Identifying Untyped Relation Mentions in a Corpus given an Ontology

Gabor Melli
VigLink Inc.
539 Bryant St. #400
San Francisco, CA, USA
gabor@viglink.com

Abstract

In this paper we present the $SDOI_{rmi}$ text graph-based semi-supervised algorithm for the task for relation mention identification when the underlying concept mentions have already been identified and linked to an ontology. To overcome the lack of annotated data, we propose a labelling heuristic based on information extracted from the ontology. We evaluated the algorithm on the kdd09cma1 dataset using a leave-one-document-out framework and demonstrated an increase in F1 in performance over a co-occurrence based AllTrue baseline algorithm. An extrinsic evaluation of the predictions suggests a worthwhile precision on the more confidently predicted additions to the ontology.

1 Introduction

The growing availability of text documents and of ontologies will significantly increase in value once these two resources become deeply interlinked such that all of the concepts and relationships mentioned in each document link to their formal definitions. This type of semantic information can be used, for example, to aid information retrieval, textual entailment, text summarization, and ontology engineering (Staab & Studer, 2009; Buitelaar et al, 2009). An obstacle to this vision of semantically grounded documents however is the significant amount of effort required of domain experts to semantically annotate the text (Erdmann et al, 2000; Uren et al, 2006). Some automation of the annotation task is a precondition to the envisioned future of deeply interlinked information. Fortunately, the task of linking concept mentions to their referent in an ontology has matured (Milne & Witten, 2008; Melli & Ester, 2010). Far less progress has been made on the task of linking of relation mentions to the referent relation in a knowledge base. In part, we believe, this is because current approaches attempt to both identify mentions of relations between two or more concepts and to classify the type of the relation, such as one of: IsA(); HeadquarteredIn(); SubcecullarLocalization(), and ComposerOf()

In this paper, we present a weakly-supervised algorithm for the task of **r**elation **m**ention **i**dentification, SDOI [1] $_{RMI}$. Given a corpus of documents whose concept mentions have been identified and linked to an ontology, the algorithm trains a binary classification model that predicts the relations mentioned within a document that should be (and possibly already are) in an ontology. To overcome the lack of explicit annotation of relation mentions, we propose the use of a data labelling heuristic that assigns a TRUE or FALSE label if the candidate mention refers to a link that exists or does not exist in the ontology. $SDOI_{RMI}$.is related to proposals by (Riedel et al, 2010) and (Mintz et al, 2009) except that their proposal attempt to both identify and to classify relation mentions. By only tackling the first (identification) portion of the task our

[1] **SDOI** is for **S**upervised **D**ocument to **O**ntology **I**nterlinking

Proceedings of the TextGraphs-7 Workshop at ACL, pages 30–38,
Jeju, Republic of Korea, 12 July 2012.

algorithm can identify relation mentions of types that are not yet present (or are poorly represented) in the ontology. An extrinsic evaluation of the usability of identified relation mentions to update an ontology provides evidence that $SDOI_{RMI}$'s performance levels can contribute to a real-world setting.

Our envisioned real-world application is to assist a knowledge engineer to process a new set of documents by receiving a ranked list of candidate relation mentions not yet in the ontology. With such a list, the knowledge engineer could dedicate more attention to comprehending the meaning of the passages that (very likely) contain high-quality relation mention candidates.

The paper is structured as follows: we first define our proposed algorithm: **$SDOI_{rmi}$**,, and conclude with an empirical analysis of its performance.

2 Algorithm Overview

For the task of relation mention identification, we propose a semi-supervised algorithm inspired by the **TeGRR** text graph-based relation recognition algorithm proposed in (Melli & al, 2007). The algorithm first applies a labelling heuristic to unlabeled candidate relation mentions, and then trains a binary classification model. We were motivated to follow this approach used by **TeGRR** for the following reasons:

1) It is based on relation recognition approaches, such as (Jiang & Zhai, 2007), that achieve state-of-the-art performance (e.g. on benchmark tasks such as ACE[2]).
2) It is designed to recognize relation mentions that span beyond a single sentence (by the use of a text graph representation)
3) It exposes an extensible feature space (that can be extended with information drawn from our task's ontology).
4) It provides a natural path for the future support of tasks with labelled training data – possibly even labelled with the actual relation type.

