Repairing Inconsistent Taxonomies Using MAP Inference and Rules of Thumb

Elie Merhej
Ghent University
Ghent, Belgium
elie.merhej@ugent.be

Steven Schockaert
Cardiff University
Cardiff, United Kingdom
s.schockaert@cs.cardiff.ac.uk

Martine De Cock*
University of Washington Tacoma
Tacoma, United States of America
mdecock@u.washington.edu

Marjon Blondeel
Vrije Universiteit Brussel
Brussels, Belgium
mblondee@vub.ac.be

Daniele Alfarone
Katholieke Universiteit Leuven
Leuven, Belgium
daniele.alfarone@cs.kuleuven.be

Jesse Davis
Katholieke Universiteit Leuven
Leuven, Belgium
jesse.davis@cs.kuleuven.be

ABSTRACT

Several authors have developed relation extraction methods for automatically learning or refining taxonomies from large text corpora such as the Web. However, without appropriate post-processing, such taxonomies are often inconsistent (e.g. they contain cycles). A standard approach to repairing such inconsistencies is to identify a minimally consistent subset of the extracted facts. For example, we could aim to minimize the sum of the confidence weights of the facts that are removed for restoring consistency. In this paper, we present MAP inference as a base method for this approach, and analyze how it can be improved by taking into account dependencies between the extracted facts. These dependencies correspond to rules of thumb such as "if a given fact is wrong then all facts that have been extracted from the same sentence are also likely to be wrong", which we encode in Markov logic. We present experimental results to demonstrate the potential of this idea.

Categories and Subject Descriptors

I.2.7 [**ARTIFICIAL INTELLIGENCE**]: Natural Language Processing—*Language parsing and understanding*

Keywords

Taxonomy extraction; inconsistency; Markov Logic; MAP inference

1. INTRODUCTION

Relation extraction from natural language is a promising method for learning logical theories. Among others, several authors have proposed methods for learning taxonomies ("is-a" relations [10]) in this way, as well as temporal (*before/after*, *during*, etc. [14]) and spatial (*leftOf*, *under/above*, etc. [13]) relations. Due to the imperfect nature of methods for relation extraction from natural language, the resulting theories are likely to contain mistakes, which usually leads to logical inconsistencies. It is then of interest to repair these inconsistencies, as the theories may otherwise be difficult to use in the considered application, and because doing so may allow us to partially correct the mistakes that were made in the relation extraction process. A common method for repairing a logical theory is to identify a maximally consistent subset, or equivalently, to identify a minimal set of facts to be removed such that consistency is restored. Given that most relation extraction methods provide us with confidence weights, a standard approach is to minimize the sum of the weights of the removed facts. However, we often have some information about the context in which extracted facts have been obtained. This allows us to formulate (soft) dependencies between these facts. For example we could consider that "if a given fact is wrong, then other facts that have been extracted from the same sentence are more likely to be wrong", or that "if a fact is wrong, then other facts that have been obtained from the same document using the same extraction pattern are more likely to be wrong".

Although the methods that we propose are applicable more generally, in this paper we will focus on taxonomies. In particular, we assume that we are given a set of facts in the form of is-a statements, each with an associated confidence weight. For instance, $0.9 : isa(Dog, Animal)$ is a fact where *Dog* is a subtype, *Animal* is its corresponding supertype, and 0.9 is the confidence weight. These facts have either been extracted from text directly, or have been deduced from other extracted terms using some heuristic (e.g. if most entities that are similar to a given entity X are subclasses of Y, we may plausibly derive that X is also likely to be a subclass of Y). We expect that the is-a relationship is transitive, asymmetric and irreflexive, i.e. it satisfies the following rules:

$$isa(A, B) \wedge isa(B, C) \Rightarrow isa(A, C)$$

$$isa(A, B) \Rightarrow \neg isa(B, A)$$

$$\neg isa(A, A).$$

*On leave from Ghent University.

Web-KR'14, November 3, 2014, Shanghai, China.

ACM 978-1-4503-1606-4/14/11 ...$15.00.
http://dx.doi.org/10.1145/2663792.2663804.

These rules are *hard* constraints. When they are applied to the initial set of facts, inconsistencies are likely to arise. To restore consistency, we then need to remove some facts from the initial set. In choosing which set of facts to remove, we usually aim to minimize the sum of the confidence degrees of the removed facts.

