Event Extraction using Structured Learning and Rich Domain Knowledge: Application across Domains and Data Sources

EINAT MINKOV, University of Haifa

We consider the task of record extraction from text documents, where the goal is to automatically populate the fields of target relations, such as scientific seminars or corporate acquisition events. There are various inferences involved in the record-extraction process, including mention detection, unification, and field assignments. We use structured learning to find the appropriate field-value assignments. Unlike previous works, the proposed approach generates feature-rich models that enable the modeling of domain semantics and structural coherence at all levels and across fields. Given labeled examples, such an approach can, for instance, learn likely event durations and the fact that start times should come before end times. While the inference space is large, effective learning is achieved using a perceptron-style method and simple, greedy beam decoding. A main focus of this article is on practical aspects involved in implementing the proposed framework for real-world applications. We argue and demonstrate that this approach is favorable in conditions of data shift, a real-world setting in which models learned using a limited set of labeled examples are applied to examples drawn from a different data distribution. Much of the framework's robustness is attributed to the modeling of domain knowledge. We describe design and implementation details for the case study of seminar event extraction from email announcements, and discuss design adaptations across different domains and text genres.

Categories and Subject Descriptors: I.2.8 [**Machine Learning**]; I.2.9 [**Natural Language Processing**]: Text Analysis

General Terms: Algorithms, Experimentation

Additional Key Words and Phrases: Information extraction, template filling, structured learning, perceptron, beam search, domain knowledge

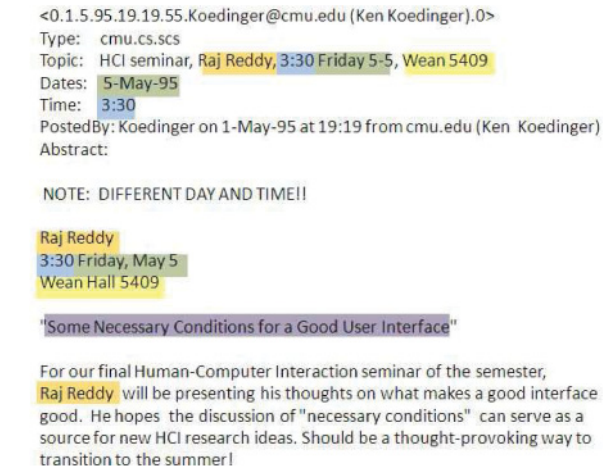
ACM Reference Format:
Einat Minkov. 2015. Event extraction using structured learning and rich domain knowledge: Application across domains and data sources. ACM Trans. Intell. Syst. Technol. 7, 2, Article 16 (December 2015), 34 pages.
DOI: http://dx.doi.org/10.1145/2801131

1. INTRODUCTION

Information extraction (IE) systems aim at recovering structured information from text [Mooney and Bunescu 2005; Sarawagi 2008]. Record extraction, or *template filling*, is an IE task where the goal is to populate the fields of a database record from a given text: for example, to extract the attributes of a job posting [Califf and Mooney 2003] or the details of scientific seminar events [Freitag and McCallum 2000]. Databases extracted from text by means of template filling may be readily used by downstream applications, supporting efficient retrieval and high-level processing of this data. In this article, we consider the extraction of *event* details from textual announcements. Structured

Authors' addresses: E. Minkov, Dept. of Information Systems, University of Haifa, Haifa, Israel, 31905; email: einatm@is.haifa.ac.il.

DOI: http://dx.doi.org/10.1145/2801131



Date	5/5/1995
Start Time	3:30PM
Location	Wean Hall 5409
Speaker	Raj Reddy
Title	Some Necessary Conditions for a Good User Interface
End Time	–

Fig. 1. An example email from the CMU seminar announcement corpus and the corresponding filled *seminar event* template. Field mentions are highlighted in the text, grouped by color.

event data can be highly valuable for location-based systems. Given information about users' whereabouts on one hand, and a database of event details on the other hand, such systems can inform their users about relevant events that take place nearby. Consider, for example, location-tracking systems inside the facilities of a university or workplace [Park et al. 2010]. Given a user profile and an up-to-date dataset of scheduled scientific seminars, it is desired to recommend seminars of interest to the user [Minkov et al. 2010] that take place nearby.

Figure 1 demonstrates the extraction of a *seminar* event from an email announcement, included in the CMU seminar announcement corpus [Freitag and McCallum 2000]. Here, the fields of the target template include the *date* on which the seminar is scheduled to take place, the planned *start time* and *end time*, the *speaker's name*, the assigned *location*, and the seminar's *title*. The template-filling process consists of the following main inference steps. First, *mention detection* is performed, having text spans that describe field values identified. The relevant field mentions are highlighted in Figure 1. Due to inherent redundancy, field values typically appear multiple times in the text, possibly in different forms. These mentions need to be unified; the lexical variants should then be normalized to provide the final extraction per field. For example, in Figure 1, the *location* mentions "Wean 5409" and "Wean Hall 5409" refer to the same conference room, and should be unified. Similarly, the mention "3:30" appears three times; we must infer that the starting time is 3:30PM, and that the end time is never explicitly mentioned. The normalized values per field are shown in the populated template at the bottom of Figure 1.

So far, researchers have mostly focused on learning tagging models for mention detection, which can be difficult to aggregate into a full template extraction, or on learning template field value extractors in isolation and with no reasoning across different fields of the same template. The approach described here generates coherent records, modeling rich relational domain knowledge. Figure 2 gives a general description of the

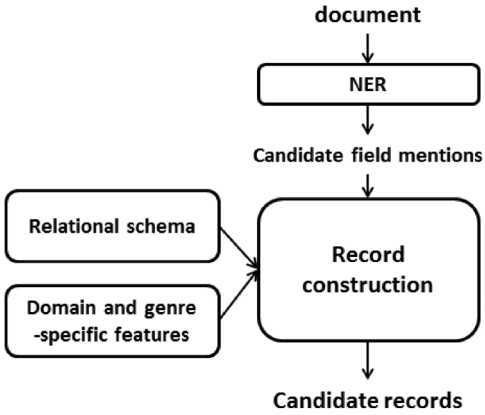


Fig. 2. A general overview of the proposed approach. Given candidate field mentions extracted from the given document using NER techniques, these mentions are organized into candidate record structures. The record construction process is guided by a relational schema of the target domain, as well as features that describe rich domain- and genre-specific phenomena, including high-level regularities across fields.

approach. Given candidate field value mentions, extracted from the input document using complimentary named entity recognition (NER) techniques, relevant mentions are identified and mapped into the slots of the target relation. Record construction is guided by a detailed relational schema of the target domain, as well as domain- and genre-specific regularities encoded as features. Concretely, there are several novel aspects of the proposed approach compared with past works:

—We model the target template as a structured relational entity, embedded within an extended relational schema. The proposed template-filling process is hierarchical, following the relational schema. This allows us to represent fine-grained domain semantics.
—We apply a discriminative, feature-rich, learning framework that can reason about value assignments at template level. Such an approach allows us to model and learn the likely lengths for events or the fact that start times should come before end times, for example.
—The approach is scalable, using structured a perceptron-style algorithm [Collins 2002] for learning and a greedy beam decoding procedure for inference.

The main advantage of the proposed framework is the effective modeling of semantic and structural coherence in the template-filling process. We have previously shown that this framework yields state-of-the-art performance on two benchmark event extraction datasets [Minkov and Zettlemoyer 2012]. In this article, our main emphasis is on assessing the benefit alongside the cost expected in applying this framework in real-world settings. In particular, we address the following questions:

(1) To what extent is the proposed approach effective in practical settings, in which models learned using limited labeled data are applied to instances drawn from a different data distribution?

We explore this question through a set of experiments on seminar extraction. Having learned extraction models using the benchmark CMU seminar announcements dataset, these models are applied to a new set of seminar announcements, published on MIT's seminars distribution list [Minkov et al. 2010].[1] It is shown that, while performance

[1]The annotated MIT seminar announcement set is available upon request.

degrades on such new data, as one may expect due to a drift in lexicon and document layout conventions, the modeling of domain semantics and global coherence using our approach shows better generalization in these settings.

(2) Encoding useful task-specific knowledge requires human intervention. Another question of practical importance is this: What adaptations need to take place in order to apply the framework across different knowledge domains or text genres?

This article includes a thorough description of design and implementation details concerning the application of the framework to the task of seminar extraction, considering semistructured text like email. Additional discussion is dedicated to adaptations performed in applying the framework to another task, namely, the extraction of corporate acquisition events from news articles. This latter task requires the modeling of different domain knowledge and adaptation across text genres.

Finally, we examine the impact of design choices on performance and scalability. Specifically, tuning the search beam size involves a trade-off between extraction performance and computational cost. This trade-off is evaluated empirically, showing the effect of the beam size on extraction performance versus runtimes.

