Independent variables grouped in 23 clusters. Please cfr. Visualisation.ipynb on Github for better quality.

5.3 Domain Expert Evaluation

In order to qualitatively evaluate the generated hypotheses, 5 domain experts from the CODA team were asked to fill out a user-study. A Google form was created where the experts, after receiving information about the background and the goal of the study, were shown the 10 most likely and 10 most unlikely hypotheses predicted by the model. The 20 hypotheses were shown in a random order using the structure presented in Sect. 4.3. The experts were asked to indicate which 10 hypotheses they considered likely, and which 10 unlikely. A final part for remarks was also included.

Table 6 shows how the experts rated the likelihood of the hypotheses. These can be easily read as “[Hypothesis Statement] *when comparing* [T1 value] vs. [T2 value]”, e.g. “SVO type has negative effect on cooperation when comparing a group of individualists vs. a group of competitors”. Two hypotheses including miscellaneous IV values did not make sense according to the experts and were omitted. Overall, the majority of the experts rated 12 out of 18 hypotheses as

Table 6. Expert evaluation. #L and #UL refer to the number of experts scoring a hypothesis as likely and unlikely, respectively. *Pred.* indicates the model prediction.

	Hypothesis statement	T1 value	T2 value	#L	#UL	Pred.
1	MPCR has positive effect on contributions	(−0.401, 0.3)	(0.3, 0.5)	3	2	Likely
2	Partner’s group membership has negative effect on contributions	Ingroup	Ingroup and outgroup	0	5	Likely
3	Intergroup competition has positive effect on contributions	Individual group	Intergroup competition	4	1	Likely
4	Anonymity manipulation has positive effect on cooperation	High	Low	3	2	Likely
5	Time pressure has negative effect on contributions	Time-pressure	Time delay	2	3	Likely
6	SVO type has negative effect on cooperation	Individualist	Competitor	3	2	likely
7	Ethnicity (us) has positive effect on cooperation	White	Black or African American	1	4	Likely
8	Iterated strategy has positive effect on cooperation	Predominantly cooperative	Other	4	1	Likely
9	Nationality has negative effect on contributions	JPN	AUS	3	2	Unlikely
10	Exit option has negative effect on contributions	0	1	2	3	Unlikely
11	Exit option has positive effect on contributions	0	1	1	4	Unlikely
12	Emotion has negative effect on cooperation	Neutral	Disappointment	2	3	Unlikely
13	Emotion has positive effect on cooperation	Neutral	Disappointment	2	3	Unlikely
14	Preference for conditional cooperation has negative effect on cooperation	Freeriders	Hump-shaped contributors	4	1	Unlikely
15	Uncertainty target has positive effect on cooperation	Loss	Threshold	4	1	Unlikely
16	Iterated strategy has positive effect on contributions	Tit-for-tat	Tit-for-tat+1	1	4	Unlikely
17	Preference for conditional cooperation has positive effect on cooperation	Freeriders	Hump-shaped-contributors	1	4	Unlikely
18	Uncertainty target has negative effect on cooperation	Loss	Threshold	0	5	Unlikely

the model did, while only 6 hypotheses were rated opposite of the model. Out of 5 experts, 2 rated more than 9 hypotheses the same as the model, which is higher than chance level, while the other 3 experts scored exactly on chance level. No experts scored below chance level. It should be noted that some experts took more time to fill the evaluation form, as they provided more details in the open-ended questions, and some variety could be seen in how experts rated the hypotheses. We relate this to the complexity of social science data, causing

different perspectives to reach different conclusions. Looking at the overall average however, the experts rated 10 hypotheses the same as the model did. This shows that the model output is not random, and that similarities between the expert opinions and the model were found. More in general, we consider our results encouraging enough to confirm the idea that a link prediction-based approach is a valuable method to predict hypotheses over structured data.

6 Conclusions

We have introduced a hybrid approach to automatically support domain experts to identify new research hypotheses. Our novel solution is based on a link prediction task over a knowledge graph in the social science domain, where new edges between nodes are predicted in order to create fully formulated hypotheses in the form of a hypothesis statement, a hypothesis evidence and a hypothesis history. The quantitative and qualitative evaluation carried using experts in the field has shown encouraging results, namely that a simple combination of machine learning and knowledge graphs methods can support designing more complex systems for the automated scientific discovery.

Improvements of our approach could be made as future work, namely by optimising the data modelling and the machine learning approach for social science data. Such type of data has in fact an uncertain nature, and missing information can create an inner bias in the model and have implications for the results. Solutions to cope with such bias should be investigated. As mentioned, the approach should be also tested on datasets of different domains to see how it could perform. Some information from the data was lost due to binning continuous variables (including effect size values), and the learning task could be improved on this aspect. An end-to-end task could be envisioned to learn the subgraphs directly, instead of pre-processing them and reducing the task to predicting one link between two entities. Finally, our model generally predicted only links with the highest probabilities based on statistical frequency learnt from the data structure. An interesting avenue would be to investigate what makes an hypothesis interesting other than popularity, and how to learn them.