One of the distinctive aspects of **TeGRR** is its representation of a document into a graph-based representation, where each concept mention or token in the text is mapped to an 'external' node in a graph, and which represents other syntactic and structural features of the text as internal nodes and edges between nodes. In Section 3 we define the text graph representation and its effect on the algorithm definition.

Given a document's text-graph, we can proceed to define a feature space for each relation mention candidate. Table 1 illustrates the structure of the training data and its feature space that we propose for $SDOI_{rmi}$. We divide the feature space into three information sources. An initial feature source is based on the shortest path between the concepts mentions, all of which have been proposed for TeGRR in (Melli & al, 2007). We also propose to inherit the concept mention linking features defined in (Melli & al, 2010) for each of the two concept mentions associated to a relation mention candidate. Finally, we also propose features that draw on information from the ontology.

Relation Mention			TeGRR Text-Graph based	Concept Mention (CM) Linking based		Ontology based	label
doc_d	m_i	m_j		CM_a	CM_b		
			Feature Space				T
							F
							...

Table 1 – A high-level representation of training examples of a document's unique concept mention pairs (relation mention candidates). The label assignment procedure and the feature definitions are presented in the two coming subsections.

2.1 Label Assignment

Annotating relations in text is a time consuming process – more so than annotating entities. To overcome the lack of annotated relation mention data, we propose to use the ontology for the labeling decision. For each combination of concept mention pairs the heuristic automatically assign labels according to the following rule. If the concepts in the ontology associated with the relation mention share a direct internal link in the ontology in either direction then the training example is marked as true; otherwise it is labeled as False.

[2] ACE Relation Detection Recognition (RDR) task http://projects.ldc.upenn.edu/ace/annotation/