The article is structured as follows. A discussion of related work is presented in Section 2. We define the extended relational settings of the template-filling problem in Section 3. The learning and inference algorithms are outlined in Section 4. Section 5 describes the application of the framework to the task of seminar event extraction from email announcements, including detailed evaluation results on the benchmark CMU seminar-announcement dataset. Section 6 presents the results of applying the models learned using the CMU dataset to another dataset of MIT seminar events. Section 7 discusses the application of the framework across domain and genre, to corporate event extraction from newswire. Scalability considerations are quantitatively evaluated and discussed in Section 8. Section 9 includes our conclusions and suggestions of future research directions.

2. RELATED WORK

Research on the task of template filling has focused on the extraction of field-value mentions from the underlying text. Typically, these values are extracted based on local evidence, having the most likely entity assigned to each slot [Roth and Yih 2001; Califf and Mooney 2003; Finn 2006; Siefkes 2008]. There has been little research effort toward a comprehensive approach that includes mention unification and considers the structure of the target relational schema to create semantically valid outputs.

Haghighi and Klein [2010] presented a generative semisupervised approach for template filling. In their model, slot-filling entities are first generated, and entity mentions are then realized in text. Thus, their approach performs coreference at slot level. In addition to proper nouns (named entity mentions) that are considered in this work, they also account for nominal and pronominal noun mentions. Their model is hierarchical in the sense that entities share the properties of their super-class entity. This article presents a discriminative approach to the template-filling problem. An advantage of a discriminative framework is that it allows the incorporation of rich and possibly overlapping features. In addition, we enforce label consistency and semantic coherence at record level.

Other related works perform structured relation discovery for different settings of information extraction. In *open information extraction*, the goal is to automatically construct ontologies of world knowledge from free text on the Web, typically extracting facts about the relations between entity pairs [Etzioni et al. 2008]. In this scenario, entities and relations may be inferred jointly [Roth and Yih 2002; Yao et al. 2011], in which case the identified relations must agree with the entity types linked by them;

for example, a *born-in* relation requires a *person* and a *place* as its arguments. In addition, extracted relations may be required to be consistent with an existing ontology [Alani et al. 2003; Carlson et al. 2010]. Compared with open IE, the task of template filling aims at populating a target relational schema that includes a larger number of attributes; this task therefore requires the prediction of relatively complex structured entities that are characterized with rich semantics. Since the source text may be limited (as opposed to the whole Web), modeling domain semantics is critical to both precision and recall performance in extraction.

Previous efforts of modeling domain knowledge in information extraction have been limited and task-specific. MedScan, a system designed to extract interprotein interactions from biomedical texts, uses a manually constructed ontology to confirm the validity of candidate semantic parses [Daraselia et al. 2004]. Another system, aimed at the extraction of structured representations from patient-related texts, uses domain knowledge to guide named entity recognition [Angelova 2010]. In contrast with these approaches, we model domain knowledge in a framework that performs the various stages of template filling jointly using learning, rather than as a pipeline of manually designed processing steps. Cox et al. [2005] have attempted to enforce domain validity in extracting workshop announcements. They use a sequential learning model and sample candidate assignments from the sequence model. The candidate assignments are evaluated using a set of "templates," which represent intuitions about agreement in the domain: for example, workshop acronyms should resemble their names, and workshop dates occur after paper submission dates. Each template is assigned a relational score, meant to correspond to a probability given a domain model. As stated by the authors, the relational scores assigned are not well founded, and the experimental results reported show limited success.

Several researchers have attempted to model label consistency and high-level relational constraints using sequential models of NER. Mainly, predetermined word-level dependencies were represented as links in graphical models [Sutton and McCallum 2004; Finkel et al. 2005]. Finkel et al. [2005] further modeled high-level semantic constraints; for example, using the CMU seminar announcements dataset, spans labeled as *start time* or *end time* were required to be semantically consistent. In the proposed framework, we take a bottom-up approach to identifying entity mentions in text, where, given a noisy set of candidate named entities, described using rich semantic and surface features, discriminative learning is applied to label these mentions. We will show that this approach yields better performance on the CMU seminar announcement dataset when evaluated in terms of NER. More recently, Chang et al. [2012] proposed the Constrained Conditional Model framework, modeling prior knowledge into a conditional model in the form of constraints. They use constraints that are more expressive than the features used in earlier works (e.g., Finkel et al. [2005]). The domain knowledge that is modeled in this work, however, is substantially richer, as we model various aspects of the extracted record as a structured object.

This article is a revised and extended version of Minkov and Zettlemoyer [2012], adding a detailed discussion of the task and of related work, a formal presentation of the methodology, and extended presentation of the experimental results. Additional results that are included in this article establish the robustness of the proposed approach with respect to search beam size. A main focus of this article is on practical aspects involved in implementing the approach for real-world applications. In another set of experiments, we demonstrate improved generalization of the proposed approach when applied across different data distributions. A dataset of seminar announcements published on an MIT mailing list has been annotated for this purpose, which we make available to the research community.

seminar	*date*	*time*	*person*	*location*	*title*
date <*date*>	day-of-month <*s*>	**hour** <*s*>	**name** <*s*>	**location** <*s*>	**title** <*s*>
stime <*time*>	month <*s*>	minutes <*s*>			
etime <*time*>	year <*s*>	am/pm <*s*>			
location <*location*>	day-of-week <*s*>				
speaker <*person*>					
title <*title*>					

Fig. 3. An extended relational schema proposed for the *seminars* template-filling task. The field types are defined as string (<*s*>) or as pointers to tuples of designated types (e.g., <*person*>); individual fields that are defined as mandatory are marked in boldface.

3. PROBLEM SETTING

In the template-filling task, the goal is to extract structured information to fill predefined templates from text. The templates are domain-specific. Given reports on earthquake events, for example, information items of interest include *date*, *time*, *location*, and *magnitude* [Grishman and Sundheim 1996]. In the case of seminar events, the target template's slots designate *date*, *time*, *location*, as well as *speaker* and *title* information (as shown in Figure 1).

In relational terms, a target template corresponds to a relation r^T, which is comprised of a set of attributes, $A(r^T)$. Given a document d, which is known to describe a tuple of r^T, the goal is to populate every field $a \in A(r^T)$ with its correct value based on the text.

3.1. The Relational Schema

Rather than consider the target relation r^T in isolation, we will describe domain knowledge through an extended relational database schema R, $r^T \in R$. Figure 3 describes an extended relational schema for the seminar-announcement domain. In addition to the target relation *seminar*, the extended schema includes the relations *date*, *time*, *person*, *location*, and *title*. A field may be populated with a pointer to a tuple of another relation; as shown in the figure, the type of the field *seminar.speaker*[2] is defined as *person*, that is, this field links to tuples of the *person* relation. Similarly, the *start time* and *end time* fields both point to *time* tuples.

Some domain knowledge may be represented using deterministic relation-level integrity constraints, which are typically realized in a database. Concretely, fields may be defined as *mandatory* or *optional* (i.e., disallowing, or allowing, empty values). Mandatory attributes are denoted with boldface in Figure 3; as shown, the *seminar.date* field is defined as mandatory, while *seminar.title* is optional. Similar constraints can be defined per a set of attributes; for example, either *day-of-month* or *day-of-week* must be populated in the *date* relation.

Integrity constraints can also be defined over the values of the fields; for instance, the value of the *time.hour* field must reside in the range {0–24}. Similarly, complex constraints can model value agreement between fields; for example, the values of *day*, *month*, *year*, and *day-of-week* must be calendar-compatible.

We will apply a *validity* test for every relation $r \in R$, generally defined as follows:

Definition 1. A tuple $v_i \in r$ is *valid* if it obeys all data integrity constraints specified for relation r.

It is possible that multiple tuples of a given relation r be coreferent, describing the same real-world entity. We next define the *tuple contradiction* test, indicating whether a given pair of valid tuples, $v_i, v_j \in r$, are necessarily *not* coreferent:

[2]We use the notation *r.a* to denote the field (or attribute) a of relation r.

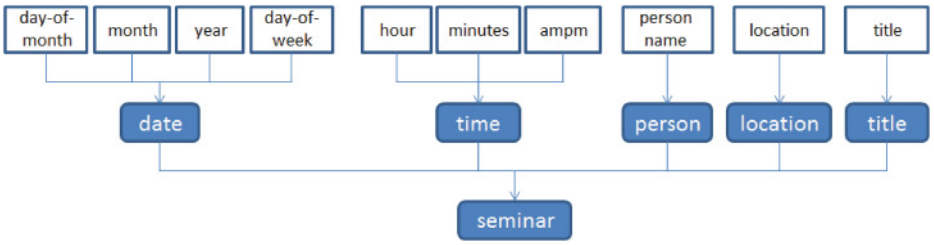


Fig. 4. The hierarchical dependency structure derived from the relational schema describing the seminars domain (Figure 3). The relations *date*, *time*, *person*, *location*, and *title* reside at the lowest level of the hierarchy, and their field values are populated with named entity mentions identified in the given text. The types of named entities that map to each relation are shown at the top of the figure; for example, the *time* relation is populated with named entities of types *hour*, *minutes*, and *ampm*.