Moreover, having some prior knowledge about the considered domain, we incorporate dependencies by imposing a number of *soft* constraints, in addition to the aforementioned hard constraints that encode the semantics of the is-a relationship. These soft constraints are essentially "rules of thumb" that express our intuition about the problem domain. Unlike hard constraints, these soft constraints do not necessarily need to be satisfied; their only aim is to help us find more plausible repairs.

To repair inconsistencies in the set of extracted facts, we encode our problem in Markov logic [18] because it allows us to specify both soft and hard constraints in an intuitive way. We use *Maximum-A-Posteriori* (MAP) inference [8] as our base method to find the most likely set of facts that satisfies all the constraints.

The paper is structured as follows. In Section 2, we recall some basic notions about Markov logic, Markov Logic Networks (MLN) and MAP inference. In Section 3, we show how we encode the evidence obtained by a taxonomy as an MLN and briefly describe how MAP inference is used to repair inconsistencies in the noisy set of extracted facts. We present our running example, based on an animal taxonomy extracted from the Web. In Section 4, we introduce the rules of thumb that make explicit the dependencies between facts. We report experimental results in Section 5. In Section 6, we discuss related work. We conclude in Section 7.

2. MARKOV LOGIC

Markov logic is a probabilistic extension of first-order logic [18]. In this paper, we will restrict ourselves to a function-free version of Markov logic. Moreover, we will assume that the domain of each variable is finite. In this case, a first-order logic theory can be seen as a concise representation of a propositional theory, which can be obtained using a process called grounding. An expression of the form $P(x_1, ..., x_n)$, where P is a predicate symbol, and $x_1, ..., x_n$ are constants or variables, is called an atom. In our running example, atoms are in the form of $isa(x, y)$ and denote that x is a subtype of y. When all arguments of an atom are constants, it is called a ground atom. The groundings of a clause are formed by replacing every variable in this clause with constants in all possible ways. The Herbrand base **B**(C) of a set of clauses C is the set of all ground atoms constructable from the predicate symbols and constants in C. We define a possible world as a mapping from the Herbrand base to {true, false}, i.e. a mapping assigning a truth value to every ground atom in the language.

A *Markov Logic Network* (MLN) is a set of first-order formulas, each with an associated weight in $\mathbb{R}$ [18]. This weight intuitively shows the importance of the formula it is assigned to. Instead of presuming a world to be impossible if it violates even one formula, Markov logic allows a world to violate one or more formulas and still be considered possible, but less probable. Naturally, the more formulas the world violates, and the higher our confidence is in these formulas, the less probable it becomes. Thus, formulas in Markov logic can be seen as soft constraints, and their weights as penalties.

Together with a finite set of constants, an MLN defines a log-linear probability distribution of possible worlds as follows:

$$P(X = x) = \frac{1}{Z} \exp\left(\sum_{i=1}^{F} w_i n_i(x)\right) \quad (1)$$

where X is a random variable representing the true world and x is a possible world, Z is the normalization constant, F is the number of first-order formulas in the MLN, w_i is the weight of the ith formula, and $n_i(x)$ is the number of true groundings of the ith formula in x. Note that hard constraints, i.e. formulas that have to be satisfied, can be presented using infinite weights.[1] They resemble pure first-order logic formulas.

Given a set of observed (input) atoms, *Maximum -A- Posteriori* (MAP) inference aims to find the most likely truth assignment of a set of query (output) atoms. This corresponds to finding a truth assignment for the output variables that maximizes the sum of the weights of the satisfied formulas in the grounded version of the MLN. To perform MAP inference, we use the solver RockIt [16], which compiles MAP queries to integer linear programs, and solves them internally using the integer linear solver Gurobi.[2] During the mapping process, RockIt applies the Cutting Plane Inference (CPI) algorithm described in [19] and the Cutting Plane Aggregation (CPA) algorithm [16] to speed-up the inference.

3. MAP INFERENCE TO REPAIR INCONSISTENCIES IN TAXONOMIES

To encode our available evidence about a taxonomy as an MLN, we first add a number of hard constraints that encode the semantics of the is-a relationship:

$$\textit{isa}(A, B) \wedge \textit{isa}(B, C) \Rightarrow \textit{isa}(A, C). \quad (2)$$

$$\textit{isa}(A, B) \Rightarrow \neg\textit{isa}(B, A). \quad (3)$$

$$\neg\textit{isa}(A, A). \quad (4)$$

To repair a taxonomy, we propose the following MLN. We use as evidence the is-a facts extracted from a Web corpus, along with their respective confidence weights computed by [1]. Additionally, the MLN contains the rules (2)-(4), defining the semantics of the is-a relationship, as hard constraints (i.e. formulas with infinite weight). These hard constraints will ensure that the output of the MAP inference process is a consistent set of facts. The MAP inference output then tells us which facts need to be removed to restore consistency.