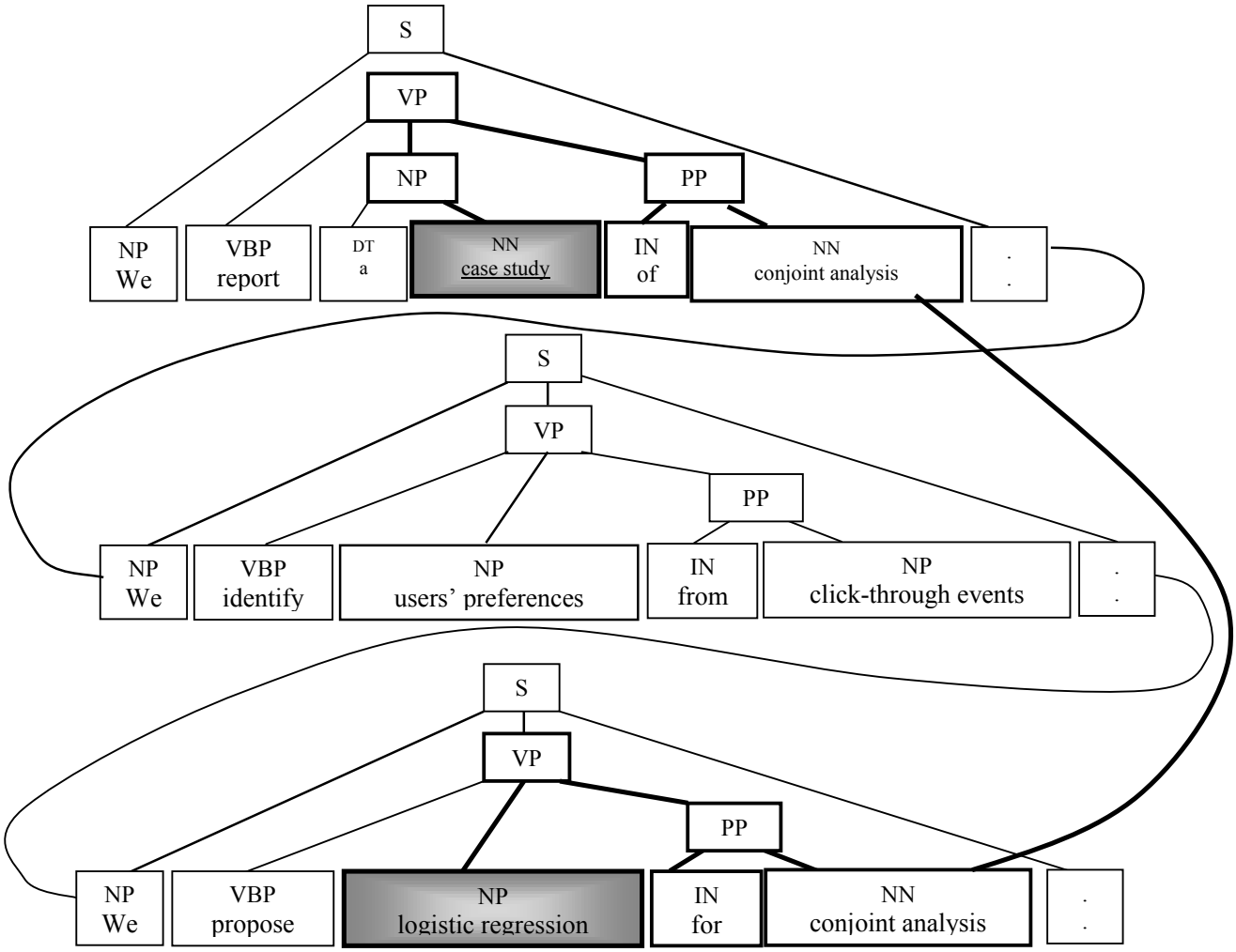
This approach to labeling is similar to the one used by relation mention recognition task such as (Melli & al, 2007). Our proposal in this paper however extends this automatic labeling approach for False example labeling to also automatically label true relation mentions. This approach is more likely to lead to erroneously mislabeled candidates. In many cases, the passages associated with a candidate relation mention that happens to refer to directly linked concepts in the ontology do not substantiate a direct semantic relation. In these cases, after reading the passage, an expert would instead conclude that a direct relation is not implied by the passage and would label the candidate relation mention as False. Alternatively, the heuristic would label some relation mention candidates as False simply because the relation did not yet exist in the ontology; while, upon manual inspection of the passage, the annotator would label the relation as a True candidate.

Despite this appreciation of noise in the generated labels, we hypothesize that this heuristic labeling approach provides a sufficient signal for the supervised classification algorithm to detect many direct relation mentions with sufficient accuracy to be useful in some real-world tasks, such as ontological engineering.

3 Text Graph Representation

The **TeGRR** feature space is based on a graph representation of the document under consideration. The text graph representation is composed of the three types of edges: 1) Intra-sentential edges; 2) Sentence-to-sentence edges; and 3) Co-reference edges.

Figure 1 - An illustration of $SDOI_{RMI}$'s text graph to create feature vectors. The highlighted nodes and path represent the information used for a specific candidate pair assessment.



Intra-sentential edges in a text-graph represent edges between nodes associated with tokens from the same sentence. These edges can vary from being: word-to-word edges, shallow parsing edges, dependency parse tree edges, and phrase-structure parse tree edges. We propose the use the phrase-structure parse tree as the source of intrasentential edges for two reasons. The choice of this data source over the others is the analysis by (Jiang & Zhai, 2007) that suggests that the phrase-structure parse tree is the best single source of information for relation detection. Secondly, all other proposed intra-sentential edge types can be derived, or approximated, from phrase-structure parse trees by means of transformations.