Definition 2. A pair of tuples $v_i, v_j \in r$ are *inconsistent* (*contradictory*) if there exists a field $a \in A(r)$ for which the respective field values $v_i.a$, $v_j.a$ are known to be semantically different.

Similarly to the tuple validity test, the tuple contradiction test allows one to encode domain semantics. Consider, for example, a *date* relation schema, which consists of the fields {*day-of-week, day-of-month, month, year*}; the *date* tuples {"Mon","3","9","12"} and {"Mon","10","9","12"} are contradictory due to different values of the *day-of-month* field, whereas the tuple pair {"Mon","10","9","12"} and {"Monday","10","September","2012"} should not be considered as contradictory due to the semantic equality of the respective field values. Testing for semantic equivalence can be implemented using dictionaries of known synonyms in the case that the set of possible values is well defined. In addition, one may incorporate string similarity measures in order to allow a certain level of string variance [Bilenko et al. 2003].

In this work, we use hand-built dictionaries to model semantic equivalence between possible values of each of the *date* and *time* fields. According to the outlined schema (Figure 3), the remaining relations of *person*, *location*, and *title* consist of a single attribute modeled as a string. As a rule of a thumb, we require string values to have a minimal number of tokens in common in order to avoid contradiction; for example, the *person* tuples {"Dr. John A. Smith"} and {"John Smith"} will not be considered as contradictory if a threshold of two (or less) common tokens is used for this relation.[3]

Finally, we note that tuple contradiction is transitive. This means that a pair of tuples $v_i, v_j \in r$, for which there exists $a \in A(r)$, such that $v_i.a$ and $v_j.a$ are both nonempty and map to contradictory tuples of relation r', will also be considered as contradictory.

3.2. Template Filling

The relational schema is hierarchical in the sense that some relations point to tuples of other relations. Figure 4 illustrates the hierarchical structure that corresponds to the relational schema in Figure 3. The two-level hierarchy consists of the high-level target *seminar* relation, which maps to tuples of the relations *date*, *time*, *location*, *person*, and *title*. The fields of these low-level relations are shown in unshaded boxes in the figure; the values of these fields must be populated based on document d. That is, the schema R is populated in a bottom-up fashion, in which field values of low-level relations are filled based on relevant mentions in the source document d.

[3]One may apply a higher level of granularity to the *person* relation, differentiating between personal title, and first, middle, and last names. We leave this for future work, as we found empirically that it was not necessary for our datasets.

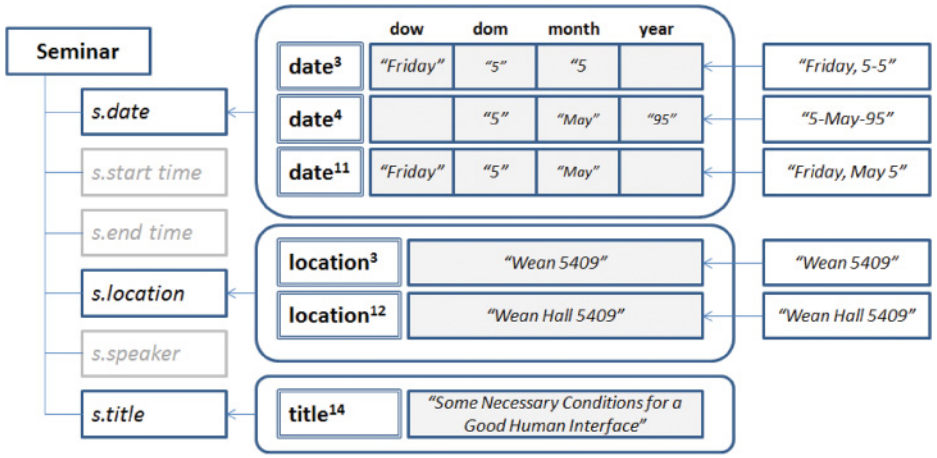


Fig. 5. A record partially populated given the example text in Figure 1, using the proposed relational schema (Figure 4). Coreferent tuples are unified, in which the respective fields of the *seminars* relation map to the unified tuples.

The first step in the template-filling process therefore involves the identification of possibly relevant named entity mentions in the text, and the representation of these mentions as valid records. Possible coreference between multiple text spans is then addressed through unification of the respective tuples. Once the conversion of named entities into tuples, and the grouping of these tuples, is complete, the remaining relations are populated in a bottom-up fashion. Invalid tuples are discarded throughout this process. Figure 5 illustrates the result of the proposed template-filling process using the example text and true entity mentions shown in Figure 1, and the relational schema given in Figure 3. The figure shows the conversion of the relevant text spans into *date*, *location*, and *title* tuples. The coreferent tuples are grouped where the respective fields of the target *seminar* relation map to the unified tuples.

In the rest of this section, we outline the steps involved in the proposed template-filling process. Multiple hypotheses are instantiated at each step; our goal is to instantiate and identify an overall correct structure, such as the one illustrated in Figure 5. Learning and inference are discussed in detail in Section 4.

3.2.1. Mention Detection. Let L be the set of relations that reside at the lowest level of the dependency structure derived from the relational schema R. In Figure 4, these relations are *date*, *time*, *location*, *person*, and *title*. The goal of the mention detection step is to extract named entity mentions from document d that correspond to the field values of each relation $r \in L$. The types of named entities that need to be found in the text are derived by the semantic types of the individual fields. The types of entities that need to be annotated in the text per the *date* relation, for example, include *day-of-month*, *month*, *year*, and *day-of-week* (Figure 4). The extracted text spans should then be processed into valid structured tuples of the respective relation.[4] As shown in Figure 5, the consecutive text spans "Friday", "5", "5" are mapped to the *date* tuple {"Friday","5","5",}; similarly, the mentions "Wean 5409" and "Wean Hall 5409" are both represented as tuples of the *location* relation.

For every tuple constructed, we maintain pointers to the source text spans in document d. For instance, there are two *time* mentions in the document shown in Figure 1

[4]In the multiattribute relations of *date* and *time*, an additional requirement is that the set of text spans that map to a tuple be sequential up to a small distance.

with the identical value "3:30"; two separate *time* tuples will be created to represent these mentions, with identical field values, but different text spans coordinates associated with them.

3.2.2. Tuple Unification and Normalization. The mention detection phase results in the representation of entity mentions as structured tuples, some of which may be coreferent; for example, the *location* mentions "Wean 5409" and "Wean Hall 5409" corefer to the same physical room. We are interested in reasoning at entity, rather than at mention, level. It is therefore desired to *unify* coreferent tuples; for example, as shown in Figure 5, the tuples representing the mentions "Wean 5409" and "Wean Hall 5409" are resolved into a single unified set.

The tuples in a coreferent set may have different string values, due to lexical variation. Normalizing the different values of the unified tuples into a single value is another component of template filling. The normalized values can be readily used by downstream applications or for coherent presentation to an end user. We partially address normalization in this work: tuples of semantically detailed (multiattribute) relations, for example, *date* and *time*, are resolved into their semantic union, whereas tuples of other relations, for example, *location*, are normalized to the longest string in the unified set. (Alternatively, if a list of valid conference room names were available, one could normalize the set of extracted room names to the closest listed value using string similarity and learning techniques, e.g., Crim et al. [2005].)

3.2.3. Template Population. To complete the template-filling process, the remaining relations of the schema $r \in \{R - L\}$ are populated bottom-up, following the relational hierarchy structure. We iterate over the fields of every relation, in which each field in turn links to an already populated tuple (or to a unified tuple set) of the respective relation. For example, as shown in Figure 5, the *seminar.location* field should be linked in this case to the unified set of the extracted *location* tuples {"Wean Hall 5409","Wean 5409"}.

4. STRUCTURED LEARNING

The product of the template-filling procedure is a populated tuple of the target relation, including detailed mappings of its field values to populated entries of other relations of the extended schema. In addition, we maintain the association of field values with the source text spans in document d. The generated output therefore comprises a rich and complex structured entity. We thus apply structured learning to this problem. In this framework, high-level information about candidate output structures is modeled as features; for example, we model semantic correspondences between the values of multiple fields. Section 4.1 outlines the discriminative structured-learning scheme due to Collins [2002], which is used in this work. This approach requires a complimentary inference procedure. Here, we propose using beam-search inference. Section 4.2 describes the general process of instantiating a search space of *valid* structured candidates. As the resulting search space can be very large, beam search is incorporated throughout the candidate-generation process, as discussed in Section 4.3.

4.1. Learning

We employ *structured perceptron*, a discriminative structured-learning algorithm by Collins [2002], which has been successfully applied to a variety of natural-language processing tasks in the past [Collins and Roark 2004; McDonald et al. 2005; Zettlemoyer and Collins 2009].

