Combining Information Extraction, Deductive Reasoning and Machine Learning for Relation Prediction

Xueyan Jiang[2], Yi Huang[1,2], Maximilian Nickel[2], and Volker Tresp[1,2]

[1] Siemens AG, Corporate Technology, Munich, Germany
[2] Ludwig Maximilian University of Munich, Munich, Germany

Abstract. Three common approaches for deriving or predicting instantiated relations are information extraction, deductive reasoning and machine learning. Information extraction uses subsymbolic unstructured sensory information, e.g. in form of texts or images, and extracts statements using various methods ranging from simple classifiers to the most sophisticated NLP approaches. Deductive reasoning is based on a symbolic representation and derives new statements from logical axioms. Finally, machine learning can both support information extraction by deriving symbolic representations from sensory data, e.g., via classification, and can support deductive reasoning by exploiting regularities in structured data. In this paper we combine all three methods to exploit the available information in a modular way, by which we mean that each approach, i.e., information extraction, deductive reasoning, machine learning, can be optimized independently to be combined in an overall system. We validate our model using data from the YAGO2 ontology, and from Linked Life Data and Bio2RDF, all of which are part of the Linked Open Data (LOD) cloud.

1 Introduction

The prediction of the truth value of a (instantiated) relation or statement (i.e., a link in an RDF graph) is a common theme in such diverse areas as information extraction (IE), deductive reasoning and machine learning. In the course of this paper we consider statements in form of (s, p, o) RDF-triples where s and o are entities and where p is a predicate. In IE, one expects that the relation of interest can be derived from subsymbolic unstructured sensory data such as texts or images and the goal is to derive a mapping from sensory input to statements. In deductive reasoning, one typically has available a set of facts and axioms and deductive reasoning is used to derive additional true statements. Relational machine learning also uses a set of true statements but estimates the truth values of novel statements by exploiting regularities in the data. Powerful methods have been developed for all three approaches and all have their respective strengths and shortcomings. IE can only be employed if sensory information is available that is relevant to a relation, deductive reasoning can only derive a small subset of all statements that are true in a domain and relational machine learning is only applicable if the data contains relevant statistical structure. The goal of this paper is to combine the strengths of all three approaches modularly, in the sense that each step can be optimized independently. In a first step, we extract triples using IE, where we assume that the extracted triples have associated certainty values. In this paper we will only consider IE from textual data. Second, we perform deductive reasoning to derive the set of provably true triples. Finally, in the third step, we employ machine learning to exploit the dependencies between statements. The predicted triples are then typically ranked for decision support. The complete system can be interpreted as a form of scalable hierarchical Bayesian modeling. We validate our model using data from the YAGO2 ontology, and from Linked Life Data and Bio2RDF, all of which are part of the Linked Open Data (LOD) cloud.

The paper is organized as follows. The next section discusses related work. Section 3 describes and combines IE and deductive reasoning. Section 4 describes the relational learning approach. Section 5 presents various extensions and in Section 6 we discuss scalability. Section 7 contains our experimental results and Section 8 presents our conclusions.

2 Related Work

Multivariate prediction generalizes supervised learning to predict several variables jointly, conditioned on some inputs. The improved predictive performance in multivariate prediction, if compared to simple supervised learning, has been attributed to the sharing of statistical strength between the multiple tasks, i.e., data is used more efficiently (see [32] and citations therein for a review). Due to the large degree of sparsity of the relationship data in typical semantic graph domains, we expect that multivariate prediction can aid the learning process in such domains.

Our approach is also related to conditional random fields [20]. The main differences are the modularity of our approach and that our data does not exhibit the linear structure in conditional random fields.

Recently, there has been quite some work on the relationship between kernels and graphs [7] [33] [11]. Kernels for semi-supervised learning have, for example, been derived from the spectrum of the Graph-Laplacian. Kernels for semantically rich domains have been developed by [8]. In [36] [35] approaches for Gaussian process based link prediction have been presented. Link prediction is covered and surveyed in [27] [13]. Inclusion of ontological prior knowledge to relational learning has been discussed in [28].