A phrase-structure parse tree is composed of two types of nodes: leaf nodes and internal nodes. Leaf nodes (which map to our external nodes) are labelled with the text token (or concept mention), and with the part-of-speech role. Internal nodes contain the syntactic phrase-structure label.

The text graph in **Figure 1** contains 26 intrasentential edges connecting 12 internal nodes and 19 leaf nodes.

Edges in a text graph can also cross sentence boundaries. The first type of inter-sentential edge to be considered is the "sentence-to-sentence" edge that simply joins an end-of-sentence punctuation node with the first word of the sentence that follows. The intuition for this edge type is that a concept that is mentioned in one sentence can be in a semantic relation with a concept mention in the adjacent sentence, and that the likelihood of it being a relation increases as you reduce the number of sentences between the two entities. The text graph in **Figure 1** contains two sentence-to-sentence edges.

Co-reference Edges

The other source of inter-sentential edges to be considered, also taken from (Melli & al, 2007), are based on concept mentions in the same document that are linked to (co-refer to) the same concept in the ontology. For example if "*hidden-Markov models*" is mentioned in one sentence, "*HMMs*" is mentioned in a subsequent one, and the pronoun "*they*" is used to refer to the concept further on in the document, then coreference edges would exist between "*hidden-Markov models*" and "*HMMs*", and between "*HMM*" and "*they*" (via the Hidden Markov Models concept). The intuition for this edge type is that concept mentions in separate sentences but that are near some coreferent concept mention are more likely to be in a semantic relation than if that co-referent mention did not exist. The text graph in **Figure 1** contains a coreference edge between the mentions of to the Conjoint Analysis Algorithm that were identified by the concept mention identifier and disambiguator described in (Melli & Ester, 2010).

Text-Graph Properties

We describe properties of a text graph used to define **SDOI$_{rmi}$**'s text-graph related features:

1) A text-graph is a connected graph: for every pair of nodes *n* and *v* there is a walk from *n* to *v*
2) A text-graph can be a cyclic graph, and such cycles must involve co-reference edges.
3) A text-graph has at least one shortest path between any two nodes, *n* and *v*, and the number of edges between them is their *distance*.
4) A concept mention m_i is in a *p-shortest path* with concept mention m_j if there are only *p*-1 other concept mentions in a shorter shortest-path relation with m_i. The value of *p* can be interpreted as the rank of the proximity between the two concept mentions, e.g. 1st nearest, 2nd nearest, etc. If two alternate mention pairs are in equal *p-shortest path relation* then both are True for the relation.
5) A path-enclosed subtree is the portion of the syntactic tree enclosed by the shortest-path between two leaf-nodes. This inner portion of a syntactic tree is predictive in relation extraction tasks (Jiang & Zhai, 2007).

4 Relation Mention Identification Features

We begin the definition of the feature space with the text-graph based features that we retain from (Melli & al, 2007). We then proceed to describe the ontology-based features, and conclude with the concept linking features inherited from the previous (concept linking) task.

4.1 Text-Graph based Features

This section describes the features that we directly inherit from TeGRR. We first describe the underlying text graph representation that is then used to define the associated features.

Path-Enclosed Shortest Path Features
From the path-enclosed shortest-path subgraph we identify all distinct subtrees with up to ***e*** edges as proposed in (Jiang & Zhai, 2007) to replicate the convolution-kernel approach of (Haussler, 1999). A feature is created for each possible *neighborhood* in the subgraph, where a neighborhood is defined by a subtrees with ***e*** edges, where ***e*** ranges from zero through to some upper limit on edges: $e \in [0, e_{max}]$. We retain the ***e*** proposed in (Jiang & Zhai, 2007) of $e_{max}=2$. Subtree-based features associated to the subtrees of size zero ($e=0$) simply summarize the number of nodes of a certain content type in either the entire relation mention graph, or one of its pairings. For example, one feature would count the number of **NP** (Noun Phrase) nodes in the relation mention graph, while another feature would count the number of times that the word "*required*" is present. Subtree-based features associated to the subtrees of size $e>0$ represent the number of times that a subgraph with ***e*** edges appears within the subgraph. For example, one feature would count the number of times that the triple IN – PP – NP appears in the graph.

