Assessing the Impact of Single and Pairwise Slot Constraints in a Factor Graph Model for Template-based Information Extraction

Hendrik ter Horst[1], Matthias Hartung[1], Roman Klinger[2], Nicole Brazda[3], Hans Werner Müller[3] and Philipp Cimiano[1]

[1] CITEC, Bielefeld University
{hterhors, mhartung, cimiano}@techfak.uni-bielefeld.de
[2] IMS, University of Stuttgart
roman.klinger@ims.uni-stuttgart.de
[3] CNR and Neurology, HHU Düsseldorf
{nicole.brazda, hanswerner.mueller}@uni-duesseldorf.de

Abstract. Template-based information extraction generalizes over standard token-level binary relation extraction in the sense that it attempts to fill a complex template comprising multiple slots on the basis of information given in a text. In the approach presented in this paper, templates and possible fillers are defined by a given ontology. The information extraction task consists in filling these slots within a template with previously recognized entities or literal values. We cast the task as a structure prediction problem and propose a joint probabilistic model based on factor graphs to account for the interdependence in slot assignments. Inference is implemented as a heuristic building on Markov chain Monte Carlo sampling. As our main contribution, we investigate the impact of soft constraints modeled as single slot factors which measure preferences of individual slots for ranges of fillers, as well as pairwise slot factors modeling the compatibility between fillers of two slots. Instead of relying on expert knowledge to acquire such soft constraints, in our approach they are directly captured in the model and learned from training data. We show that both types of factors are effective in improving information extraction on a real-world data set of full-text papers from the biomedical domain. Pairwise factors are shown to particularly improve the performance of our extraction model by up to +0.43 points in precision, leading to an F_1 score of 0.90 for individual templates.

Keywords: Ontology-based Information Extraction; Slot Filling; Probabilistic Graphical Models; Soft Constraints; Database Population

1 Introduction

Initiated by the advent of the distant supervision [13] and open information extraction paradigms [2], the last decade has seen a tendency to reduce information extraction problems to relation extraction tasks. In the latter, the focus is on

extracting binary entity-pair relations from text by applying various types of discriminative classification approaches. We argue that many tasks in information extraction (in particular, when being used as an upstream process for database population) go beyond the binary classification of whether a given text expresses a given relation or not, as they require the population of complex *template structures*. Such templates consist of a number of typed slots to be filled from unstructured text [6]. Following an ontology-based approach [20], we assume that the templates (including slots and the types of their potential fillers) are pre-defined in a given ontology.

We frame template-based information extraction as an instance of a structured prediction problem [17] which we model in terms of a joint probability distribution over value assignments to each of the slots in a template. Subsequently, we will refer to such templates as *schemata* in order to avoid ambiguities. Formally, a schema S consists of typed slots $(s_1, s_2, \ldots, s_n)$. The slot filling task corresponds to the maximum a posteriori estimation of a joint distribution of slot fillers given a document d

$$(s_1, s_2, \ldots, s_n) = \underset{s'_1, s'_2, \ldots, s'_n \in \Phi}{\operatorname{argmax}} P(s_1 = s'_1, \ldots, s_n = s'_n \mid d)\,, \tag{1}$$

where Φ is the set of all possible slot assignments.

Slots in a schema are interdependent, and these dependencies need to be taken into account to avoid incompatible slot assignments. A simple formulation in terms of n binary-relation extraction tasks would therefore be oversimplifying. On the contrary, measuring the dependencies between all slots would render inference and learning intractable. We therefore opt for an intermediate solution, in which we analyze as to what extent measuring *pairwise* slot dependencies helps in avoiding incompatibilities and finally to improve an information extraction model for the task.

We propose a factor graph approach to schema/template-based information extraction which incorporates factors that are explicitly designed to encode such constraints. Our main research interest is therefore to (1) understand whether such constraints can be learned from training data (to avoid the need for manual formulation by domain experts), and (2) to assess the impact of these constraints on information extraction performance.

We evaluate our information extraction model on a corpus of scientific publications reporting the outcomes of pre-clinical studies in the domain of spinal cord injury. The goal is to instantiate multiple schemata to capture the main parameters of each study. We show that both types of constraints are effective, as they enable the model to outperform a naive baseline that applies frequency-based filler selection for each slot.