From early on there has been considerable work on supporting ontologies using machine learning [24] [9] [21], while data mining perspectives for the Semantic Web have been described by [1] [25]. A recent overview of the state of the art has been presented in [29]. The transformation of text into the RDF structure of the semantic web via IE is a highly active area of research [23] [30] [5] [6] [2] [4] [34] [3] [26] [14]. [22] describes a perspective of ILP for the Semantic Web. We consider machine learning approaches that have been applied to relation prediction in the context with the Semantic Web. In [19] the authors describe SPARQL-ML, a framework for adding data mining support to SPARQL. SPARQL-ML was inspired by Microsoft's Data Mining Extension (DMX). A particular ontology for specifying the machine learning experiment is developed. The approach uses Relational Bayes Classifiers (RBC) and Relational Probabilistic Trees (RPT).

3 Combining Sensory Information and Knowledge Base

3.1 Relation Prediction from Sensory Inputs

The derivation of relations from subsymbolic unstructured sensory information such as texts and images is a well-studied area in IE. Let X stand for a random variable that has state one if the (s, p, o) statement of interest is true and is zero otherwise. We assume that the IE component can estimate

$$P(X = 1|S)$$

which is the probability that the statement represented by X is true given the sensory information S. Otherwise no restrictions apply to the IE part in our approach, e.g., it could be based on rules or on statistical classifiers. Note that IE is limited to predict statements for which textual or other sensory information is available.

In the applications we have textual information text_s describing the subject and textual information text_o describing the object and we can write[3]

$$P(X = 1|\text{text}_s, \text{text}_o). \tag{1}$$

In other applications we might also exploit text that describes the predicate text_p or text that describes the relationship $\text{text}_{s,p,o}$ (e.g, a document where a user (s) evaluates a movie (o) and the predicate is p="likes") [16]. A recent overview on state of the art IE methods for textual data can be found in [29].

3.2 Relations from the Knowledge Base

In addition to sensory information, we assume that we have available a knowledge base in form of a triple store of known facts forming an RDF graph. Conceptually we add all triples that can be derived via deductive reasoning.[4] State of the art scalable deductive reasoning algorithms have been developed, e.g., in [10]. Note that deductive reasoning typically can only derive a small number of nontrivial statements of all actually true statements in a domain.

We will also consider the possibility that the knowledge base contains some uncertainty, e.g., due to errors in the data base. So for triples derived from the knowledge base KB we specify

$$P(X = 1|KB)$$

[3] For example, these texts might come from the corresponding Wikipedia pages.

[4] Here, those tripes can either be inferred explicitly by calculating the deductive closure or on demand.

to be a number close to one.

For all triples that cannot be proven to be true, we assume that $P(X = 1|KB)$ is a small nonnegative number. This number reflects our belief that triples not known to be true might still be true.

3.3 Combining Sensory Information and Knowledge Base

Now we combine sensor information and information from the knowledge base. Let $P(X = 1|S, KB)$ be the probability that the statement presented by X is true given the knowledge base and sensory information. The heuristic rule that we apply is very simple:

$$P(X = 1|S, KB) = P(X = 1|S) \quad \text{if} \quad P(X = 1|S) > P(X = 1|KB)$$

$$P(X = 1|S, KB) = P(X = 1|KB) \quad \text{otherwise.}$$

Thus the probability of a statement derived from sensory information overwrites the default knowledge base values, if the former one is larger. Therefore, we rely on the knowledge base unless IE provides substantial evidence that a relation is likely.

4 Adding Relational Machine Learning

In many applications there is information available that is neither captured by sensory information nor by the knowledge base. A typical example is collaborative preference modeling which exploits correlations between preferences for items. Such probabilistic dependencies cannot easily be captured in logical expressions and typically are also not documented in textual or other sensory form. Relational machine learning attempts to capture exactly these statistical dependencies between statements and in the following we will present an approach that is suitable to also integrate sensory information and a knowledge base. Although there are probably a number of heuristic ways to combine sensory information and the knowledge base with machine learning, it is not straightforward to come up with consistent probabilistic models. Probabilistic generative models would require $P(S, KB|\{X\})$ where $\{X\}$ is the set of all random variables of all statements. Unfortunately, it is not clear how such a term could be derived. In the next subsections we develop an approach that works with the simpler $P(X|S, KB)$ and can be justified from a Bayesian modeling point of view.

4.1 Notation

Consider (s, p, o) triple statements where s and o are entities and p is a predicate. Note that a triple typically describes an attribute of a subject, e.g., (Jack, height, Tall), or a relationship (Jack, likes, Jane). Consider, that $\{e_i\}$ is the set of known entities in the domain. We assume that each entity is assigned to exactly one class $c(i)$. This assumption will be further discussed in Section 5. Let N_c be the number of entities in class c.