References

1. Bahler, D., Stone, B., Wellington, C., Bristol, D.W.: Symbolic, neural, and Bayesian machine learning models for predicting carcinogenicity of chemical compounds. *J. Chem. Inf. Comput. Sci.* **40**(4), 906–914 (2000). <https://doi.org/10.1021/ci990116i>
2. Bianchi, F., Rossiello, G., Costabello, L., Palmonari, M., Minervini, P.: Knowledge graph embeddings and explainable AI (April 2020). <https://doi.org/10.3233/SSW200011>
3. Chen, N.C., Drouhard, M., Kocielnik, R., Suh, J., Aragon, C.R.: Using machine learning to support qualitative coding in social science: shifting the focus to ambiguity. *ACM Trans. Interact. Intell. Syst.* **8**(2), 1–3 (2018). <https://doi.org/10.1145/3185515>

4. Clark, T., Ciccarese, P.N., Goble, C.A.: Micropublications: a semantic model for claims, evidence, arguments and annotations in biomedical communications. *J. Biomed. Semant.* **5**(1), 1–33 (2014). <https://doi.org/10.1186/2041-1480-5-28>
5. Dessì, D., Osborne, F., Reforgiato Recupero, D., Buscaldi, D., Motta, E., Sack, H.: AI-KG: an automatically generated knowledge graph of artificial intelligence. In: Pan, J.Z., et al. (eds.) *ISWC 2020. LNCS*, vol. 12507, pp. 127–143. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-62466-8_9
6. Garijo, D., et al.: Towards automated hypothesis testing in neuroscience. In: Gadeppally, V., et al. (eds.) *DMAH/Poly -2019. LNCS*, vol. 11721, pp. 249–257. Springer, Cham (2019). <https://doi.org/10.1007/978-3-030-33752-0-18>
7. Garijo, D., Gil, Y., Ratnakar, V.: The DISK hypothesis ontology: capturing hypothesis evolution for automated discovery. *CEUR Workshop Proc.* **2065**, 40–46 (2017)
8. Groth, P., Gibson, A., Velterop, J.: The anatomy of a nanopublication. *Inf. Serv. Use* **30**(1–2), 51–56 (2010). <https://doi.org/10.3233/ISU-2010-0613>
9. Huang, S., Wan, X.: AKMiner: domain-specific knowledge graph mining from academic literatures. In: Lin, X., Manolopoulos, Y., Srivastava, D., Huang, G. (eds.) *WISE 2013. LNCS*, vol. 8181, pp. 241–255. Springer, Heidelberg (2013). <https://doi.org/10.1007/978-3-642-41154-0-18>
10. Katukuri, J.R., Xie, Y., Raghavan, V.V., Gupta, A.: Hypotheses generation as supervised link discovery with automated class labeling on large-scale biomedical concept networks. *BMC Genomics* **13**(Suppl 3), 12–15 (2012). <https://doi.org/10.1186/1471-2164-13-s3-s5>
11. Nagarajan, M., et al.: Predicting future scientific discoveries based on a networked analysis of the past literature. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 2019–2028 (2015)
12. Natarajan, N., Dhillon, I.S.: Inductive matrix completion for predicting gene-disease associations. *Bioinf.* **30**(12), 60–68 (2014). <https://doi.org/10.1093/bioinformatics/btu269>
13. Nickel, M., Murphy, K., Tresp, V., Gabrilovich, E.: A review of relational machine learning for knowledge graphs (2016). <https://doi.org/10.1109/JPROC.2015.2483592>
14. Nye, B., et al.: A corpus with multi-level annotations of patients, interventions and outcomes to support language processing for medical literature. In: *ACL 2018*, vol. 1, pp. 197–207 (2018). <https://doi.org/10.18653/v1/p18-1019>
15. Pankratius, V., et al.: Computer-aided discovery: toward scientific insight generation with machine support why scientists need machine support for discovery search. *IEEE Intell. Syst.* **31**(4), 3–10 (2016). <https://doi.org/10.1109/MIS.2016.60>
16. Sang, S., et al.: GrEDeL: a knowledge graph embedding based method for drug discovery from biomedical literatures. *IEEE Access* **7**(2016), 8404–8415 (2019). <https://doi.org/10.1109/ACCESS.2018.2886311>
17. Sateli, B., Witte, R.: Semantic representation of scientific literature: bringing claims, contributions and named entities onto the Linked Open Data cloud. *PeerJ Comput. Sci.* **2015**(12), 1–e37 (2015). <https://doi.org/10.7717/peerj-cs.37>
18. Sawilowsky, S.S.: New Effect Size Rules of Thumb. *J. Mod. Appl. Stat. Methods* **8**(2), 597–599 (2009). <https://doi.org/10.22237/jmasm/1257035100>

19. Sosa, D.N., Derry, A., Guo, M., Wei, E., Brinton, C., Altman, R.B.: A literature-based knowledge graph embedding method for identifying drug repurposing opportunities in rare diseases. *Pacific Symposium on Biocomputing* **25**, 463–474 (2020). https://doi.org/10.1142/9789811215636_0041
20. Srinivasan, P.: Text mining: generating hypotheses from MEDLINE. *J. Am. Soc. Inf. Sci. Technol.* **55**(5), 396–413 (2004). <https://doi.org/10.1002/asi.10389>
21. Swanson, D.R., Smalheiser, N.R.: An interactive system for finding complementary literatures: a stimulus to scientific discovery. *Artif. Intell.* **91**(2), 183–203 (1997). [https://doi.org/10.1016/S0004-3702\(97\)00008-8](https://doi.org/10.1016/S0004-3702(97)00008-8)
22. Tiddi, I., Balliet, D., ten Teije, A.: Fostering scientific meta-analyses with knowledge graphs: a case-study. In: Harth, A., et al. (eds.) *ESWC 2020. LNCS*, vol. 12123, pp. 287–303. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-49461-2_17
23. Trouillon, T., Welbl, J., Riedel, S., Ciatossier, E., Bouchard, G.: Complex embeddings for simple link prediction. In: *33rd International Conference on Machine Learning, ICML 2016*, vol. 5, pp. 3021–3032 (2016)
24. Wallace, B.C., Kuiper, J., Sharma, A., Zhu, M., Marshall, I.J.: Extracting PICO sentences from clinical trial reports using supervised distant supervision (2016)
25. Ware, M., Mabe, M.: The STM report: an overview of scientific and scholarly journal publishing (2015)