2 Related Work

Template/Schema-based information extraction dates back to the MUC-4 Shared Task [18] which aimed at extracting instantiations of templates describing terrorist

attacks. More recently, Haghighi et al. [7] focus on corporate acquisition events. Information extraction approaches in this line of research are commonly limited to only one or a fixed set of templates, each of them containing only a comparably small set of slots. Obviously, these assumptions pose severe restrictions to real-world application scenarios. Many tasks in the context of knowledge discovery from scientific literature [8], for instance, require a rich representation of the technical domain of interest, which commonly involves numerous templates with multiple (and possibly hierarchically embedded) slots.

Recent examples of reducing slot filling problems to relation extraction tasks are Riedel et al. [15] with a focus on knowledge base completion, Zhang et al. [21], Adel et al. [1], and Singh et al. [16] in the context of cold-start knowledge base population. While our work also addresses the cold-start problem, our domain of application requires the population of complex ontologically typed schemata. We approach this challenge using undirected probabilistic graphical models which integrate coherence constraints over pairs of slots within a schema. Similar techniques have been proposed for the more shallow problems of HMM-based sequence labeling by Chang et al. [5] and relation extraction by Lopez de Lacalle & Lapata [12]. In line with the latter approach, we aim at inducing constraint knowledge automatically from training data.

Methodologically, our work is similar to collective information extraction with undirected graphical models as proposed by Bunescu et al. [4] or Kluegl et al. [9]; however, these approaches are limited to problems of text segmentation, entity tagging and extraction of individual relations.

As the only precursor of our work towards information extraction in the spinal cord injury domain, Paassen et al. [14] address entity extraction in isolation, *i. e.*, they aim at detecting all entities taking part in a relation, without considering the relation classification task as such.

3 Method

We frame the slot filling task as a joint inference problem in undirected probabilistic graphical models. Our model is a factor graph [11] which probabilistically measures the compatibility of a given textual document d consisting of tokenized sentences χ, a fixed set of entity annotations $\mathcal{A}$, and a to be filled ontological schema S. The schema S is automatically derived from an ontology and is described by a set of typed slots, $S = \{s_1, \ldots, s_n\}$. Let $\mathcal{C}$ denote the set of all entities from the ontology, then each slot $s_i \in S$ can be filled by a pre-defined subset of $\mathcal{C}$ called slot fillers. Further, each annotation $a \in \mathcal{A}$ describes a tuple $\langle t, c \rangle$ where $t = (t_i, \ldots, t_j) \in \chi$ is a sequence of tokens with length ≥ 1 and a corresponding filler type $c \in \mathcal{C}$.

3.1 Factorization of the Probability Distribution

We decompose the overall probability of a schema S into probability distributions over single slot and pairwise slot fillers. Each individual probability distribution

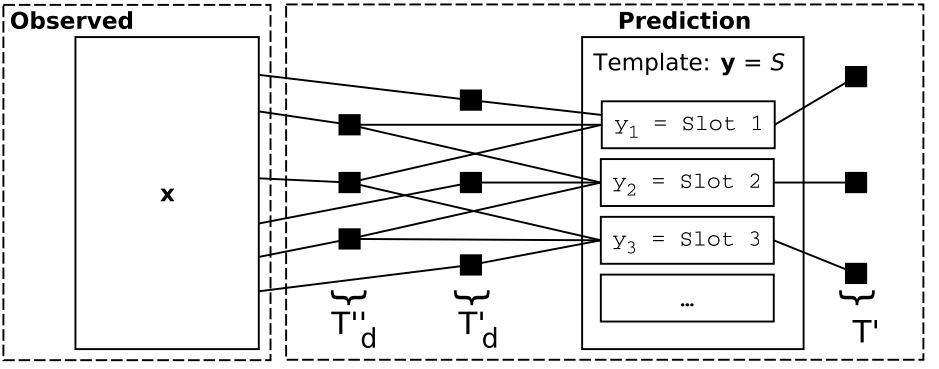

Fig. 1. Factor graph of our model for an exemplary ontological schema S.