Sentence Count:
This feature informs the classifier about the number of sentences that intervene between concept mentions. For example, the number of intervening sentences between the "case study" and "logistic regression" mention in the relation mention in **Figure 1** is two (2) sentences. This information will help the classifier adjust its predictions based on the separation. Nearer mentions are more likely to be in a relation.

Intervening Concept Mentions:
This set of features informs the classifier about the number of concept mentions that intervene between two concept mention pairs. For example, in **Figure 1** "*conjoint analysis*" is counted as one intervening concept mention between "*case study*" and "*logistic regression*". This information will help the classifier adjust its predictions based on how many other concept mention candidates exist; the greater then number of intervening concept mentions the less likely that a semantic relation between the two concept mentions is being stated.

4.1.1 Concept Mention Linking-based Features

A second source of features that we propose is to include the pair of feature sets for each concept mention defined for concept mention linking (Melli & Ester, 2010). We concatenate the two feature vectors in the following order: the concept mention that appears first in the text, followed by the other concept mention. These features provide signals of the context of each mention, such as even simply what sentence it is locate on. In **Figure 1** for example, the "*case study*" concept mention is located on the first sentence and the closer a mention is to the first sentence may affect the importance of the mention.

4.2 Ontology-based Features

We further propose four features based on information from the ontology – that differ from the ones inherited from the concept-mention linking task. These four features capture information signals from their pairing in the ontology: Shared_Outlinks, Shared_Inlinks, Shortest_gt1-Edge_Distance, and TF-IDF_Concepts_Similarity.

Shared_Outlinks Feature
The Shared_Outlinks feature counts the number of shared concept outlinks. The intuition for this feature is that two concepts that reference many of the same other concepts in the ontology are more likely to be themselves in a direct relation.

Shared_Inlinks Feature
The Shared_Inlinks feature counts the number of shared concept inlinks. The intuition for this feature is that two concepts that are referenced by many of the same other concepts in the ontology are more likely to be themselves in a direct relation.

Shortest1-Edge_Distance Feature
The Shortest1-Edge_Distance feature reports the shortest distance (in the ontology) that is greater than one counts the number of edges that separate

the two concepts. This feature is the one that introduces the risk of giving away the presence of a direct link between the two concepts in the candidate. An edge distance of one (1) versus any other edge distance would be a perfect predictor of the label. However, information about the distance of alternate paths can provide a signal that the two concepts should be (or are) linked.

TF-IDF_Concepts_Similarity Feature

The TF-IDF_Concepts_Similarity feature reports the tf-idf bag-of-words similarity between the two concept descriptions in the ontology. The intuition is similar to that of the "Shared Outlinks" feature: two concepts that reference many of the same words are more likely to be themselves in a relation. Unlike the "Shared Outlinks" feature however, this feature normalizes for very common and uncommon words.

Corpus-based Features

A final source of information for features that we propose is the training corpus itself. As with the corpus-based features for concept linking (Melli & Ester, 2010), the use of cross-validation for performance estimation requires that the document associated with the training record does not inform these features. For this feature, the count is on "other" documents.

4.3 Relation_Mention_Other_Doc_Count Feature

The Relation_Mention_Other_Doc_Count feature counts the number of other documents in the corpus that contain the pair of linked concept mentions. For example, if one other document contains the two linked concept mentions (and thus contains the same candidate relation mention) this feature is set to one (1).