We also assume that the set of all triples in which an entity e_i can occur as a subject is known and is a finite, possibly large, ordered set (more details later) and contains $M_{c(i)}$ elements. For each potential triple (s, p, o) we introduce a random variable X which is in state one when the triple is true and is in state zero otherwise. More precisely, $X_{i,k} = 1$ if the k-th triple involving e_i as a subject is true and $X_{i,k} = 0$ otherwise. Thus, $\{X_{i,k}\}_{k=1}^{M_{c(i)}}$ is the set of all random variables assigned to the subject entity e_i.

We now assume that there are dependencies between all statements with the same subject entity.

4.2 A Generative Model

Following the independence assumptions we train a separate model for each class. So in this section we only consider the subset of statements which all have entities from the same entity class c.

The generative model is defined as follows. We assume that for each entity e_i which is a subject in class c there is a d-dimensional latent variable vector h_i which is generated as

$$h_i \sim N(0, I) \tag{2}$$

from a Gaussian distribution with independent components and unit-variance.

Then for each entity e_i a vector $\alpha_i = (\alpha_{i,1}, \ldots, \alpha_{i,M_c})^T$ is generated, following

$$\alpha_i = A h_i \tag{3}$$

where A is a $M_C \times d$ matrix with orthonormal columns.

From α_i we derive

$$P(X_{i,k} = 1 | S, KB) = \text{sig}(\alpha_{i,k}) \tag{4}$$

where $\text{sig}(in) = 1/(1 + \exp(-in))$ is the logistic function. In other words, $\alpha_{i,k}$ is the true but unknown activation that specifies the probability of observing $X_{i,k} = 1$. Note that $\alpha_{i,k}$ is continuous with $-\infty < \alpha_{i,k} < \infty$ such that a Gaussian distribution assumption is sensible, whereas discrete probabilities are bounded by zero and one.

We assume that $\alpha_{i,k}$ is not known directly, but that we have a noisy version available for each $\alpha_{i,k}$ in the form of

$$f_{i,k} = \alpha_{i,k} + \epsilon_{i,k} \tag{5}$$

where $\epsilon_{i,k}$ is independent Gaussian noise with variance σ^2. $f_{i,k}$ is now calculated in the following way from sensory information and the knowledge base. We simply write

$$\hat{P}(X_{i,k} = 1 | S, KB) = \text{sig}(f_{i,k})$$

and sensory and the knowledge base is transferred into

$$f_{i,k} = \text{inv-sig}(\hat{P}(X_{i,k} = 1|S, KB)) \tag{6}$$

where inv-sig is the inverse of the logistic function. Thus probabilities close to one are mapped to large positive f-values and probabilities close to zero are mapped to large negative f-values. The resulting F-matrix contains the observed data in the probabilistic model (see Figure 1).

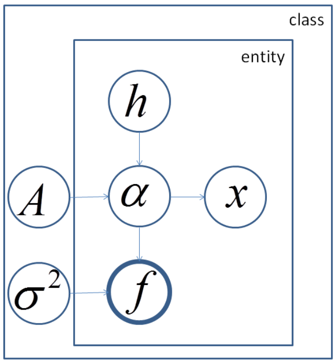

Fig. 1. Graphical plate model for the data generating process.

4.3 Calculating the Solution

Note that our generative model corresponds to the probabilistic PCA (pPCA) described in [31] and thus we can use the learning equations from that paper.

Let F be the $N_c \times M_c$ matrix of f-values for class c and let

$$C = F^T F$$

be the empirical correlation matrix. The likelihood is maximized when

$$\hat{A} = U_d(\Lambda_d - \sigma^2 I)^{1/2} R \tag{7}$$

where the d column vectors in the $N_c \times d$ matrix U_d are the principal eigenvectors of C, with corresponding eigenvalues $\lambda_1, ..., \lambda_d$ in the $d \times d$ diagonal matrix Λ_d and R is an arbitrary $d \times N_c$ orthogonal rotation matrix.[5] We also get

$$\hat{\sigma}^2 = \frac{1}{M_c - d} \sum_{j=d+1}^{M_c} \lambda_j.$$

[5] A practical choice is the identity matrix $R = I$. Also note that we assume that the mean is equal to zero, which can be justified in sparse domains.