is described through factors that measure the compatibility of single/pairwise slot assignments. An unrolled factor graph that represents our model structure is depicted in Figure 1. The factor graph consists of different types of factors that are connected to subsets of variables of $\boldsymbol{y} = \{y_0, y_1, \ldots, y_n\}$ and of $\boldsymbol{x} = d = \{\chi, \mathcal{A}\}$, respectively. We distinguish three factor types by their instantiating factor graph template $\{T', T'_d, T''_d\} \in \mathcal{T}$: (i) **Single slot factors** $\Psi'(y_i) \in T'$ that are solely connected to a single slot y_i, (ii) **Single slot+text factors** $\Psi'(\boldsymbol{x}, y_i) \in T'_d$ that are connected to a single slot y_i and $\boldsymbol{x}$, (iii) **Pairwise slot+text factors** $\Psi''(\boldsymbol{x}, y_i, y_j) \in T''_d$ that are connected to a pair of two slots y_i, y_j and $\boldsymbol{x}$.

The conditional probability $P(\boldsymbol{y} \mid \boldsymbol{x})$ of a slot assignment $\boldsymbol{y}$ given $\boldsymbol{x}$ is then

$$P(\boldsymbol{y}|\boldsymbol{x}) \quad = \quad \frac{1}{Z(\boldsymbol{x})} \prod_{y_i \in S} \left[\Psi'(y_i) \;\cdot\; \Psi'(\boldsymbol{x}, y_i) \right] \prod_{y_i \in S} \prod_{y_j \in S} \left[\Psi''(\boldsymbol{x}, y_i, y_j) \right], \quad (2)$$

where $Z(\boldsymbol{x})$ denotes the partition function and all factors are formulated as $\Psi(\cdot) = \exp(\langle f_T(\cdot), \theta_T \rangle)$ with sufficient statistics $f_T(\cdot)$ and parameters θ_T ($T \in \mathcal{T}$ and $\Psi \in \{\Psi', \Psi''\}$).

3.2 Inference and Learning

We perform Markov chain Monte Carlo (MCMC) sampling to approximate a posterior distribution, while sharing the factorization properties as defined by the factor graph [10]. We learn the parameters via SampleRank [19].

Ontological Sampling The generation of proposal states in our MCMC sampling procedure follows the idea of Gibbs sampling, mainly applying atomic changes to slots. The initial state s_0 in our exploration is empty, thus $\boldsymbol{y} = (\varnothing)$. A set of potential successors is generated by a proposal function changing a slot by either deleting an already assigned value or changing the value to another slot filler. The state with the highest probability s_{t+1} is chosen as successor state only if $p(s_{t+1}) > p(s_t)$. The inference procedure stops, iff $s_{t+3} = s_t$.

Objective Function Given a predicted assignment $\boldsymbol{y'}$ of all slots in schema type $\hat{S}$ and a set $\mathcal{G}$ of instantiated schemata of type $\hat{S}$ from the gold standard, the training objective is

$$\max_{\boldsymbol{y}^* \in \mathcal{G}} \mathrm{F}_1(\boldsymbol{y}^*, \boldsymbol{y'}) \,, \tag{3}$$

where F_1 is the harmonic mean of precision and recall, based on the overlap of assigned slot values between $\boldsymbol{y'}$ and $\boldsymbol{y}^*$.

3.3 Factors and Constraints

At the core of our model are features that encode soft constraints to be learned from training data. In general, these constraints are intended to measure the compatibility of slot fillers within a predicted schema. Such soft constraints are designed through features that are described in the following.

Single-slot constraints in template T' We include features which measure common, acceptable fillers for single slots with numerical values. Given filler annotations $\{a_i = \langle v, c \rangle\}$ of slot y_i, the model can learn individual intervals for different types of fillers such as temperature (−10–40), or weight (200–500), for example. For that, we calculate the average μ and standard deviation σ for each particular slot based on the training data. For each slot s_i in schema S, a boolean feature $f^{s_i}_{\sigma=n}$ is instantiated for each $n \in \{0, \ldots, 4\}$, indicating whether the value y_i is within n standard deviations σ_{s_i} of the corresponding mean μ_{s_i}. To capture the negative counterpart, a boolean feature $f^{s_i}_{\sigma>n}$ is instantiated likewise:

$$f^{s_i}_{\sigma=n}(y_i) = \begin{cases} 1 & \text{iff } \left\lceil (\frac{v-\mu_{s_i}}{\sigma_{s_i}}) \right\rceil = n \\ 0 & \text{otherwise.} \end{cases} \qquad f^{S_i}_{\sigma>n}(y_i) = \begin{cases} 1 & \text{iff } \left\lceil (\frac{v-\mu_{s_i}}{\sigma_{s_i}}) \right\rceil > n \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