5 Empirical Evaluation of Relation Mention Identification

In this section, we empirically evaluate the performance of the proposed relation-mention identification algorithm: $SDOI_{rmi}$. For this evaluation, we again used the SVMlight[3] package with its default parameter settings, as the underlying supervised classification algorithm. For the syntactic parse trees, we use Charniak's parser[4].

Evaluation Setup

Similar to evaluation of SDOI's two other component algorithms for concept mention identification and linking, we use a leave-one-document-out method on the kdd09cma1 corpus (Melli, 2010). For each unseen document, we predict which of its binary relation mention candidates (with linked concept mentions) already exist in the ontology. Those relations that do not exist in the ontology are proposed candidates for addition to the ontology.

A challenge associated with this task, as found in the concept-mention linking task, is the highly skewed distribution of the labels. In this case, we do not propose a filtering heuristic to change the training data. Instead, we propose an algorithmic change by tuning SVMlight's cost-factor parameter that multiplies the training error penalty for misclassification of positive examples. We set aside three documents to tune the parameter, and based on an analysis to optimize F1 we set the cost-factor to 8.

Table 2 presents some of the key statistics for the kdd09cma1 from the perspective of relation mention candidates. The corpus contains 44,896 relation mention candidates. Of these, which quantifies the task's data skew, only 3.55% of the mention candidates are found in the ontology.

Table 2 – Key statistics of the number of binary relation mentions in the kdd09cma1 corpus, per abstract and for entire corpus. The final row reports the total number of concept pairings where, at the document-level, pairs to the same two concepts are consolidated.

	Binary Relation Mention Candidates	Positive Candidates	Proportion
Minimum (per abstract)	42.0	1.0	0.88%
Average (per abstract)	322.1	11.5	3.86%
Maximum (per abstract)	1,582.0	4.3	12.50%
Entire corpus	44,896.0	1,593.0	3.55%
Entire corpus (only distinct relations)	34,181.0	1,080.0	3.16%

[3] http://svmlight.joachims.org/

[4] ftp://ftp.cs.brown.edu/pub/nlparser/

Baseline Algorithm(s)

The baseline algorithm that we compare $SDOI_{rml}$'s performance against on the relation-mention identification task is an unsupervised co-occurrence-based algorithm that predicts all permutations of linked concept mention pairs regardless of distance between them. This is the baseline algorithm compared against in (Melli & al, 2007, and Shi & al, 2007). We refer to this algorithm as **AllTrue**.

We also include as a baseline a version of $SDOI_{rml}$ with a restricted feature space that contains the features originally proposed for TeGRR.

Intrinsic Performance Analysis

Table 3 presents the results of the leave-one out performance analysis. $SDOI_{rml}$ outperforms the baseline algorithm in terms of precision and F1. The proposed feature space for SDOI also outperforms the original feature space proposed for TeGRR.

Algorithm	Feature Space	Precision	Recall	F1
SDOI	All	18.2%	24.3%	20.8%
	TeGRR	7.7%	41.8%	13.0%
AllTrue		3.7%	100.0%	7.1%

Table 3 – Leave-one-out performance results on the relation mention identification task on the kdd09cma1 corpus (excluding the three tuning abstracts) by SDOI, SDOI with its feature space restricted to those originally proposed for TeGRR, and the AllTrue baseline.

Extrinsic Performance Analysis

We analyze the performance on a real-world usage scenario where an ontology engineer receives the generated list of relation mention candidates predicted as True for being a direct link, which upon inspection of the ontology does not exist. We manually analyzed the top 40 predicted relation mention candidates proposed for insertion into the kddo1 ontology ranked on their likelihood score[5]. **Table 4** reports a snapshot of these relation candidates. Of the 40 candidates 31 (77.5%) were deemed candidates for insertion into the ontology[6]. Given the high proportion of relation candidates worthy of insertion, this result illustrates some benefit to the ontology engineer.