The classical perceptron is a simple and efficient online-learning algorithm [Rosenblatt 1958]. It uses a linear scoring function of the form $F(x) = \sum_j \alpha_j f_j(x)$,

that is, a linear combination of predefined features f describing the observed input x, weighted using a respective set of parameters α. In a binary setting, it is desired that the function F correctly predict the label of every given instance x, in which a positive prediction score ($F(x) > 0$) implies positive association with the target label, and vice versa. The weights α are learned using an error-driven procedure. The training procedure involves the processing of individual labeled examples; in the case that the predicted label is incorrect, the weights vector is updated to better fit that example. In the case that the feature representations of the considered examples are linearly separable, the algorithm is guaranteed to converge. It is common practice to make several passes, called *epochs*, over the training examples until the number of errors converges to a minimum.

In structured learning settings, the goal is to correctly predict a structured entity for each input example, rather than a binary label. Concretely, in this work, each input example is a text document d and the structured output y is a populated relational schema. Using the structured perceptron algorithm, an input example and candidate output pair is described using a set of joint features. Here, we encode a set of feature functions $f_j(y, d, R)$, $j = 1, \ldots, m$, which jointly describe the candidate record y, its mappings onto the source document d, as well as any relevant domain knowledge that is available through the extended schema R.

Similar to the classical perceptron, a mapping of a given input into a candidate structure is evaluated using a weighted linear "goodness" function:

$$F(d, y, R, \alpha) = \sum_{j=1}^{m} \alpha_j f_j(y, d, R), \tag{1}$$

where α is the weight vector. In order to infer the most likely structured entity, one must first generate candidate mappings of input (document d) into a set of plausible structured outputs, $GEN(d)$; ideally, each candidate output must be evaluated using Equation (1), so that the top-scoring output $\hat{y} \in GEN(d)$ is selected. In sequence tagging problems, optimal dynamic programming algorithms are applied to scan the search space and find the top-scoring structured prediction efficiently [Collins 2002]. In cases when exact inference is intractable, because the search space is too big or does not factor locally, approximate inference must be used. The record extraction task falls into the latter category. In this work, we therefore perform approximate incremental inference using beam search. A somewhat similar approach has previously achieved competitive results on a set of syntactic natural-language processing tasks, compared with dynamic programming methods [Zhang and Clark 2011]. The proposed inference procedure is discussed in detail in the Sections 4.2 and 4.3.

As in the classical perceptron, learning involves updating the weights α in a supervised fashion. Given each input (document) in turn, in the case that the inferred mapping $\hat{y}$ is different from the true mapping t: (i) α is incremented with the feature vector pertaining to the correct known structure, and (ii) the feature vector of the erroneously predicted candidate is subtracted from α. The following equation summarizes the weight update rule:[5]

$$\alpha = \alpha - f(\hat{y}, d, R) + f(t, d, R). \tag{2}$$

The perceptron algorithm is prone to overfit the training data. Further, if the examples are not linearly separable with respect to the defined feature set, then the

[5]We refer the reader to a discussion on the validity of this update rule using inexact beam search decoding; while it has been shown to be empirically successful, a set of refined update strategies may improve results further [Huang et al. 2012].

weight-learning process will not converge, and the parameter weights will oscillate instead. The averaged perceptron variant for weight learning [Freund and Schapire 1999] alleviates these issues [Collins 2002]. If an averaged perceptron is used, then the algorithm outputs a running average of the weights:

$$\bar{\alpha} = \frac{\sum \alpha_{ir}}{NP}, \tag{3}$$

where N is the total number of training examples, P is the number of training epochs, and α_{ir} is the weight vector after processing example i ($i = 1, \ldots, N$) during epoch r ($r = 1, \ldots, P$). Weight averaging reduces the variance between the different weight vectors, thus producing a regularization effect. This improves performance by neutralizing random oscillations in this online-learning procedure. A recent work [Zhang et al. 2014] discusses the regularization of the structured perceptron algorithm using weight averaging, and proposes several alternative regularization approaches. In the experiments reported in this article, all weights were initialized to zero. The number of epochs was set to 7, allowing the performance of the learned models to converge on the training examples in terms of average F_1 performance. (Alternatively, if possible, the number of training iterations may be determined using a set of development test data, selecting the parameter vector that performs best on the held-out examples.) Learning of the weight parameters was efficient, requiring about a couple of hours using a standard PC. Learning time is generally governed by the inference phase, in which candidates are generated and scored. In the case that large training sets are used, or high-inference complexity is encountered, the training time of the structured perceptron can be reduced by means of distributed computing [McDonald et al. 2010].

4.2. Inference

In order to find the best scoring candidate $\hat{y}$, the space of candidate outputs needs to be instantiated and evaluated. Since it can generally be very large, we control the search space size by applying proximate beam search throughout the candidate generation process. As will be shown, using beam search results in computational gains while retaining the correct candidate within the instantiated search space in most cases.

In this section, we describe how the search space of populated instances of the relational schema R is generated from document d, following the conceptual steps described in Section 3.2. The beam search procedure is formalized in Section 4.3.

4.2.1. Named Entity Recognition. For every relation $r \in L$, where L is the set of relations residing at the lowest level of the hierarchical structure derived from the relational schema R, a set of candidate mentions $S_d(a)$ is extracted from document d per each attribute $a \in A(r)$. We aim at extracting a set of (possibly overlapping) candidate text spans that is characterized with *high recall*, that is, it is desired that $S_d(a)$ contain the correct mentions with high probability. Various NER techniques, as well as an ensemble of methods, can be employed for this purpose [Kou et al. 2005; Minkov et al. 2006].

Given the extracted text spans $\bigcup_{a \in A(r)} S_d(a)$, a set of candidate tuples $E_d(r)$ is constructed that form all valid combinations of field mappings to the extracted text spans. Formally, if r consists of n attributes ($\mid A(r) \mid = n$), then the size of tuples constructed in this fashion may reach $\mid S_d(a_1) \mid \times \mid S_d(a_2) \mid \ldots \times \mid S_d(a_n) \mid$. However, records of relation r that fail the validity test are discarded in this process. For example, while the strings "3", "30", "Friday" may be extracted as *month*, *day-of-month* and *day-of-week* mentions, respectively, based on the text "3:30 Friday" (Figure 1), a *date* tuple constructed from

this combination of field values may be found to be inconsistent,[6] resulting in the elimination of this tuple. In addition, it is required that a single record does not map to overlapping text spans.

4.2.2. Unification. For every relation $r \in L$, we construct candidate sets of unified tuples, $\{C_d(r) \subseteq E_d(r)\}$. Naively, the number of subsets is exponential in the size of $E_d(t)$. Importantly, however, the tuples within a candidate unification set are required to be *noncontradictory*. In addition, the text spans that comprise the mentions within each set must not overlap. Finally, we do not split tuples with identical string values between different unified subsets.

For example, suppose that, given the header "Topic: HCI Seminar, Raj Reddy, 3:30 Friday 5-5, Wean 5409" (Figure 1), two valid *date* tuples were constructed, including {"Friday","30","3",}, and {"Friday","5","5",} (based on overlapping text spans). Assume that two additional tuples were extracted from the full text: {"Friday","5","May",} and {"5","May","95"}. Overall, there are $2^4 = 16$ different unified subsets possible in this case. However, the first tuple contradicts each of the other tuples due to different *day-of-month* and *month* values, and would constitute a separate unified set; the three remaining tuples would be grouped into $2^3 = 8$ unified sets.

4.2.3. Candidate Tuples. In order to construct the space of candidate tuples of the target relation, the remaining relations $r \in \{R - L\}$ are visited bottom-up, where each field $a \in A(r)$ is mapped in turn to a (possibly unified) populated tuple, or to an empty value. The valid (and nonoverlapping) combinations of field mappings constitute a set of candidate tuples of r.

Following our earlier example, the *date* field of the candidate *seminar* tuples would point to one of the 8 different unified sets constructed, or would be left empty. Assuming that there are N unified sets of *time* tuples, the number of partially populated candidate *seminar* tuples can reach $9(N + 1)$, and so forth. In practice, the number of valid candidates is much lower. For instance, a *seminar* candidate record that maps both to the *date* tuple {"Friday","30","3",} and to the *time* tuple {"3","30",}, having both tuples map to overlapping text spans, would be discarded as a candidate *seminar* record.

In this work, it is assumed that each field contains one value at most. This restriction can be removed at the cost of increasing the size of the decoding search space.

The candidate tuples generated using this procedure are structured entities, constructed using typed named entity recognition, unification, and hierarchical assignment of field values. We evaluate the "goodness" of the candidate tuples using features that describe local as well as global properties of each candidate, encoding various types of information.