4. RULES OF THUMB FOR MORE PLAUSIBLE REPAIR

One of the limitations of the MAP inference approach outlined in the previous section is that it treats all confidence weights as being independent. However, in practice two facts may have been extracted from the same sentence, or using the same extraction pattern, which makes it more likely that they are either both correct or both incorrect. Moreover, in some problem domains, we may have prior knowledge (or intuitions) about which type of configurations are more likely. For example, it has been observed in [1] that long

[1] In most solvers, an "infinite" weight is represented by a very large number (e.g. sum of the weights of soft constraints + 10).

[2] http://www.gurobi.com

chains of is-a relations usually indicate an error, even though having long chains may not lead to logical inconsistencies. To take account of dependencies between facts and prior knowledge, we will add a number of additional rules to the Markov logic theory.

4.1 Introducing Features

We want to introduce rules that describe dependencies between facts that share particular features. These features will correspond to meta-data about how these facts have been extracted, such as which sentence the fact was extracted from, which extraction pattern was used to obtain the fact [10], etc. Features can also encode that a fact has not been extracted from text directly, but has instead been inferred using a heuristic. For instance, in [1], extracted terms from text are put into clusters based on their similarity. If a term S is sufficiently similar to terms that share a supertype T, we plausibly infer *isa*(S, T). Moreover, facts can also be obtained using direct modifiers, e.g. adjectives that appear in front a known taxonomy term. For example, in the animals taxonomy, the extracted term *Ectothermic Species* contains the modifier *Ectothermic* from which we can infer *isa*$(Ectothermic\_Species, Species)$.

The idea is that if two facts share the same feature, we enforce the soft constraint that if one of the facts is true, the other is more likely to be true as well (and vice versa if one of them is false, the other is also more likely to be false). In our taxonomy example, we use three types of features. The first type is the sentence from which a fact was extracted. The second feature type is a modifier that describes multiple taxonomy terms. For example, in the animals corpus, the terms *Ectothermic Species* and *Ectothermic Animals* infer the facts *isa*$(Ectothermic\_Species, Species)$ and *isa*$(Ectothermic\_Animals, Animals)$. We propose that these two facts are dependent since they were inferred from the same modifier. The third feature type is an ID that represents a cluster of taxonomy terms, i.e. subtypes that share the same supertype. Using this ID, we can encode that two facts that have been deduced from the same cluster of sister-terms are also dependent.

4.2 Encoding Rules of Thumb

To encode the rules of thumb in the MLN, instead of only having facts of the form *isa*(x, y), we now also consider facts of the form *FromSentence*(s_i, x, y), *FromModifier*(m_i, x, y) and *FromCluster* (c_i, x, y), where x and y are respectively the subtype and supertype of an extracted is-a relationship, and s_i, m_i and c_i each correspond to a feature from one of the three types described earlier. For example, if *isa*(A, B) and *isa*(C, D) were both extracted from the sentence s_1, we add the facts *FromSentence*(s_1, A, B) and *FromSentence*(s_1, C, D). The following rule then encodes the dependency that facts that have been extracted from the same sentence are likely to be both correct or both incorrect:

$$w_1 : \textit{FromSentence}(s_i, x, y) \wedge \textit{FromSentence}(s_i, u, v) \Rightarrow (\textit{isa}(x, y) \Leftrightarrow \textit{isa}(u, v)) \quad (5)$$

with w_1 a certainty weight indicating how strongly we want to impose this dependency (see below). Similarly, we encode the second dependency rule between two facts that were inferred from the same modifier, and the third one between two facts that were deduced from the same cluster in the following way:

$$w_2 : \textit{FromModifier}(m_i, x, y) \wedge \textit{FromModifier}(m_i, u, v) \Rightarrow (\textit{isa}(x, y) \Leftrightarrow \textit{isa}(u, v)) \quad (6)$$

$$w_3 : \textit{FromCluster}(c_i, x, y) \wedge \textit{FromCluster}(c_i, u, v) \Rightarrow (\textit{isa}(x, y) \Leftrightarrow \textit{isa}(u, v)) \quad (7)$$

All these dependency rules are soft constraints with finite weights. We update our MLN by adding these rules to the hard constraints (2), (3) and (4). The weights w_1, w_2 and w_3 will be learned from training data (see below).