	Binary Relation		
Score	Concept A	Concept B	Document
20.873	Computing System	Algorithm	doi:10.1145/1557019.1557112
…	…	…	…
15.975	Computing System	Algorithm	doi:10.1145/1557019.1557144
23.584	Conditional Probability	Marginal Probabilty	doi:10.1145/1557019.1557130
22.345	Conjoint Analysis	User Preference	doi:10.1145/1557019.1557138
22.075	Optimization Task	Gradient Descent Algorithm	doi:10.1145/1557019.1557129
20.349	Optimization Task	Gradient Descent Algorithm	doi:10.1145/1557019.1557100
21.788	Set	Pattern	doi:10.1145/1557019.1557071
19.849	Set	Pattern	doi:10.1145/1557019.1557077
21.047	Training Dataset	Performance Measure	doi:10.1145/1557019.1557144

Table 4 – A sample of candidate relations (and their source document) with high likelihood score predicted by SDOI as candidates for addition to the kddo1 ontology. The table groups candidates that refer to the same concept pairs.

Boostrapping Experiment

In practice, a common method of applying self-labelled learning is to treat the labelling heuristic as a means to seed a bootstrapped process where subsequent rounds of labelling are based on the most confident predictions by the newly trained model (Chapelle & al, 2006). Generally, evaluations of this approach have assumed high-accuracy seed labels - either from a small manually curated training set, such as in (Agichtein & Gravano, 2000), or with high-accuracy labelling patterns, such as in (Yarowsky, 1995). Each iteration sacrifices some precision for additional recall performance. In our case a bootstrapped process does not begin with high precision to sacrifice, because of our labelling heuristic does not start with high-precision predictions.

However, we performed a bootstrap experiment by iteratively selecting the 10% of relation mentions that were predicted to be True with the highest likelihood score, and then labelled these candidates as True in the subsequent iteration (even if no

[5] We used SVMlight's real-number predictions, and did not boost the selection based on whether more than two documents resulted in predictions for the concept pair.

[6] This task-based result is likely dependent on the maturity of the ontology.

direct link existed in the ontology for the corresponding concept pair).

F1 performance dropped with each iteration. Some analysis can show that this deterioration in performance is unavoidably built into the process: with each iteration the supervised classifier trained models that were based on the increasingly false assumption that True labelled training data were representative of direct links in the ontology. Ensuing models would begin to predict links that were by definition not in the ontology and would thus be evaluated as false positives.
Thus, we again manually inspected the top 40 predicted relations for the first two iterations. The precision dropped after each iteration. After the first iteration, 29 (72.5%) candidates were correct, and after the second iteration, 21 (52.5%) candidates were correct. During the manual review, we observed that predictions in subsequent iterations began to include some of the more common False pairings listed in **Error! Reference source not found.**. Bootstrapping of $SDOI_{rml}$ does not improve the precision of the reported predictions, on the kdd09cma1 benchmark task.

Observations and Conclusion
We conclude with some observations based on the predictions reported in **Table 4** of the leave-one-out evaluation on the kdd09cma1 corpus The table includes some promising candidates for addition to the ontology. For example, because of this experiment we noted that the obvious missing direct relation between a Computing System and an Algorithm[7]. The table also includes a more nuanced missing direct relation missing in the ontology between Conditional Probability and Marginal Probability[8].

Next, we observe that suggested relation mention candidates whose concept pairs are predicted within more than one document, such as Computing System + Algorithm, may be more indicative that the direct relation is indeed missing from the ontology than when only supported by a single document. However, as counter-evidence, some of the repeated pairs in **Table 4** appear to be listed simply due to their frequent occurrence in the corpus. For example, the candidate relation between the concepts of Set and of Pattern may simply be due to documents (abstracts) that often mention "*sets of patterns*". We would not expect the Set concept to be directly linked to every concept in the ontology that can be grouped into a set. This example however does suggest that Pattern + Set may be a common and important concept in the data mining domain to deserve the addition of a Pattern Set concept into the ontology. We note further that very frequent candidates, such as Research Paper + Algorithm, were not predicted; likely because the algorithm recognized that if such a commonplace relation is always false then it likely will be false in a new/unseen document. Thus, there is some evidence that the number of repetitions can indeed signify a more likely candidate. As future work, it would be worthwhile to attempt to train a second classifier that can use the number of referring documents as a feature.