Finally, we obtain

$$\hat{\alpha}_i = \hat{A} M^{-1} \hat{A}^T f_i. \tag{8}$$

Here, $f_i = (f_{i,1}, \ldots, f_{i,M_c})^T$ is the vector of f-values assigned to e_i and $M = \hat{A}^T \hat{A} + \hat{\sigma}^2 I$. Note that M is diagonal such that the inverse is easily calculated as

$$\hat{\alpha}_i = U_d \operatorname{diag}\left(\frac{\lambda_j - \hat{\sigma}^2}{\lambda_j}\right) U_d^T f_i. \tag{9}$$

$\hat{\alpha}_i$ is now used in Equation 4 to determine the probability that $X_{i,k} = 1$, which is then, e.g., the basis for ranking. Also

$$\operatorname{diag}\left(\frac{\lambda_j - \hat{\sigma}^2}{\lambda_j}\right)$$

is a diagonal matrix where the j-th diagonal term is equal to $\frac{\lambda_j - \hat{\sigma}^2}{\lambda_j}$.[6]

5 Comments and Extensions

5.1 A Joint Probabilistic Model

There are many ways of looking at this approach, maybe the most interesting one is a hierarchical Bayesian perspective. Consider each $\alpha_{i,k}$ to be predicted as a function of S and KB. In hierarchical Bayesian multitask learning one makes the assumption that, for a given entity e_i, the $\{\alpha_{i,k}\}_{k=1}^{M_{c(i)}}$ are not independent but are mutually coupled and share statistical strength [12]. This is achieved exactly by making the assumption that they are generated from a common multivariate Gaussian distribution. Thus our approach can be interpreted as hierarchical Bayesian multitask learning which can scale up to more than a million of tasks, i.e., potential statements per item.

Note that we suggest to train an independent model for each class and we obtain a joint probabilistic model over a complete domain with

$$P(\{X\}, \{h\} | \{F\}, \Theta) = \prod_c \prod_{i:c(i)=c} P(X_i | \alpha_i(h_i))\, P(f_i | \alpha_i(h_i))\, P(h_i).$$

$P(h_i)$ is given by Equation 2, where the dimension d might be dependent on the class $c(i)$ and $\alpha_i(h_i)$ is given by Equation 3. $P(X_i | \alpha_i(h_i))$ is given by Equation 4 (with $X_i = \{X_{i,k}\}_{k=1}^{M_{c(i)}}$) and $P(f_i | \alpha_i(h_i), \sigma_c^2)$ is given by Equation 5. Furthermore, $\{F\}$ is the set of F matrices for all classes and Θ is the set of all parameters, i.e., the A matrices and the σ^2 for all classes.

[6] Note the great similarity of Equation 9 to the reduced rank penalized regression equation in the SUNS approach described in [15] which, in the notation of this paper, would assume the form $U_d \operatorname{diag}(\lambda_j/(\lambda_j + \gamma))\, U_d^T f_i$ where $\gamma \geq 0$ is a regularization parameter. In some experiments we used this equation which exhibited greater numerical stability.

Note that each class is modeled separately, such that, if the number of entities per class and potential triples per entity are constant, machine learning scales linearly with the size of the knowledge base.

Finally we want to comment on how we define the set of all possible triples under consideration. In most applications there is prior knowledge available about what triples should be considered. Also, typed relations constrain the number of possible triples. In some applications it makes sense to restrict triples based on observed triples: We define the set of all possible statements in a class c to be all statements (s, p, o) where s is in class c and where the triple (s, p, o) has been observed in the data for at least one element of s $\in c$.

5.2 Generalization to New Entities

The most interesting case is when a new entity e_n that was not considered in training becomes known. If the class of the new entity is known, one can simply use Equation 8 to calculate a new α_n for a new f_n, which corresponds to the projection of a new data vector in pPCA. In case the class of the new entity is unknown, we can calculate α_n for the different classes under consideration and use Equation 5 to calculate the class specific probability.

5.3 Aggregation

After training, the learning model only considers dependencies between triples with the same subject entity. Here we discuss how additional information can be made useful for prediction.