4.3 Learning Dependency Weights

Given an MLN containing rules[3] with unknown weights, and a set of facts that are known to be true, as training data, weight learning algorithms try to compute the optimal weights of the MLN rules that maximise the likelihood of the training data atoms. To learn these weights, we use the solver Alchemy [11], which implements a generative learner as described in [9].

5. EXPERIMENTAL RESULTS

We use MAP inference with dependency rules to repair inconsistencies in three taxonomies extracted from Web corpora: an animal taxonomy, a vehicle taxonomy and a plant taxonomy. From the animal taxonomy, 5064 facts were extracted and used as input observed data. We manually evaluated 2447 of them as a random subset, and found that 1196 facts are *correct*, and 1251 facts are *incorrect*. Moreover, from the vehicle taxonomy, 5755 facts were extracted and 2638 were manually evaluated. From the 2638 facts, we found that 1194 are *correct* and 1444 are *incorrect*. Finally, from the plant taxonomy, 14034 facts were extracted from which 1489 were manually evaluated. We found that 1000 facts of them are *correct* and 489 are *incorrect*. The goal is to discard as many of the incorrect facts as possible, while retaining as many of the correct facts as possible. In other words, we want to reduce the number of false positives (FP) while not losing true positives (TP) in the process. To evaluate our results, we use the following metrics: we first calculate the percentage of correct evidence facts that are retained in the MAP inference output, and the percentage of incorrect evidence facts that are lost. We then compute the F_1 *score* which considers both the precision p and recall r of our tests in the following way:

$$F_1 \; score = (2 \times p \times r) \; / \; (p + r)$$

with $p = \frac{(number\ of\ correct\ facts\ retained)}{(number\ of\ all\ facts\ retained)}$

and $r = \frac{(number\ of\ correct\ facts\ retained)}{(number\ of\ all\ correct\ facts)}$.

Finally, we compute the classification accuracy of our approach. For each fact, we are basically making a decision, i.e. that the fact is correct (which means we keep it) or that the fact is incorrect (which means we get rid of it), depending if it is present in the MAP inference output or not. We also know the true state of the fact from manual evaluation. Classification accuracy measures in what percentage of cases our decision corresponds to the true state of a fact.

We start with the MLN that contains only the hard constraints (rules (2)-(4)), and add the facts extracted from the animal taxonomy, the vehicle taxonomy and the plant taxonomy separately. To repair inconsistencies in the extracted facts, we run MAP inference on the resulting MLNs and report the results in tables 1, 2 and 3 respectively. For the animal taxonomy, we notice that 35.9% of the *incorrect* evidence facts were removed, while 75.8% of the *correct* facts were retained. For the vehicle taxonomy, 32.2% of the *incorrect* facts were removed, and 82.1% of the *correct* facts retained.

[3] Rules with a large number of variables should be avoided because they have too many groundings, hence making the learning process longer.

As for the plant taxonomy, 55.6% of the *incorrect* facts were removed and 77.8% of the *correct* facts retained.

Next, we add to each MLN the dependency rules as soft constraints. To learn the weights of these dependency rules, we use as training data some manually labelled *correct* facts extracted from the biomedical DDI Web corpus.[4] The extracted taxonomy from this Web corpus contains 772 facts, from which 585 are correct and 74 are incorrect. Note that when input facts are used as training data, their confidence weights are discarded, since these facts were manually verified to be correct. The learning algorithm will set the weights such that the likelihood of these evidence facts is maximised. After learning these weights from the DDI facts, we run MAP inference again on the MLNs with the animal, vehicle and plant taxonomy facts respectively. We report the results in tables 1, 2 and 3 accordingly.

For the animal taxonomy, we notice an increase from 35.9% to 76.4% of *incorrect* evidence facts removed with the addition of the rules of thumb. Also, there is an increase of the percentage of *correct* evidence facts retained from 75.8% to 77.0%. The F_1 $score$ also increased from 0.639 to 0.771. For the classification accuracy metric, we start with an initial value of 51.1% if we assume that all the facts that we extracted from text are *correct*. After doing MAP inference with our MLN containing only hard constraints, this accuracy increased to 56.3%. However, its value reached 76.7% when we added the rules of thumb to our MLN. Similarly for the vehicle taxonomy, the percentage of *incorrect* evidence facts removed increased from 32.3% to 58.6%. The percentage of *correct* evidence facts retained increased as well from 82.1% to 83.1%. There was also an increase in F_1 $score$ from 0.689 to 0.765. For the classification accuracy, it had an initial value of 54.7%, increased to 59.5% after MAP inference with only hard constraints, then reached 72.0% after the addition of rules of thumb to our MLN. Likewise for the plant taxonomy, the percentage of *incorrect* evidence facts removed increased from 55.6% to 72.0%, along with an increase in the percentage of *correct* evidence facts retained from 77.8% to 87.7%. The F_1 $score$ increased as well from 0.780 to 0.871. As for the classification accuracy metric, it starts off with 67.1% assuming all extracted facts are *correct*, increased to 70.5% after doing MAP inference with only hard constraints in our MLN, and reached 82.5% after the addition of rules of thumb.