4.3. Beam Search

The space of structured candidates constructed using the procedure described in Section 4.2 may be large. Unfortunately, optimal local decoding algorithms (such as the Viterbi algorithm in tagging problems [Collins 2002]) cannot be applied to our problem. We therefore propose using beam search to efficiently find the top scoring candidate. This means that, rather than instantiating the full space of valid candidate records, we are interested in instantiating only those candidates that are likely to be assigned a high score by the scoring function (Equation (1)).

Algorithm 1 outlines the proposed beam search procedure. Using this procedure, a limited set of top-scoring tuples of predefined beam size k is maintained for every relation $r \in R$ during candidate generation. For high-level relations, $r \in \{R - L\}$, fields

[6]Assuming that year information is inferred automatically, and the combination of field values mismatches the calendar.

ALGORITHM 1: The Beam Search Procedure

(1) Populate every low-level relation $r \in L$ from text d:
  —Construct a set of candidate valid tuples $E_d(r)$ given high-recall typed candidate text spans $\cup_{a \in A(r)} S_d(a)$.
  —Group $E_d(r)$ into possibly overlapping unified sets, $\{C_d(r) \subseteq E_d(r)\}$.
(2) Iterate bottom-up through relations $r \in \{R - L\}$:
  —Initialize the set of candidate tuples $E_d(r)$ to an empty set.
  —Iterate through attributes $a \in A(r)$:
    —Retrieve the set of candidate tuples (or unified tuple sets) $E_d(r')$, where r' is the relation that attribute a links to in R. Add an empty tuple to the set.
    —For every pair of candidate tuples $e \in E_d(r)$ and $e' \in E_d(r')$, modify e by linking attribute $a(e)$ to tuple e'.
    —Add the modified tuples, if valid, to $E_d(r)$.
    —Apply Equation (1) to rank the partially filled candidate tuples $e \in E_d(r)$. Keep the k top-scoring candidates in $E_d(r)$, and discard the rest.
(3) Apply Equation (1) to output a ranked list of extracted records of the target relation, $E_d(r^t)$.

are populated incrementally, having each attribute $a \in A(r)$ map in turn to (unified) populated tuples of its type, where Equation (1) is applied to find the k highest-scoring *partially* populated tuples. In this process, evaluated records that are not included in the set of k top-scoring candidates are discarded. Thus, assuming that $r \in \{R - L\}$ consists of $|A(r)| = n$ fields, the total number of candidates evaluated for relation r is confined by nk^2. Ideally, the top candidates retained should include the correct field mappings.

Note that features that pertain to yet unfilled fields are meaningless (inactive). In particular, features that describe interactions between multiple field values become active once all of these fields have been processed. For example, only once candidate *seminar* tuples are populated with both *stime* and *etime* values, features that describe interactions between these values will contribute their weights to the evaluation score.

5. SEMINAR EXTRACTION TASK

In this article, we discuss in detail the task of automatically extracting scientific seminars from email announcements. There are several reasons why we are interested in this task. First, the automatic processing of email announcements is a representative case study of learning to decode semistructured text, including the genres of email, Web pages, and more. Secondly, we experiment with a well-studied dataset, on which a variety of methods have been previously investigated. This allows us to carefully evaluate the proposed approach against alternative methods applied in the past in similar settings. Finally, application-wise, we are interested in the extraction of events in the context of location-based services. It may be desired to recommend events to users, such as scientific seminars, that take place nearby [Minkov et al. 2010], or to detect users' presence at known events in order to implicitly learn about their interests [Park et al. 2010]. For both purposes, a dataset of event details must be maintained, consolidated from various resources.

5.1. Dataset

We experiment with the CMU seminar-announcement corpus [Freitag and McCallum 2000], which includes 485 emails containing seminar announcements. Figure 1 shows a representative email message from this corpus. The messages have been originally annotated with text spans referring to four slots: *speaker*, *location*, *stime* (denoting the seminar's designated start time), and *etime* (end time). We have annotated this dataset

with two additional attributes, denoting the seminar's *date* and *title*.[7] As shown, this corpus contains semistructured text, in which some of the field values appear in the email header, in a tabular structure, or using special formatting [Califf and Mooney 1999; Freitag 2000; Minkov et al. 2005].[8]

5.2. Extraction of Candidate Entity Mentions

As discussed in Section 4.2.1, the first step in the template-filling process involves NER, in which possible mentions of relevant entity types are identified in the given document. While automatic NER is imperfect, it is needed to reach a high level of recall, to consider as much relevant evidence as possible in the subsequent template-filling process; ideally, noisy extractions will be discarded at later stages.

Given grammatical text, which applies capitalization conventions consistently, it is possible to parse the text and extract all capitalized noun phrases as candidate name mentions [Haghighi and Klein 2010]. The processing of semistructured and informal text, such as email, is more challenging. A variety of NER techniques, as well ensemble of methods, can be used in these settings [Minkov et al. 2005]. In this work, we designed a set of rules to extract candidate named entity mentions from the email seminar announcements. The rule language used is based on cascaded finite state machines [Minorthird 2008]. The rules encode information typically used in NER, including content and contextual patterns, as well as lookups in available, or hand-crafted, dictionaries [Finkel et al. 2005; Minkov et al. 2005]. For example, candidate *location* mentions include words that contain digits, or indicative words like "room", "hall", and "office". Seminar *titles* are typically long and demonstrate highly diverse dictionary usage. We extracted sequences of capitalized words as candidate titles, if they captured a full line designated with special styling, such as centering, indentation, tabular form, or appearing within quotes. Likewise, we consider as candidate *speaker* names, any sequence of capitalized words following personal titles, such as "Professor", "Doctor', "Dr.", "Mrs." and so on, as well as contiguous word sequences in which one of the words appears in a dictionary of personal names, and so forth. In order to increase recall further, we applied the following techniques to extract additional possible mentions of *speaker* names, following Minkov et al. [2005]:

—We annotated individual capitalized words included in already extracted name spans as candidate *speaker* names. In this fashion, identifying formal and structured name mentions, such as "Prof. Raj Reddy", enables the extraction of less trivial single-word name mentions, for example, "Reddy".
—We also extracted subspans that appeared in multiple extracted names as additional name candidates. For example, if "Dr. John Smith" was extracted, as well as "John Smith CMU" (where "CMU" is, in fact, the affiliation), this rule would result in another candidate name mention with the correct value "John Smith".

The set of extracted candidates is characterized with high recall where some of the extracted text spans overlap. Specifically, the recall of the extracted named entities associated with the fields of *date* and *time* is nearly perfect, and is estimated at 96%, 91%, and 90% for *location*, *speaker*, and *title*, respectively.

5.3. Features

Figure 6 lists the features used in learning to rank the candidate seminar records. All features are binary and typed, where real-value features were discretized into segments. We next discuss these features by category.

[7]The modified annotations are available on the first author's homepage.
[8]Such structure varies across messages. Otherwise, the problem would reduce to wrapper learning [Zhu et al. 2006].

Feature type	Comments	Examples
Lexical features		
(l1) lexical	Lexical values of content and context words of field mentions.	*location.content.wean* *location.content.5409* *location.context.right.1.dates*
(l2) pattern	String pattern of content words of the field mentions.	*location.pattern.X+x+* *location.pattern.9+*
(l3) dictionary lookups		*speaker.includesKnownName* *location.includesRoomWord*
Structural features		
(s1) in subject	Does one of the field mentions appear in the subject line?	*stime.inSubject* *location.inSubject*
(s2) is tabular	Does one of the field mentions appear in tabular form?	*stime.isTabular* *date.isTabular*
(s3) field name	If the previous feature is true, what is the field name?	*stime.fieldName.time* *date.fieldName.dates*
(s4) is indent	Does one of the field mentions appear in indent line?	(none)
(s5) is full-line	Does one of the field mentions capture a whole line?	*location.fullLine* *title.fullLine*
(s6) is separated by space lines	Does one of the field mentions appear between blank lines?	*title.btwBlankLines*
(s7) is quoted	Does one of the field mentions appear within quotes?	*title.isQuoted*
(s8) max. length	What is the maximal length (in tokens) of the field mentions?	*title.maxLen.*>7 *title.maxLen.*>6 ...
(s9) num. mentions	What is the number of mentions associated with the field?	*speaker.numMentions.*>2 *speaker.numMentions.*>1
(s10) num. identical	What is the number of mentions with identical values associated with the field?	*speaker.sameMentions.*>2 *speaker.sameMentions.*>1
Semantic features		
(e1) duration	Event duration in minutes, computed based on *stime* and *etime* values. Discretized at $30, 60, 90, 120, 150, 180$ minutes.	*duration.none*
(e2) round time	Is time value round at either 5- or 10-minute resolution?	*stime.round.5* *stime.round.10*
(e3) specifies field value	Applies to *time* and *date* fields.	*date.specifiesDayOfWeek* *date.specifiesDayOfMonth* *date.specifiesMonth* *seminar.specifiesDate*
*(e4) max. num. fields specified**	What is the maximal number of fields specified by a single mention?	*date.maxFields.*>2 *date.maxFields.*>1
*(e5) min. num. fields specified**	What is the minimal number of fields specified by a single mention?	*date.minFields.*>2 *date.minFields.*>1
*(e6) total field mentions**	What is the overall number of field mentions of the target relation? For example, the three *date* mentions specify 9 (duplicate) field values overall.	*date.totalFieldMentions.*>8
Cross-field features		
(c1) sentence co-occurrence		
(c2) same format		*date.time.inField*
*(c3) min-separating-lines**		*title.stime.minSepLines.1*

Fig. 6. A detailed list of features types used for seminar extraction. Feature types that are marked with an asterisk represent counts rather than Boolean values, and are discretized into Boolean feature representation. Example features are given in the rightmost column, describing the correct field extractions shown in Figure 1.