Supplementing the Knowledge Base The first approach is simply to add a logical construct into deductive reasoning that explicitly adds aggregated information. Let's assume that the triple (?Person, livesIn, Germany) can be predicted with some certainty from (?Person, bornIn, Germany). If the triple store does not contain the latter information explicitly but contains information about the birth city of a person, one can use a rule such as

$$\begin{aligned}&\text{(?Person, bornIn, Germany)}\\ \leftarrow\ &\text{(?Person, bornIn, ?City)} \wedge \text{(?City, locatedIn, Germany)}\end{aligned}$$

and the derived information can be used in machine learning to predict the triple (?Person, livesIn, Germany).

Enhancing IE Some aggregation happens at the IE level. As an example, consider a text that describes a person (subject) and reveals that this person is a male teenager and consider another text that reveals that a movie (object) is an action movie. Then an IE system can learn that (Person, likes, Movie) is more likely when the keywords "male", "young" are present in the text describing the person and the keyword "action" is present in the test describing the movie.

We can also enhance the textual description using information from the knowledge base. If the knowledge base contains the statement (Person, gender, Male) and (Person, age, Young), we add the terms “male” and “young” to the keywords describing the person. Similarly, if the knowledge base contains the statement (Movie, isGenre, Action), we add the term “action” to the keywords describing the movie.

5.4 Multiple Class Memberships

So far we have assumed that each entity can uniquely be assigned to a class. In many ontologies, an entity is assigned to more than one class. The most straightforward approach is to define for each entity a most prominent class. For example we might decide that from the class assignments (Jack, rdf:type, Student), (Jack, rdf:type, Person), (Jack, rdf:type, LivingBeing) that the second one is the prominent class which is used in the probabilistic model. The other two class assignments (i.e., type-of relations) are simply interpreted as additional statements (Jack, rdf:type, Student), (Jack, rdf:type, LivingBeing) assigned to the entity. As part of future work we will develop mixture approaches for dealing with multiple class assignments, but this is beyond the scope of this paper.

6 Scalability

We consider the scalability of the three steps: deductive reasoning, IE, and machine learning. Deductive reasoning with less expressive ontologies scales up to billions of statements [10]. Additional scalability can be achieved by giving up completeness. As already mentioned, each class is modeled separately, such that, if the number of entities per class and potential triples per entity are constant, machine learning scales linearly with the size of the knowledge base. The expensive part of the machine learning part is the eigen decomposition required in Equation 7. By employing sparse matrix algebra, this computation scales linearly with the number of nonzero elements in F. To obtain a sparse F, we exploit the sensory information only for the test entities and train the machine learning component only on the knowledge base information, i.e., replace $\hat{P}(X_{i,k} = 1|S, KB)$ with $\hat{P}(X_{i,k} = 1|KB)$ in Equation 6. Then we assume that $P(X = 1|KB) = \epsilon$ is a small positive constant ϵ for all triples that are not and cannot be proven true. We then subtract inv-sig(ϵ) from F prior to the decomposition and add inv-sig(ϵ) to all α. The sparse setting can handle settings with millions of entities in each class and millions of potential triples for each entity.

7 Experiments

7.1 Associating Diseases with Genes

As the costs for gene sequencing are dropping, it is expected to become part of clinical practice. Unfortunately, for many years to come the relationships between genes and diseases will remain only partially known. The task here is to predict diseases that are likely associated with a gene based on knowledge about gene and disease attributes and about known gene-disease patterns.

Disease genes are those genes involved in the causation of, or associated with a particular disease. At this stage, more than 2500 disease genes have been discovered. Unfortunately, the relationship between genes and diseases is far from simple since most diseases are polygenic and exhibit different clinical phenotypes. High-throughput genome-wide studies like linkage analysis and gene expression profiling typically result in hundreds of potential candidate genes and it is still a challenge to identify the disease genes among them. One reason is that genes can often perform several functions and a mutational analysis of a particular gene reveal dozens of mutation cites that lead to different phenotype associations to diseases like cancer [18]. An analysis is further complicated since environmental and physiological factors come into play as well as exogenous agents like viruses and bacteria.

Despite this complexity, it is quite important to be able to rank genes in terms of their predicted relevance for a given disease as a valuable tool for researchers and with applications in medical diagnosis, prognosis, and a personalized treatment of diseases.