All our metrics showed a great improvement in MAP inference results when we add rules of thumb to the hard constraints in our MLN. Not only did we remove more *incorrect* facts, but we also retained more *correct* ones. We notice that the addition of rules of thumb affected the percentage of *incorrect* evidence facts removed much more than the percentage *correct* evidence facts retained. We think that this is due to the generally high confidence values of extracted facts that are later deemed *correct*. The dependency rules won't affect these facts greatly as they will probably be present in the MAP inference output since they have a high confidence value and satisfy all hard constraints. Regarding our other metrics, we also have a better F_1 $score$ which means better accuracy in the removal process of *incorrect* facts, and our overall classification accuracy improved drastically.

6. RELATED WORK

Several approaches for dealing with inconsistent knowledge bases have already been proposed. A natural approach, which we also take in this paper, is to revise such a knowledge base by removing or weakening some of the formulas. The key problem with such approaches is that there usually are many different minimal sets of formulas that can be removed to restore consistency. MAP inference makes a particular choice by minimizing the sum of the confidence scores of the removed formulas. This approach is intuitive and straightforward to implement, but from a conceptual point of view, it relies on the assumption that confidence degrees are accurate and additive, and that the truth of one formula does not affect the likelihood that another formula is correct. Therefore, a number of more qualitative alternatives have been proposed, based on the notion of priorities, which can be thought of as order of magnitudes of probabilities [6, 4, 12]. Rather than removing a minimal set of formulas, some approaches aim to weaken formulas in a minimal way, without necessarily removing any formulas [5, 20]. Unfortunately, the complexity of many of these alternatives is prohibitively high (i.e. at the second of even third level of the polynomial hierarchy [7]).

Another class of approaches is based on argumentation systems. The basic idea is then to construct arguments in favour and against a statement, select the acceptable ones and then determine whether the original statement can be accepted or not. If the initial knowledge base is inconsistent, arguments may be defeated by counterarguments. In [2] for example, a preference relation is used to select the most acceptable arguments. A survey on argumentation methods in artificial intelligence can be found in [3]. Other possibilities for managing inconsistency include the use of a paraconsistent logic [17] or involving the user in an interactive process for 'debugging' a logical theory. The latter type of approaches are less useful in the considered context, where we want to analyze inconsistencies for identifying likely errors by an information extraction module. In particular, none of the existing methods seems entirely adequate for this purpose: MAP based approaches require accurate weights while many other approaches are not sufficiently scalable.

Our work is also related to [15]. In this paper the problem of determining correspondences between concepts, properties and individuals of two or more different formal ontologies, i.e. ontology matching, is considered. It is argued that Markov Logic Networks provide an excellent framework for ontology matching. Moreover, it is shown that this approach has several advantages to existing methods.

7. CONCLUSION

To repair an inconsistent set of taxonomy facts extracted from the Web, we encoded these facts in Markov logic and added "rules of thumb" in the form of dependency rules, based on the extraction information shared between these facts. We used MAP inference to find the most likely truth assignment of the input facts and repair inconsistencies accordingly. Our results show that the addition of rules of thumb to the MAP inference approach produces a significant improvement in the process of removing *incorrect* facts that cause inconsistency. We managed to increase the percentage of *incorrect* facts removed while also increasing the percentage of *correct* facts retained, improving the F_1 $score$ and greatly improving the classification accuracy, which lead to a better inconsistency repair.