A separate challenge that we observe from the predictions in **Table 4** is illustrated by the Optimization Task + Gradient Descent Algorithm entry. While this seems like a reasonable candidate for addition at first glance, these two concepts are more likely indirectly related via the Optimization Algorithm concept (*an* optimization task *can be solved by an* optimization algorithm; *a* grandient descent algorithm *is an* optimization algorithm.). The resolution of these situations could require additional background knowledge from the ontology, such as relation types, to inform the classifier that in some situations when the parent is linked to the concept then the child is not directly linked to it.

[7] The direct relation can naturally added in both directions "*an ALGORITHM can be implemented into a COMPUTING SYSTEM*" and "*a COMPUTING SYSTEM can implement an ALGORITHM*."
[8] Based on passage "*…assumption made by existing approaches, that the marginal and conditional probabilities are directly related....*" From 10.1145/1557019.1557130 and due to the fact that the two concept descriptions are briefly described in `kddo1`.

References

Eugene Agichtein, and Luis Gravano. (2000). Snowball: Extracting Relations from Large Plain-Text Collections. In: Proceedings of the 5th ACM International Conference on Digital Libraries (DL 2000).

Paul Buitelaar, Philipp Cimiano, Peter Haase, and Michael Sintek. (2009). Towards Linguistically Grounded Ontologies. In: Proceedings of the 6th European Semantic Web Conference (ESWC 2009).

Michael Erdmann, Alexander Maedche, Hans-Peter Schnurr, and Steffen Staab. (2000). From Manual to Semi-automatic Semantic Annotation: About Ontology-Based Text Annotation Tools. In: Proceedings of the COLING 2000 Workshop on Semantic Annotation and Intelligent Content.

David Haussler. (1999). Convolution Kernels on Discrete Structures. Technical Report UCSC-CLR-99-10, University of California at Santa Cruz.

Jing Jiang, and ChengXiang Zhai. (2007). A Systematic Exploration of the Feature Space for Relation Extraction. In: Proceedings of NAACL/HLT Conference (NAACL/HLT 2007).

Gabor Melli. (2010). Concept Mentions within KDD-2009 Abstracts (kdd09cma1) Linked to a KDD Ontology (kddo1). In: Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010).

Gabor Melli, and Martin Ester. (2010). Supervised Identification of Concept Mentions and their Linking to an Ontology. In: Proceedings of CIKM 2010.

Gabor Melli, Martin Ester, and Anoop Sarkar. (2007). Recognition of Multi-sentence n-ary Subcellular Localization Mentions in Biomedical Abstracts. In: Proceedings of the 2nd International Symposium on Languages in Biology and Medicine (LBM 2007).

Mike Mintz, Steven Bills, Rion Snow, Dan Jurafsky. (2009). Distant Supervision for Relation Extraction without Labeled Data. In: Proceedings of ACL 2009.

Sebastian Riedel, Limin Yao, and Andrew McCallum. (2010). Modeling Relations and their Mentions without Labeled Text. In: Proceedings of ECML 2010.

Steffen Staab (editor), and Rudi Studer (editor). (2009). Handbook on Ontologies - 2nd Ed. Springer Verlag.

Victoria Uren, Philipp Cimiano, José Iria, Siegfried Handschuh, Maria Vargas-Vera, Enrico Motta, and Fabio Ciravegna. (2006). Semantic Annotation for Knowledge Management: Requirements and a survey of the state of the art. In: Web Semantics: Science, Services and Agents on the World Wide Web, 4(1).

David Yarowsky. (1995). Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In: Proceedings of the 33rd annual meeting on Association for Computational Linguistics (ACL 1995)