In our experiments we extracted information on known relationships between genes and diseases from the LOD cloud, in particular from Linked Life Data and Bio2RDF, forming the triples (Gene, related_to, Disease). In total, we considered 2462 genes and 331 diseases. We retrieved textual information describing genes and diseases from corresponding text fields in Linked Life Data and Bio2RDF. For IE, we constructed one global classifier that predicts the likelihood of a gene-disease relationship based on the textual information describing the gene and the disease. The system also considered relevant interaction terms between keywords and between keywords and identifiers and we selected in total the 500 most relevant keywords and interaction terms. We did the following experiments

- ML: We trained a model using only the gene disease relationship, essentially a collaborative filtering system. Technically, Equation 6 uses $\hat{P}(X_{i,k} = 1|KB)$, i.e., no sensory information.
- IE: This is the predictive performance based only on IE, using Equation 1.
- ML + IE: Here we combine ML with IE, as discussed in the paper. We combine the knowledge base with IE as described in Section 3.3 and then apply Equation 6 and Equation 8.

Figure 2 shows the results. As can be seen, the performance of the IE part is rather weak and ML gives much better performance. It can nicely be seen that the combination of ML and IE is effective and provides the best results.

7.2 Predicting Writer's Nationality in YAGO2

The second set of experiments was done on the YAGO2 semantic knowledge base. YAGO2 is derived from Wikipedia and also incorporates WordNet and GeoNames. There are two available versions of YAGO2: core and full. We used the

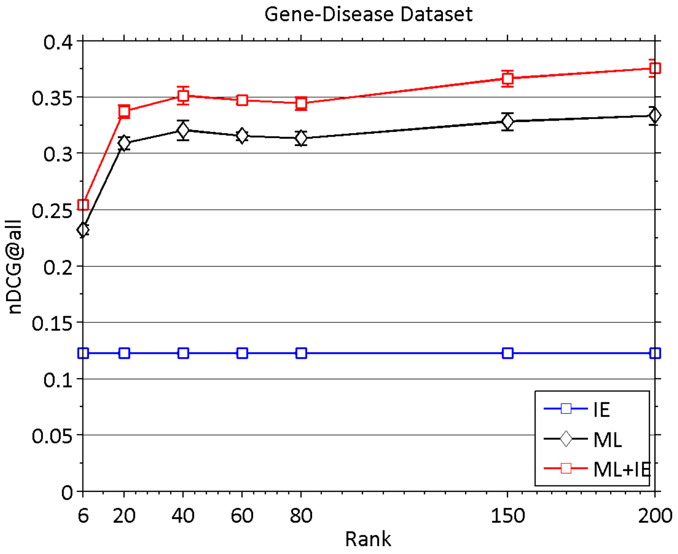

Fig. 2. Results on the Gene-Disease Data as a function of the rank d of the approximation. For each gene in the data set, we randomly selected one related_to statement to be treated as unknown (test statement). In the test phase we then predicted all unknown related_to entries, including the entry for the test statement. The test statement should obtain a high likelihood value, if compared to the other unknown related_to entries. The normalized discounted cumulative gain (nDCG@all) [17] is a measure to evaluate a predicted ranking.

first one, which currently contains 2.6 million entities, and describes 33 million facts about these entities. Our experiment was designed to predict the nationalities of writers. We choose four different types of writers: American, French, German and Japanese. E.g., the triples for American writers are obtained with the SPARQL query:

```
SELECT ?writer ?birthPlace ?location WHERE {
    ?writer rdf:type ?nationality .
    ?writer yago:wasBornIn ?birthPlace .
    ?birthPlace yago:isLocatedIn ?location .
    FILTER regex(str( ?nationality ), "American_writers", "i")
}
```

We obtained 440 entities representing the selected writers. We selected 354 entities with valid yago:hasWikipediaUrl statements. We built the following five models:

- ML: Here we considered the variables describing the writers' nationality (in total 4) and added information on the city where a writer was born. In total, we obtained 233 variables. Technically, Equation 6 uses $\hat{P}(X_{i,k} = 1|KB)$, i.e., no sensory information.
- IE: As textual source, we used the Wikipage of the writers. We removed the terms 'German, French, American, Japanese' and ended up with 36943 keywords.
- ML+IE: We combined the knowledge base with IE as described in Section 3.3 and then applied Equation 6 and Equation 8.
- ML+AGG: We performed geo-reasoning to derive the country where a writer is born from the city that a writer was born. This aggregate information was added as a statement to the writer. Naturally, we expect a high correlation between country of birth and the writer's nationality (but there is no 100% agreement!).
- ML+AGG+IE: As ML+AGG but we added IE information using Equation 1.