In this way, the model learns preferences over possible fillers for a given slot which effectively encode soft constraints such as "the weight of rats typically scatters around a mean of 300 gram by two standard deviations of 45 gram".

Pairwise Slot Constraints in T''_d In contrast to single-slot constraints, pairwise constraints are not limited to slots with real-valued fillers. Soft constraints on slot pairs are designed to measure the compatibility and (hidden) dependencies between two fillers, e.g., the dependency between the dosage of a medication and its applied compound, or between the gender of an animal and its weight. This is modeled in terms of their linguistic context and textual locality, as discussed in the following.

We assume that possible slot fillers may be mentioned multiple times at various positions in a text. Therefore, given a pair of slots (s_i, s_j), we define λ as an aggregation function that returns the subset of annotations $\lambda(s_i) = \{a = \langle t, c \rangle \in \mathcal{A} \mid a(c) = s_i(c)\}$. We measure the locality of two slots in the text by the minimum distance between two sentences containing annotations for the

corresponding slot fillers. A bi-directional distance for two annotations is defined as $\delta(a_k, a_l) = |\text{sen}(a_k) - \text{sen}(a_l)|$ where *sen* denotes a function that returns the sentence index of an annotation. For each $n \in \{0, \ldots, 9\}$, a boolean feature $f_{\delta=n}$ is instantiated as:

$$f_{\delta=n}^{s_i,s_j}(y_i, y_j) = \begin{cases} 1 & \text{iff } n = \min_{a_k \in \lambda(y_i), a_l \in \lambda(y_j)} \delta(a_k, a_l) \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

To capture the linguistic context between two slot fillers y_i and y_j, we define a feature $f_{\pi_n}^{s_i}(y_i, y_j)$ that indicates whether a given $\mathcal{N}$-gram $\pi_n \in \pi$ with $1 < \mathcal{N} \leq 3$ occurs between the annotations $a_k \in \lambda(y_i)$ and $a_l \in \lambda(y_i)$ in the document.

Textual Features in T' and T'_d Given a single slot s_i with filler y_i and the aggregated set of all corresponding annotations $\lambda(y_i)$, we instantiate three boolean features for each annotation $a \in \lambda(y_i)$ as follows.

Let $\text{L}_s(l_{y_i}, a(t))$ be the Levenshtein similarity between the ontological class label l_{y_i}, and the tokens of an annotation $a(t)$. Two boolean features $f_{bin(s_{max})<\Delta}(y_i)$ and $f_{bin(s_{max})\geq\Delta}(y_i)$ are computed as:

$$f_{bin(s_{max})<\Delta}(y_i) = \begin{cases} 1 & \text{iff } b < \Delta \\ 0 & \text{otherwise.} \end{cases} \quad f_{bin(s_{max})\geq\Delta}(y_i) = \begin{cases} 1 & \text{iff } b \geq \Delta \\ 0 & \text{otherwise.} \end{cases}, \quad (6)$$

where $b = \text{bin}(s_{max})$ is the discretization of the maximum similarity s_{max} into intervals of size 0.1, and

$$s_{max} = \max_{a \in \lambda(y_i)} L_s(l_{y_i}, a(t)) \text{ with } L_s = 1 - \frac{\text{levenshtein}(l_{y_i}, a(t))}{\max(\text{len}(l_{y_i}), \text{len}(a(t)))}. \quad (7)$$

Finally, we instantiate features $f_{\pi_k\ context}^{s_i}(y_i)$ and $f_{\pi_k\ within}^{s_i}(y_i)$, indicating whether an $\mathcal{N}$-gram π_k occurs in the context (before or after) or within any annotation of slot y_i.