8. REFERENCES

[1] D. Alfarone and J. Davis. Unsupervised learning of an is-a taxonomy from a limited domain-specific corpus. Technical Report CW664, Katholieke Universiteit Leuven,

[4] http://www.cs.york.ac.uk/semeval-2013/task9/

Table 1: Comparison of results of MAP inference on animal taxonomy

	Incorrect Evidence Facts			Correct Evidence Facts			F-1 Score	Classification Accuracy
	Initial	*Removed*	*%*	*Initial*	*Retained*	*%*		
Only Hard Constraints	1196	430	35.9 %	1251	949	75.8 %	0.639	56.3 %
Hard Constraints + Rules of Thumb	1196	914	76.4 %	1251	963	77.0 %	0.771	76.7 %

Table 2: Comparison of results of MAP inference on vehicle taxonomy

	Incorrect Evidence Facts			Correct Evidence Facts			F-1 Score	Classification Accuracy
	Initial	*Removed*	*%*	*Initial*	*Retained*	*%*		
Only Hard Constraints	1194	384	32.2 %	1444	1186	82.1 %	0.689	59.5 %
Hard Constraints + Rules of Thumb	1194	700	58.6 %	1444	1200	83.1 %	0.765	72.0 %

Table 3: Comparison of results of MAP inference on plant taxonomy

	Incorrect Evidence Facts			Correct Evidence Facts			F-1 Score	Classification Accuracy
	Initial	*Removed*	*%*	*Initial*	*Retained*	*%*		
Only Hard Constraints	489	272	55.6 %	1000	778	77.8 %	0.780	70.5 %
Hard Constraints + Rules of Thumb	489	352	72.0 %	1000	877	87.7 %	0.871	82.5 %

http://www.cs.kuleuven.be/publicaties/rapporten/cw/CW664.abs.html, 2014.

[2] L. Amgoud and C. Cayrol. Inferring from inconsistency in preference-based argumentation frameworks. *Journal of Automated Reasoning*, 29:125–169, 2002.

[3] T. Bench-Capon and P. Dunne. Argumentation in artificial intelligence. *Artificial Intelligence*, 171:619–641, 2007.

[4] S. Benferhat, C. Cayrol, D. Dubois, J. Lang, and H. Prade. Inconsistency management and prioritized syntax-based entailment. In *International Joint Conferences on Artificial Intelligence*, pages 640–645, 1993.

[5] R. Booth. Social contraction and belief negotiation. *Information Fusion*, 7(1):19–34, 2006.

[6] G. Brewka. Preferred subtheories: An extended logical framework for default reasoning. In *International Joint Conferences on Artificial Intelligence*, pages 1043–1048, 1989.

[7] C. Cayrol and M.-C. Lagasquie-Schiex. On the complexity of non-monotonic entailment in syntax-based approaches. In *Proc. of the 11th European Conference on Artificial Intelligence workshop on Algorithms, Complexity and Commonsense Reasoning*, 1994.

[8] M. H. DeGroot. *Optimal statistical decisions*, volume 82. John Wiley & Sons, 2005.

[9] P. Domingos, S. Kok, H. Poon, M. Richardson, and P. Singla. Unifying logical and statistical ai. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*, volume 6, pages 2–7, 2006.

[10] M. A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th Conference on Computational Linguistics*, pages 539–545, 1992.

[11] S. Kok, P. Singla, M. Richardson, P. Domingos, M. Sumner, H. Poon, and D. Lowd. The alchemy system for statistical relational ai. *University of Washington, Seattle*, 2005.

[12] S. Konieczny, J. Lang, and P. Marquis. DA^2 merging operators. *Artificial Intelligence*, 157(1):49–79, 2004.

[13] P. Kordjamshidi, M. Van Otterlo, and M.-F. Moens. Spatial role labeling: Towards extraction of spatial relations from

natural language. *ACM Transactions on Speech and Language Processing*, 8, 2011.

[14] X. Ling and D. S. Weld. Temporal information extraction. In *Association for the Advancement of Artificial Intelligence*, 2010.

[15] M. Niepert, C. Meilicke, and H. Stuckenschmidt. A probabilistic-logical framework for ontology matching. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*, pages 1413–1418, 2010.

[16] J. Noessner, M. Niepert, and H. Stuckenschmidt. Rockit: Exploiting parallelism and symmetry for map inference in statistical relational models. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*, 2013.

[17] G. Priest. Paraconsistent logic. In *Handbook of philosophical logic*, pages 287–393. Springer, 2002.

[18] M. Richardson and P. Domingos. Markov logic networks. *Machine learning*, 62(1-2):107–136, 2006.

[19] S. Riedel. Improving the accuracy and efficiency of map inference for Markov logic. *arXiv preprint arXiv:1206.3282*, 2012.

[20] S. Schockaert and H. Prade. Solving conflicts in information merging by a flexible interpretation of atomic propositions. *Artificial Intelligence*, 175(11):1815–1855, 2011.