We performed 10-fold cross validation for each model, and evaluated them with the area under precision and recall curve. Figure 3 shows the results. We see that the ML contribution was weak but could be improved significantly by adding information on the country of birth (ML+AGG). The IE component gives excellent performance but ML improves the results by approximately 3 percentage points. Finally, by including geo-reasoning, the performance can be improved by another percentage point. This is a good example where all three components, geo-reasoning, IE and machine learning fruitfully work together.

8 Conclusions

In this paper we have combined information extraction, deductive reasoning and relational machine learning to integrate all sources of available information in a

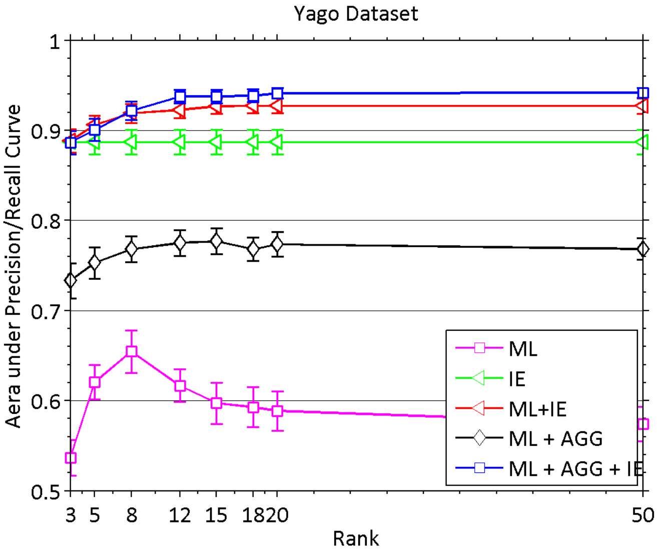

Fig. 3. The area under curve for the YAGO2 Core experiment as a function of the rank d of the approximation.

modular way. IE supplies evidence for the statements under consideration and machine learning models the dependencies between statements. Thus even if it is not evident that a patient has diabetes just from IE from text, our approach has the ability to provide additional evidence by exploiting correlations with other statements, such as the patient's weight, age, regular exercise and insulin intake. We discussed the case that an entity belongs to more than one ontological class and addressed aggregation. The approach was validated using data from the YAGO2 ontology, and the Linked Life Data ontology and Bio2RDF. In the experiments associating diseases with genes we could show that our approach to combine IE with machine learning is effective in applications where a large number of relationships need to be predicted. In the experiments on predicting writer's nationality we could show that IE could be combined with machine learning and geo-reasoning for the overall best predictions. In general, the approach is most effective when the information supplied via IE is complementary to the information supplied by statistical patterns in the structured data and if reasoning can add relevant covariate information.

References

1. Bettina Berendt, Andreas Hotho, and Gerd Stumme. Towards semantic web mining. In *ISWC*, 2002.
2. Chris Biemann. Ontology learning from text: A survey of methods. *LDV Forum*, 20(2), 2005.
3. Paul Buitelaar and Philipp Cimiano. *Ontology Learning and Population: Bridging the Gap between Text and Knowledge*. IOS Press, 2008.
4. Philipp Cimiano. *Ontology Learning and Population from Text: Algorithms, Evaluation and Applications*. Springer-Verlag, 2006.
5. Philipp Cimiano, Andreas Hotho, and Steffen Staab. Comparing conceptual, divise and agglomerative clustering for learning taxonomies from text. In *Proceedings of the 16th Eureopean Conference on Artificial Intelligence, ECAI'2004*, 2004.
6. Philipp Cimiano and Steffen Staab. Learning concept hierarchies from text with a guided agglomerative clustering algorithm. In *Proceedings of the ICML 2005 Workshop on Learning and Extending Lexical Ontologies with Machine Learning Methods*, 2005.
7. Chad M. Cumby and Dan Roth. On kernel methods for relational learning. In *ICML*, 2003.
8. Claudia D'Amato, Nicola Fanizzi, and Floriana Esposito. Non-parametric statistical learning methods for inductive classifiers in semantic knowledge bases. In *IEEE International Conference on Semantic Computing - ICSC 2008*, 2008.
9. Nicola Fanizzi, Claudia dAmato, and Floriana Esposito. Dl-foil: Concept learning in description logics. In *ILP*, 2008.
10. Dieter Fensel, Frank van Harmelen, Bo Andersson, Paul Brennan, Hamish Cunningham, Emanuele Della Valle, Florian Fischer, Zhisheng Huang, Atanas Kiryakov, Tony Kyung il Lee, Lael Schooler, Volker Tresp, Stefan Wesner, Michael Witbrock, and Ning Zhong. Towards larkc: A platform for web-scale reasoning. In *ICSC*, pages 524–529, 2008.
11. T. Gärtner, J.W. Lloyd, and P.A. Flach. Kernels and distances for structured data. *Machine Learning*, 57(3), 2004.
12. Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian Data Analysis*. Chapman and Hall/CRC Texts in Statistical Science, 2 edition, 2003.
13. Lise Getoor and Christopher P. Diehl. Link mining: a survey. *SIGKDD Explorations*, 2005.
14. Marko Grobelnik and Dunja Mladenic. Knowledge discovery for ontology construction. In John Davies, Rudi Studer, and Paul Warren, editors, *Semantic Web Technologies*. Wiley, 2006.
15. Yi Huang, Volker Tresp, Markus Bundschus, Achim Rettinger, and Hans-Peter Kriegel. Multivariate prediction for learning on the semantic web. In *ILP*, 2010.
16. Niklas Jakob, Mark-Christoph Müller, Stefan Hagen Weber, and Iryna Gurevych. Beyond the stars: Exploiting free-text user reviews for improving the accuracy of movie recommendations. In *TSA'09 - 1st International CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion Measurement*, 2009.
17. K. Jarvelin and J. Kekalainen. IR evaluation methods for retrieving highly relevant documents. In *SIGIR'00*, 2000.
18. Maricel G. Kann. Advances in translational bioinformatics: computational approaches for the hunting of disease genes. *Briefing in Bioinformatics*, 11, 2010.