Lexical features. The lexical information modeled includes the value and pattern of words within and around the text spans that correspond to each field. For example, the message in Figure 1 includes the *location* mentions "Wean 5409" and "Wean Hall 5409"; the corresponding *location* (and *seminar*) tuples are described accordingly by the lexical features *location.content.word.wean*, *location.content.word.hall*, *location.content.word.5409*, and *location.pattern.X+x+*, denoting a capitalized word pattern. Similar features are derived for a window of three words to the right and to the left of each text span. In addition, we observe whether the words that comprise the text spans appear in relevant dictionaries: for example, whether the spans assigned to the *location* field include words typical of location, such as "room" or "hall".

Structural features. It has been previously shown that structured information available in semistructured documents, such as email messages, is useful for information extraction [Minkov et al. 2005; Schneider 2006; Siefkes 2008]. As demonstrated in Figure 1, an email message typically includes a header, specifying textual fields such as *topic*, or *subject*, *date*, and *time*. In addition, blank lines and line breaks are used to emphasize blocks of important information. We propose a set of features that model correspondences between the text spans assigned to each field in the relational schema and the underlying document structure. These features model whether at least one of the spans mapped to each field appears in the email header, captures a full line in the document, is indent, appears within quotes, appears within space lines, or in a tabular format. Following the example shown in Figure 1, structural features that describe the correctly populated relational schema in this case include *location.inSubject* (since the text span "Wean 5409" appears in the subject line of the message), *location.fullLine* (since the text span "Wean Hall 5409" captures a full line), *title.betweenBlankLines* (the text span tagged as the seminar's title is separated from the rest of the content by blank lines), and so on.

Additional features pertain to properties of the unified entity sets that map to each field. These features encode the number of mentions that comprise the unified set and the number of mentions with identical values within each set. For example, the speaker name "Raj Reddy" is mentioned three times in the message in Figure 1. The mentions are identical, so that the number of mentions with identical values that map to the field *person* equal three as well. These features are intended to encourage the creation of complete unified sets so that richer evidence is available at the field level.

Semantic features. We use an additional set of features to represent the semantic interpretation of field values. According to the proposed relational schema (Figure 4), the *date* and *time* relations consist of multiple attributes, whereas other relations are represented as a single field. The encoded semantic features therefore refer to *date* and *time* only. Specifically, these features indicate whether a unified set of *date* or *time* tuples defines a value for all attributes. For example, in Figure 1, the unified set of *date* tuples fully specifies the attribute values of this relation, including *day-of-month*, *month*, *year*, and *day-of-week*. Intuitively, semantically detailed tuples are to be preferred over less detailed extracted tuples. Another feature encodes the size of the most semantically detailed individual tuple extracted. For example, each of the entity mentions of type *date* in Figure 1 details three attribute values; for instance, the mention "Friday 5-5" maps to a *date* tuple populated with the field values of *day-of-week*, *day-of-month*, and *month*. Similarly, the total number of semantic units included in a unified set is represented as a feature. These features were all designed to favor semantically detailed mentions and unified sets, where we wish to recover the full coreferent set of relevant named entity mentions. Finally, domain-specific semantic knowledge is encoded as features, including the *duration* of the seminar, and whether a *time* value is round (e.g., minutes perfectly divide by 5 or 10).

In addition to the features described so far, one may be interested in modeling cross-field information. We have experimented with features that encode the shortest distance between named entity mentions mapping to different fields (measured in terms of lines or sentences), based on the hypothesis that the various field mentions typically appear in the same segment of the document. These features were not included in the final model since their contribution was marginal. We leave further exploration of cross-field features in this domain to future work.

The application of these feature templates to the CMU seminar-announcement dataset (using a 5-fold cross-validation procedure) results in about 15,000 features. The vast majority of these features are lexical, structural information is represented using about 200 features, and semantic information using about 120 features. We note, however, that the observed frequency of most of the lexical features is low. As the learned feature weights are affected by their frequency, the contribution of the various feature groups to the output candidate score is relatively balanced.

5.4. Experiments

We first define several evaluation measures that are used to describe our experimental results, discussed thereafter.

5.4.1. Evaluation Measures. Following previous works, we assume that a single record is described in each document, and that each field corresponds to a single value (e.g., Roth and Yih [2001], Siefkes [2008], and Haghighi and Klein [2010]). These assumptions are violated only in a few cases.

In order to gain some insights on the system's performance, and in order to compare against previous results, which used nonuniform evaluation measures, we report performance at several resolutions, as described here. At each resolution, we compute aggregate precision, recall and $F1$ metrics per field type, following previous works [Lavelli et al. 2008].

—*Entity level*. Each field of the populated template, $a \in r^T$, is associated with a set of named entity mentions $S_d(a)$. We compare this predicted set against $S_d^*(a)$, the set of text spans that are annotated as relevant (as in Figure 1). Entity-level evaluation is strict in the sense that credit is awarded for a tagged entity mention $s \in S_d(a)$ only if both of its boundaries are detected correctly.

—*Token level*. This is a lenient version of the entity-level measure, in which, rather than consider text spans, the unit of reference is a single token.[9] Both entity- and token-level performance measures are commonly used in the evaluation of NER systems [Minkov et al. 2005; Lavelli et al. 2008].

—*Field level*. The ultimate goal of template filling is to populate each field with a single correct value. We perform in this work limited value normalization based on the extracted entities per field, $S_d(a)$. Specifically, we output the semantic union of the tuples mapped to the fields *seminar.date* and *seminar.time*; otherwise, the longest string value is considered (Section 3.2). Since the labeled datasets do not provide slot-level annotations, we produce the respective reference values from the correct unified entity set $S_d^*(a)$ using the same procedure. The field-level measure is strict: partial value matches are counted as errors [Siefkes 2008]. We note, however, that while entity-level measures assess the extraction of all relevant entity mentions, the correct field value may be recovered based on a subset of these mentions.

Example. According to Figure 1, relevant *location* mentions in the shown document are "Wean 5409" and "Wean Hall 5409". Suppose that the field *seminar.location* of the

[9]We consider words as tokens, ignoring punctuation marks.

		Date	Stime	Etime	Location	Speaker	Title
Token level	Precision	.937 (.034)	.962 (.005)	.976 (.024)	.954 (.041)	.854 (.048)	.741 (.057)
	Recall	.972 (.012)	.981 (.013)	.983 (.013)	.987 (.012)	.876 (.071)	.770 (.064)
	F1	*.954* (.022)	*.971* (.006)	*.979* (.011)	*.970* (.021)	*.865* (.054)	*.756* (.055)
Entity level	Precision	.918 (.034)	.965 (.005)	.972 (.027)	.951 (.044)	.815 (.060)	.699 (.039)
	Recall	.948 (.014)	.982 (.008)	.981 (.012)	.974 (.018)	.850 (.067)	.699 (.073)
	F1	*.933* (.013)	*.973* (.005)	*.977* (.018)	*.965* (.025)	*.833* (.056)	*.699* (.047)
Field level	Precision	.941 (.038)	.992 (.005)	.982 (.010)	.957 (.026)	.862 (.045)	.694 (.020)
	Recall	.981 (.018)	.994 (.006)	.991 (.013)	.970 (.027)	.856 (.120)	.696 (.056)
	F1	*.961* (.021)	*.993* (.003)	*.987* (.009)	*.964* (.017)	*.859* (.077)	*.695* (.034)

Fig. 7. Detailed results of applying the full model proposed for the CMU seminar-announcement dataset using 5-fold CV. CV standard deviation figures are shown in parentheses.

populated template maps to a unified set of predicted *location* tuples, which are associated with the extracted mention values "5409" and "Wean Hall 5409"; the normalized *location* value equals "Wean Hall 5409", based on either of these entity sets. Entity-level precision and recall both equal 0.5 in this case, as the boundaries of first text span were not perfectly identified. Token-level precision equals $4/4 = 1.0$, and token-level recall is computed as $4/5 = 0.8$. The normalized *location* value is recovered correctly, so that field-level precision and recall for this individual example are both 1.0.[10]

Notably, the annotated labels, as well as text itself, are not error-free; for example, in some announcements, the calendric date and day-of-week specified are inconsistent. Our evaluation is strict: nonempty predicted values are counted as errors in such cases.