4 Database Population in the Spinal Cord Injury Domain

4.1 Problem Description

We address the problem of ontology-based information extraction in a slot filling setting as a prerequisite for cold-start database population. The extraction task comprises multiple schemata of different types, each of them being provided by a domain ontology and containing multiple slots. Each slot in a schema needs to be filled either by a literal from the input document or by a class from the ontology, depending on whether it is derived from a data-type or object-type property (cf. Figure 2).

We consider slot filling as a document-level task, i.e., entities filling the slots of a particular schema may be dispersed across the entire text. In addition, each literal or ontological category can, in principle, fill multiple slots of the appropriate type. We approach the task in a supervised machine learning approach; supervision is available at the document level in terms of fully instantiated gold schemata without direct links between slot fillers and text mentions.

4.2 Application Context

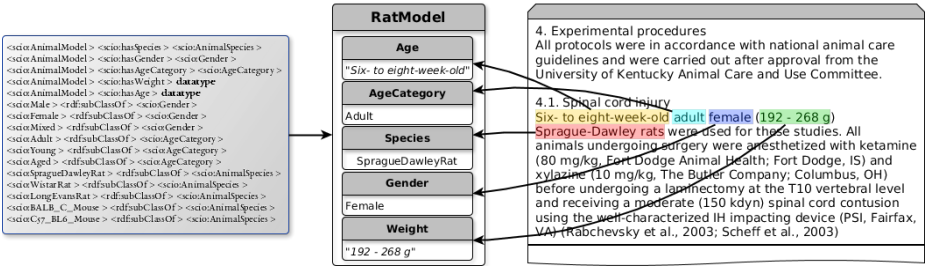

Fig. 2. Information extraction workflow: Domain concepts and associated slots are defined in a *domain ontology* (left) and transformed into *schema structures* (middle) which are automatically populated from text (right) by the *slot filling model*.

Our work in the PSINK project[4] aims at information extraction from full-text scientific publications on pre-clinical experiments in the spinal cord injury domain. The results of the extraction process (i.e., fully instantiated schemata as shown in Fig. 2) will be made accessible in a comprehensive database in order to foster translation from pre-clinical trials into clinical therapeutic concepts bearing the potential to induce neuronal regeneration in human patients suffering from spinal cord injuries.

This information extraction task is an instance of the problem described in Section 4.1, with the extraction schema being derived from the specifically designed Spinal Cord Injury Ontology (cf. Section 4.3 below).

4.3 Ontology and Data Set

Spinal Cord Injury Ontology (SCIO) Pre-clinical trials in the spinal cord injury domain follow strict methodological patterns. Experimental protocols and the main outcomes of pre-clinical studies on spinal cord injury are formally represented in SCIO [3]. In total, the ontology contains more than 500 classes and approx. 80 properties (slots). SCIO top-level classes defining the schema types are AnimalModel, InjuryModel, Treatment, InvestigationMethod and Result. Slots are either object-type properties which can be filled by a SCIO class, or data-type properties which are filled with free text. For example, Fig. 2 (left and middle part) presents the AnimalModel class along with its predefined slots: *ageCategory*, *gender* and *species* are object-type properties; *age* and *weight* are data-type properties.

[4] http://www.psink.de

Annotated Data Set The annotated data set was created by two SCI experts who annotated 25 full-text scientific papers from the SCI literature. Annotations were provided at the level of fully instantiated schemata per document, using the set of top-level classes in SCIO and their corresponding properties as annotation schema. The entire annotation process comprises three steps: (i) mention identification, (ii) entity recognition (in case of data-type properties) and linking (object-type properties), (iii) schema instantiation, and (iv) filling the slots of an instantiated schema with an appropriate entity. The latter steps are due to the fact that the cardinality of schemata of a particular type per document is unknown a priori, and multiple schemata may share individual slot fillers. The following example shows a sentence that describes two instantiations of an AnimalModel schema which share the slot fillers *species* (SpragueDawleyRat) and *ageCategory* (Adult): *"A total of 39 Sprague-Dawley rats were used for these experiments: adult males (285-330 g) and females (192-268 g)."*

Inter-annotator agreement at the level of fully instantiated schemata in terms of F_1 score between annotators amounts to 0.93 for AnimalModel, 0.79 for Injury, 0.77 for Treatment and 0.65 for InvestigationMethod.