19. Christoph Kiefer, Abraham Bernstein, and Andre Locher. Adding data mining support to sparql via statistical relational learning methods. In *ESWC 2008*. Springer-Verlag, 2008.
20. John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *ICML*, 2001.
21. Jens Lehmann. Dl-learner: Learning concepts in description logics. *JMLR*, 2009.
22. Francesca A. Lisi and Floriana Esposito. An ilp perspective on the semantic web. In *Semantic Web Applications and perspectives*, 2005.
23. Alexander Maedche and Steffen Staab. Semi-automatic engineering of ontologies from text. In *Proceedings of the 12th International Conference on Software Engineering and Knowledge Engineering*, 2000.
24. Alexander Maedche and Steffen Staab. *Handbook on Ontologies 2004*, chapter Ontology Learning. Springer, 2004.
25. Peter Mika. *Social Networks and the Semantic Web*. Springer, 2007.
26. Gerhard Paaß, Jörg Kindermann, and Edda Leopold. Learning prototype ontologies by hierachical latent semantic analysis. In *Knowledge Discovery and Ontologies*, 2004.
27. Alexandrin Popescul and Lyle H. Ungar. Statistical relational learning for link prediction. In *Workshop on Learning Statistical Models from Relational Data*, 2003.
28. Achim Rettinger, Matthias Nickles, and Volker Tresp. Statistical relational learning with formal ontologies. In *ECML/PKDD*, 2009.
29. Sunita Sarawagi. Information extraction. *Foundations and Trends in Databases*, 1(3):261–377, 2008.
30. John F. Sowa. Ontology, metadata, and semiotics. In *International Conference on Computational Science*, 2000.
31. Michael E. Tipping and Chris M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 61:611–622, 1999.
32. Volker Tresp and Kai Yu. Learning with dependencies between several response variables. In *Tutorial at ICML 2009*, 2009.
33. S. V. N. Vishwanathan, Nic Schraudolph, Risi Imre Kondor, and Karsten Borgwardt. Graph kernels. *Journal of Machine Learning Research - JMLR*, 2008.
34. Johanna Völker, Peter Haase, and Pascal Hitzler. Learning expressive ontologies. In Paul Buitelaar and Philipp Cimiano, editors, *Ontology Learning and Population: Bridging the Gap between Text and Knowledge*. IOS Press, 2008.
35. Zhao Xu, Kristian Kersting, and Volker Tresp. Multi-relational learning with gaussian processes. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI-09)*, 2009.
36. Kai Yu, Wei Chu, Shipeng Yu, Volker Tresp, and Zhao Xu. Stochastic relational models for discriminative link prediction. In *Advances in Neural Information Processing Systems (NIPS*2006)*, 2006.