5.4.2. Experimental Results. We conducted 5-fold cross-validation experiments, using the same data splits as in previous works [Sutton and McCallum 2004; Finkel et al. 2005].

Figure 7 shows detailed results of our full model using beam size $k = 10$. We discuss this choice of beam size in Section 8. As shown, field-level performance is very high for the fields of *date*, *time*, and *location* ($F1 > .95$), and is lower for the *speaker* field ($F1 \sim .85$), where the *title* field is the most challenging ($F1 \sim .70$). Similar trends are reported for token- and entity-level performance, where results for the more lenient token-level measure are higher. Interestingly, the results are fairly balanced in terms of precision and recall, recall being slightly higher than precision. We attribute this to the high recall that characterizes the candidate named entity set that is initially extracted. In addition, we find that the learned model tends to prefer the assignment of nonempty values to the various fields, based on the training examples.

Figure 8 gives the results of model variants, where every feature group was eliminated in turn in order to evaluate its contribution to the full model's performance. As shown, removing the structural features hurt performance consistently across fields, leading to an average reduction of 2.2% in field level $F1$ scores compared with the full model. Modeling structural information is especially useful for the *title* field, which is otherwise characterized with low content and contextual regularity. Removal of the semantic features affected performance for the *stime* and *etime* fields, the two fields modeled by these features. In particular, the optional *etime* field, which has fewer occurrences in the dataset, benefits from modeling semantics.

An important question to be addressed is to what extent the joint modeling approach contributes to performance. In another experiment, we mimic the typical scenario of template filling, in which the value of the single highest scoring named entity is assigned to each field. In our framework, this corresponds to a setting in which tuple

[10]Field-level precision and recall deviate when aggregated over multiple examples due to possible empty slots.

	Date	Stime	Etime	Location	Speaker	Title	Avg.
Full model	.961 (.021)	.993 (.003)	.987 (.009)	.964 (.017)	.875 (.077)	.695 (.034)	
No structural features	.949 (.009) *-1.2*	.991 (.006) *-0.2*	.980 (.006) *-0.7*	.961 (.021) *-0.3*	.838 (.120) *-4.2*	.651 (.046) *-6.3*	*-2.2*
No semantic features	.961 (.021) *0*	.987 (.010) *-0.6*	.954 (.043) *-3.3*	.965 (.027) *0*	.875 (.077) *0*	.695 (.034) *0*	*-0.7*
No unification	.872 (.056) *-9.3*	.970 (.025) *-2.3*	.951 (.059) *-3.6*	.945 (.042) *-2.0*	.760 (.095) *-13.1*	.627 (.058) *-9.8*	*-6.7*
Individual fields	.965 (.023) *0.4*	.972 (.026) *-2.1*	-	.964 (.017) *0*	.868 (.050) *-0.8*	.645 (.031) *-7.2*	*-1.9*

Fig. 8. Field-level F1 results of applying the full model proposed and its variants to the CMU seminar-announcement dataset using 5-fold CV. CV standard deviation figures are shown in parentheses.

	Date	Stime	Etime	Location	Speaker	Title
SNOW [Roth and Yih 2001]	-	**99.6**	96.3	75.2	73.8	-
BIEN [Peshkin and Pfeffer 2003]	-	96.0	**98.8**	87.1	76.9	-
Elie [Finn 2006]	-	98.5	96.4	86.5	**88.5**	-
TIE [Siefkes 2008]	-	99.3	97.1	81.7	85.4	-
Full model	96.3	99.1	98.0	**96.9**	85.8	67.7

Fig. 9. A comparison of seminar extraction results (trained on 50% of corpus): Field-level F1.

unification is not performed. The results are shown in Figure 8 (no unification). Due to reduced evidence in considering a single entity versus a coreferent set of entities, performance degrades significantly, where $F1$ score is 6.7% lower, on average, compared with our full model. Finally, we experimented with populating every field of the target schema independently of the other fields. While results are overall comparable on most fields, this had a negative impact on the *title* field. We found this to be largely due to erroneous assignments of entities associated with other fields (mainly, *speaker*) as titles; such errors are avoided in the full joint model, in which tuple validity is enforced.

Figure 9 provides a comparison of template-filling performance using our full joint model against previous state-of-the-art results. These results were all obtained using half of the corpus for training, and its remaining half for evaluation, reporting average performance over five random splits. In order to allow a fair comparison, we used 5-fold cross-validation, using the same data splits as in our experiments reported in Figure 7, in which only a subset of each train fold that corresponds to 50% of the corpus served for training. (Due to the reduced training set size, the results are slightly lower than in Figure 7.) The best results per field are marked in boldface. As shown, the proposed approach yields the best or second-best performance on each of the target fields, giving the best performance overall. A variety of methods have been applied in previous works. The SNOW system [Roth and Yih 2001] first identifies high recall, possibly overlapping, candidate fragments in text (similar to our approach), using a classifier to pick the right fragment per slot. TIE [Finn 2006] learns classifiers to detect the beginning and ending boundaries of every entity mention, achieving the best performance on the multitoken *speaker* field. BIEN [Peshkin and Pfeffer 2003] applies a Dynamic Bayesian Network model for token tagging with field labels. They encode limited domain knowledge, such as the fact that *etime* never precedes *stime*, as well as the fact that a speaker is never a verb, in the network's conditional probability tables. TIE [Siefkes 2008] applies a set heuristics to recognize and explicitly represent (in HTML tags) structural and formatting information, such as blocks of emphasized text, lists, and headers, considering this information additional evidence in field value extraction. Our framework represents the various document layout regularities and semantic

	Date	Stime	Etime	Location	Speaker	Title
[Sutton and McCallum 2004]	-	96.7	97.2	88.1	80.4	-
[Finkel et al. 2005]	-	**97.1**	**97.9**	90.0	84.2	-
Full model	95.4	**97.1**	**97.9**	**97.0**	**86.5**	75.5

Fig. 10. A comparison of seminar extraction results: Token-level F1 (best results in boldface).

aspects modeled in these previous works. Using discriminative structural learning, we are able to model richer relational domain knowledge, as well as record coherence. As argued before, the proposed approach is especially useful for the extraction of irregular fields, such as *title*; we provide first results on this field.

Previously, Sutton and McCallum [2004], and later Finkel et al. [2005], applied sequential models in performing NER on the CMU seminar-announcement dataset, with the goal of identifying all named entity mentions that pertain to the template slots. Both of these works incorporated coreference and high-level semantic information to a limited extent (as described in Section 2). We compare our approach to their works, having obtained and used the same 5-fold cross-validation splits, and reporting performance in terms of token $F1$. As shown in Figure 10, our results evaluated on the named mention recognition task are superior overall, giving comparable or best performance on all fields. These results demonstrate the benefit of performing mention recognition as part of a structured model that takes into account relational domain semantics.

6. GENERALIZATION ACROSS DATA DISTRIBUTION

So far, we have evaluated the proposed approach in traditional supervised learning settings, in which the training and test examples are drawn from the same data distribution. A comparison of our results against previous known results, as well as a feature ablation study, showed that joint modeling of template filling, as well as the incorporation of domain knowledge, are beneficial in these settings. In this section, we are interested in investigating a real-world scenario, in which one would like to apply a model learned using limited labeled data to instances drawn from a different data distribution. Typically, the performance of learned models degrades when applied to a different data distribution [Turmo et al. 2006; Quionero-Candela et al. 2009; Pan and Yang 2010]. This often occurs due to a learning bias toward properties that are characteristic of the training examples, rather than representing general phenomena. Since the proposed approach models general domain semantics and enforces the generation of semantically coherent predictions, we expect that learning using our framework would produce models that are relatively robust to data shift. Next, we show that the seminar extraction models learned based on the CMU seminar-announcement dataset using the proposed approach are effectively applied to seminar announcements published elsewhere. Concretely, we apply the learned models to another set of seminar announcements distributed at MIT. While the underlying semantics is similar, the two datasets differ with respect to the layout of the messages and lexicon usage. The experimental results that are presented in this section show that high performance is maintained using our proposed approach in this challenging real-world setup. In what follows, we first present the MIT seminar-announcement dataset, then discuss the contribution of the various elements of our proposed approach to robustness in such conditions of data variance.