5 Experiments

In the following section, we describe our experimental settings, the evaluation metrics and results. Model performances are independently reported for four SCIO schemata: AnimalModel, Injury, Treatment, and InvestigationMethod (cf. Section 4.3). As a preprocessing step, we apply symbolic entity recognition in order to generate annotations $\mathcal{A}$. The regular expressions used are automatically generated from ontology class labels. In case of data-type properties (e.g., weight of an animal), regular expressions are manually created.

5.1 Experimental Settings

The system is evaluated in a 6-fold cross validation on the complete data set. In all experiments, we restrict the complexity of the schemata to first-order slots, i.e., ontological properties that are directly connected to their respective domain class. In the current approach, we are not aiming at predicting the correct number of instantiations per schema type. Thus, our system is restricted to fill a single schema of each type per document, even if it contains multiple instances of the same schema type (e.g., multiple Treatments).

With respect to this restriction, we report the evaluation results for both (i) *Full Evaluation* (taking the actual number of gold schemata into account), and (ii) *Best Match Evaluation* (comparing the predicted schema to the best matching gold schema).

Further, we report the performance for two different models, in order to investigate the relative impact of single-slot constraints vs. pairwise slot constraints. In the *pairwise slot filling* (PSF) model, the inference and the factor graph is based on the joint assignment of slot pairs, whereas in *single slot filling* (SSF) model, all slots are independently filled.

Evaluation Metrics We report model performances as macro precision, recall and harmonic F_1. Given a document with a set of gold schemata $\mathcal{G}$ of type $S = \{s_0, \dots s_n\}$ and the predicted schema p, the comparison is always based on the best assignment $g' = \text{argmax}_{g \in \mathcal{G}} F_1(p, g)$. For the computation of the overall F_1 score, we convert all ontological schemata into sets of slot-filler pairs with $p = \{s'_0 = c_j, \dots, s'_n = c_k\}$ and $\mathcal{G} = \{g^0, \dots, g', \dots, g^l\} = \{(s^0_0 = c_a, \dots, s^0_n = c_b), \dots, (s'_0 = c_c, \dots, s'_n = c_d), \dots, (s^l_0 = c_e, \dots, s^l_n = c_f)\}$. The overall F_1 score is calculated based on the two sets of p and $\mathcal{G}$. We define a true positive (tp) as a slot-filler pair that are in both p and $\mathcal{G}$, a false positive (fp) as a pair that is in p but not in $\mathcal{G}$, and a false negative (fn) as a pair that is in $\mathcal{G}$ but not in p. During the *Best Match Evaluation*, we set $\mathcal{G} = \{g'\}$.

Most Frequent Filler Baseline We compare the performance of our models in all settings against a naive but plausible baseline. Following the intuition that important information is mentioned in a higher frequency than non-important information, a slot is always filled with the filler that has the highest annotation frequency. In the following, we refer to this procedure as Most Frequent Filler (MFF) baseline.

5.2 Results

In the following, we describe the evaluation results for all experiments. First, we compare the performance in the *Full Evaluation* vs. *Best Match Evaluation* settings. In the former setting, we expect a rather low recall due to the restriction of predicting exactly one schema per type. This leads to many false negatives, as multiple instances of the same type cannot be fully covered yet. Hence, we hypothesize a significant increase in recall in the *Best Match Evaluation* setting. By comparing the predicted schema to the best match only, we investigate whether the low recall is due to the large amount of missing schemata. If so, this would indicate that our model is able to select the correct slot fillers among a huge set of possible candidates. The performance of all models in both settings is reported in Table 1.

Full Evaluation Results The results show a strong recall of our baseline model with a distinct lack in precision. The baseline yields the highest recall among all models and schema types except for the AnimalModel (0.55 for baseline vs. 0.90 for SSF/PSF). Compared to the SSF model, we notice a considerable increase in precision in all schema types which is most pronounced in the InvestigationMethod (+0.64). The increase in precision for the three other schemata are between +0.24 and +0.36. Comparing the PSF to the SSF model, we observe further strong improvements in precision and slight improvements in recall. The PSF model clearly outperforms the baseline for the AnimalModel with an increase in F_1 of +0.39, the Injury +0.12, and the InvestigationMethod with +0.14. Despite the precision being increased by +0.46 in the Treatment, the baseline shows a higher F_1 score in this configuration (+0.03), due to a drop in recall by −0.10.

Table 1. Performance of Most Frequent Filler Baseline (MFF) vs. Single Slot Filler (SSF) and Pairwise Slot Filler (PSF) models in the *Full Evaluation* (full) and *Best Match* (best) setting.

		MFF			SSF			PSF		
		P	R	F_1	P	R	F_1	P	R	F_1
Animal	full	0.48	0.55	0.51	0.84	0.90	0.86	**0.91**	**0.90**	**0.90**
Model	best	0.48	0.57	0.52	0.84	**1.00**	0.91	**0.91**	**1.00**	**0.95**
Injury	full	0.28	**0.38**	0.31	0.52	0.22	0.31	**0.77**	0.30	**0.43**
	best	0.28	**0.43**	0.33	0.52	0.29	0.35	**0.77**	0.40	**0.50**
Treat-	full	0.39	**0.26**	**0.30**	0.70	0.16	0.26	**0.87**	0.16	0.27
ment	best	0.39	**0.74**	0.51	0.70	0.63	0.65	**0.87**	0.63	**0.73**
Invest.	full	0.36	**0.45**	0.36	**1.00**	0.39	0.50	**1.00**	0.39	**0.50**
Method	best	0.36	0.98	0.52	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**

Best Match Evaluation Results In this setting, we further investigate the recall performance of our models compared to the previously discussed *Full Evaluation* results. As we only remove uncaptured schema instances from $\mathcal{G}$ (cf. Section 5.1), the precision remains the same. All models show an overall increase in recall for all schema types. With respect to the PSF model, we can see a strong increase in recall for InvestigationMethod by +0.61 and for Treatment by +0.47. Further, slight increases by +0.10 and +0.07 can be observed for AnimalModel and Injury, respectively. Similar observations can be made for the SSF model.

5.3 Discussion

Comparing the baseline model with the SSF model, we notice a very strong increase in precision in combination with a slight drop in recall. This positive trend in precision is continued when considering the PSF model. Further, the results show a positive impact of pairwise over single-slot constraints on recall.

The high recall of 0.90 for the AnimalModel in the full evaluation is mainly due to a low number (1 to 2) of instances per schema type in each document. The fact that there is no difference in the performance of the SSF and SPF models for the InvestigationMethod suggests a strong slot independence, so that pairwise slot constraints do not have a big impact in this particular case. The low increase in recall between the two evaluation settings for the Injury suggests difficulties for this schema. In contrast, the recall increase for the Treatment schema from 0.16 to 0.63 clearly shows that most of the errors are due to a large number of schema instances per document.

Overall, the results show that our system is often able to select the correct set of slot fillers for a schema, even from a huge set of possible schemata and their corresponding slot filler candidates.

6 Conclusions and Outlook

We have investigated the impact of single and pairwise slot constraints in a factor graph model for schema/template-based information extraction. We found that both types of constraints increase the overall performance of the slot filling model, as they are able to capture soft slot restrictions (for single slots) and (hidden) slot dependencies (for pairwise slots). We were able to show that, compared to a plausible baseline, both constraint types are effective, with pairwise constraints outperforming the single slot constraints. For future work, we plan to extend the current model by incorporating further constraints beyond the current restriction to pairwise slot dependencies, with a potential culmination in a fully joint model.

Our approach was developed in the context of the PSINK project which aims at populating a database for pre-clinical studies in the spinal cord injury domain. Our proposed approach lays the groundwork for this task by instantiating ontologically defined schemata and filling them from unstructured text. In future work, we plan to extend our approach to more complex schemata covering the entire ontology. This raises further research questions that need to be answered, such as *How to determine the actual number of instances per schema type?* and *How to efficiently explore recursively nested properties within complex schemata?*

Acknowledgments

This work has been funded by the Federal Ministry of Education and Research (BMBF, Germany) in the PSINK project (project numbers 031L0028A/B).

References

1. Adel, H., Roth, B., Schütze, H.: Comparing convolutional neural networks to traditional models for slot filling. In: Proceedings of NAACL/HLT. pp. 828–838 (2016)
2. Banko, M., Cafarella, M., Soderland, S., Broadhead, M., Etzioni, O.: Open information extraction from the web. In: Proceedings of IJCAI. pp. 2670–2676 (2007)
3. Brazda, N., ter Horst, H., Hartung, M., Wiljes, C., Estrada, V., Klinger, R., Kuchinke, W., Müller, H.W., Cimiano, P.: SCIO: An Ontology to Support the Formalization of Pre-Clinical Spinal Cord Injury Experiments. In: Proc. of the 3rd JOWO Workshops: Ontologies and Data in the Life Sciences (2017)
4. Bunescu, R., Mooney, R.: Collective information extraction with relational markov networks. In: Proceedings of ACL. pp. 438–445 (2004)
5. Chang, M.W., Ratinov, L., Roth, D.: Structured learning with constrained conditional models. Machine Learning 88(3), 399–431 (6 2012)
6. Freitag, D.: Machine learning for information extraction in informal domains. Machine Learning 39(2-3), 169–202 (2000)
7. Haghighi, A., Klein, D.: An entity-level approach to information extraction. In: Proceedings of ACL. pp. 291–295 (2010)
8. Henry, S., McInnes, B.: Literature based discovery: Models, methods, and trends. J Biomed Inform 74, 20–32 (2017)

9. Kluegl, P., Toepfer, M., Lemmerich, F., Hotho, A., Puppe, F.: Collective information extraction with context-specific consistencies. In: Proceedings of ECML/PKDD. pp. 728–743 (2012)
10. Koller, D., Friedman, N.: Probabilistic Graphical Models. Principles and Techniques. MIT Press (2009)
11. Kschischang, F.R., Frey, B.J., Loeliger, H.A.: Factor Graphs and Sum Product Algorithm. IEEE Transactions on Information Theory 47(2), 498–519 (2001)
12. Lopez de Lacalle, O., Lapata, M.: Unsupervised Relation Extraction with General Domain Knowledge. In: Proceedings of EMNLP. pp. 415–425 (2013)
13. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: Proc. of ACL. pp. 1003–1011 (2009)
14. Paassen, B., Stöckel, A., Dickfelder, R., Göpfert, J.P., Brazda, N., Kirchhoffer, T., Müller, H.W., Klinger, R., Hartung, M., Cimiano, P.: Ontology-based Extraction of Structured Information from Publications on Preclinical Experiments for Spinal Cord Injury Treatments. In: Proc. of the 3rd Workshop on Semantic Web and Information Extraction (SWAIE). pp. 25–32 (2014)
15. Riedel, S., Yao, L., McCallum, A., Marlin, B.M.: Relation extraction with matrix factorization and universal schemas. In: Proceedings of NAACL/HLT. pp. 74–84 (2013)
16. Singh, S., Yao, L., Belanger, D., Kobren, A., Anzaroot, S., Wick, M., Passos, A., Pandya, H., Choi, J.D., Martin, B., McCallum, A.: Universal Schema for Slot Filling and Cold Start: UMass IESL at TACKBP 2013. In: Proc. of TAC-KBP (2013)
17. Smith, N.A.: Linguistic Structure Prediction. Morgan and Claypool (2011)
18. Sundheim, B.M.: Overview of the fourth message understanding evaluation and conference. In: Proceedings of MUC. pp. 3–21 (1992)
19. Wick, M., Rohanimanesh, K., Culotta, A., McCallum, A.: SampleRank. Learning Preferences from Atomic Gradients. In: Proc. of the NIPS Workshop on Advances in Ranking. pp. 1–5 (2009)
20. Wimalasuriya, D.C., Dou, D.: Ontology-based information extraction: An introduction and a survey of current approaches. Journal of Information Science 36(3), 306–323 (2010)
21. Zhang, Y., Zhong, V., Chen, D., Angeli, G., Manning, C.D.: Position-aware attention and supervised data improve slot filling. In: Proc. of EMNLP. pp. 35–45 (2017)