6.1. MIT Seminars Dataset

We consider a corpus of scientific seminar announcements published on a dedicated email list at the MIT Computer Science and Artificial Intelligence laboratory (CSAIL)

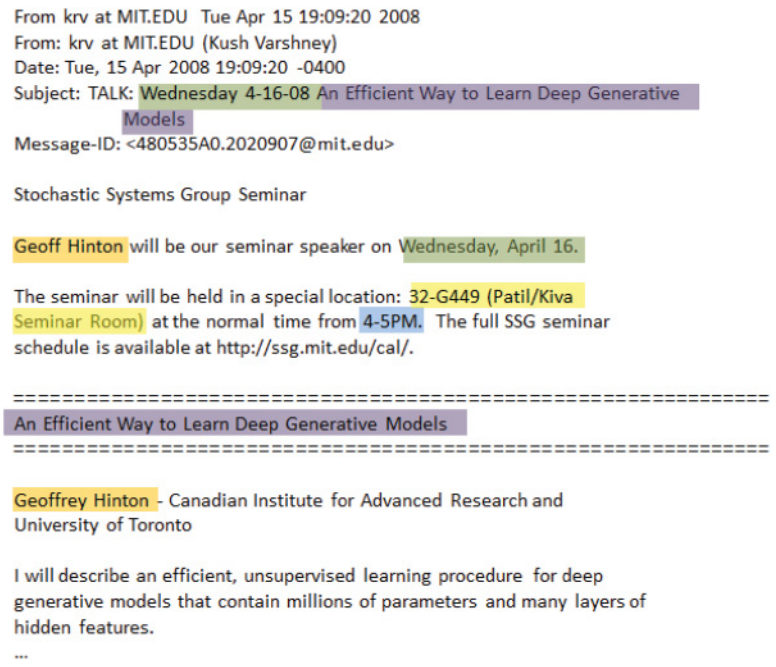


Fig. 11. An example email from the MIT seminar announcement dataset. Field mentions are highlighted in the text, grouped by color.

between June 2002 and May 2009 [Minkov et al. 2010]. Overall, this corpus includes about 5,000 seminar announcements. We have randomly drawn 100 nonduplicate announcements from this corpus and annotated them with field mentions in a similar fashion to the CMU seminar-announcement dataset. Figure 11 shows an example message from the MIT dataset, in which field values are highlighted in a similar fashion to the example seminar announcement of the CMU dataset shown in Figure 1. As expected, the semantics of the information posted in the messages from both sources is similar. Minor differences exist; for example, MIT announcements specify the local host while CMU messages do not. Yet, the information contained in the considered target schema (Figure 4) is present in both distributions. Content-wise, different naming conventions are used by the two institutions for room names (MIT room names are denoted by the concatenation of a building number and room number, e.g., "32-G449", as in Figure 11), the distribution of person names that are mentioned in the email announcements is different, and so forth. Structure-wise, the messages in the two corpora were generated independently, using different wrappers. This means that, while examples seen in test time may have been generated using similar wrappers to those observed during training using the CMU seminar-announcement dataset, there is no such overlap between the templates used to create the announcements in the CMU and MIT corpora.

6.2. Experimental Results

In order to evaluate model adaptability, we apply the models learned using the CMU dataset to the extraction of seminar events from the MIT dataset. We report average results on the MIT dataset using the same models for which 5-fold cross-validation results on the CMU dataset were previously reported in Figures 7 and 8. This supports direct comparison between test performances on the two datasets.

		Date	Stime	Etime	Location	Speaker	Title
Token level	Precision	.904 (.020)	.829 (.014)	.753 (.035)	.684 (.024)	.772 (.021)	.571 (.054)
	Recall	.848 (.024)	.894 (.024)	.954 (.018)	.795 (.027)	.852 (.064)	.629 (.048)
	F1	*.876* (.021)	*.861* (.017)	*.841* (.024)	*.735* (.025)	*.812* (.038)	*.600* (.030)
Entity level	Precision	.826 (.044)	.815 (.019)	.771 (.033)	.507 (.025)	.770 (.011)	.516 (.042)
	Recall	.786 (.029)	.881 (.031)	.938 (.024)	.650 (.025)	.813 (.070)	.543 (.042)
	F1	*.806* (.027)	*.848* (.023)	*.846* (.024)	*.578* (.019)	*.791* (.040)	*.529* (.036)
Field level	Precision	.938 (.029)	.881 (.023)	.874 (.040)	.748 (.030)	.893 (.016)	.646 (.042)
	Recall	.938 (.029)	.913 (.031)	.960 (.024)	.818 (.034)	.866 (.077)	.646 (.037)
	F1	*.938* (.029)	*.897* (.023)	*.917* (.027)	*.782* (.032)	*.880* (.038)	*.646* (.037)

Fig. 12. Detailed results of applying the full models learned using the CMU seminar-announcement dataset to the extraction of seminar events from the MIT dataset. The displayed results were averaged across the models learned using 5-fold CV. CV standard deviation figures are shown in parentheses.

Detailed performance on the MIT dataset is reported in Figure 12, using multiple evaluation metrics. We contrast these results against the respective results on CMU seminar announcements, reported in Figure 7. As expected, performance on MIT seminar announcements is generally lower. In particular, average field-level $F1$ is substantially lower for the *location* field compared with the respective results on the CMU announcements (78.3 vs. 96.4). The reason for this is that room and building names repeat across messages in the CMU corpus, where the lexical features capture this information. Since building and room names, as well as room naming conventions, are different at MIT, the lexical information learned is not transferable. This shortcoming could be addressed by dictionary lookup features, if lists of known building and room names that are relevant to the target dataset were available. Interestingly, performance on the *stime* and *etime* fields dropped dramatically as well (89.7 vs. 99.3, and 91.7 vs. 98.7, respectively). Error analysis revealed that start times in the MIT dataset were often confused with reception times, preceding the talk. While the terms "reception" or "refreshments" are mentioned in 17% of the CMU messages, different language is used in this context in the two corpora. Specifically, full sentences are used in the CMU dataset, for example, "Reception will be served at . . .", versus more abbreviated language usage in the MIT dataset. This is an example of performance degradation due to a shift in data distribution. Finally, the extracted *title* value was often confused with the name of the venue in the MIT dataset; for example, seminar series names, like "warning: dangerous ideas" (or "Stochastic Systems Group Seminar", as in the message shown in Figure 11) were erroneously assigned to the talk *title* field. If the model had been trained using similar MIT seminar announcements, it would easily learn that venue names were negative examples of a seminar *title*. Finally, the extraction performance on the *date* and *speaker* names is roughly comparable for the two datasets (93.8 vs. 96.1, and 88.0 vs. 87.5, respectively). Overall, we find that the demonstrated decrease in performance reflects a general phenomenon typical of learning transfer across different data distributions.

Figure 13 shows the results of ablation experiments, in which we applied several variants of our model to the MIT dataset, evaluating the contribution of the model's components to the final outcome. We are particularly interested in evaluating the relative importance of each of the model's components in the current settings, in which the test distribution differs from the train data distribution, against their relative contribution when there is no data shift (Figure 8). As shown, the effect of modeling structural features on top of lexical information is similar in both settings–the removal of these features results in 2.4% vs. 2.2% relative decrease in performance. This reflects the fact that the examples in the two corpora, all being seminar announcements posted via email, share similar structural properties. Removal of the semantic features, which

	Date	Stime	Etime	Location	Speaker	Title	Avg.
Full model	.938 (.029)	.897 (.023)	.917 (.027)	.783 (.032)	.880 (.038)	.646 (.037)	
No structural features	.938 (.026) *0*	.899 (.028) *0.2*	.919 (.022) *0.2*	.779 (.029) *-0.5*	.841 (.075) *-4.4*	.581 (.058) *-10.1*	*-2.4*
No semantic features	.835 (.027) *-11.0*	.796 (.062) *-11.4*	.868 (.052) *-5.3*	.792 (.031) *1.2*	.875 (.038) *-0.5*	.642 (.030) *-0.7*	*-4.6*
No unification	.915 (.067) *-2.5*	.891 (.039) *-0.7*	.912 (.026) *-0.6*	.787 (.031) *0.5*	.779 (.041) *-11.4*	.641 (.030) *-0.8*	*-2.5*
Individual fields	.906 (.053) *-3.4*	.885 (.027) *-1.4*	-	.754 (.041) *-3.6*	.866 (.035) *-1.5*	.571 (.045) *-11.6*	*-4.3*

Fig. 13. Field-level F1 results of applying the models learned using the CMU seminar-announcement dataset to the MIT seminar-announcement dataset. The displayed results were averaged across the models learned using 5-fold CV. Standard deviation figures are shown in parentheses.

apply to the *date* and *time* fields in the seminars domain, causes a major degradation in performance in the extraction of these fields when distributional shift of data occurs. As discussed before, lexical information, which models content and contextual regularity, is less accurate on the test distribution in this case. Considering semantic coherence of the extracted tuples compensates for this shortcoming. Overall, removal of the semantic features results in average degradation of 4.6% on the MIT dataset, compared with average reduction of 0.7% on the CMU dataset. This demonstrates the importance of modeling domain semantics for improved model generalization. Another model variant examines the contribution of tuple unification. Unification is less beneficial on average in the current settings (decrease of 2.6% vs. 6.7% in Figure 8). In particular, the *date* and *title* fields benefit less from unification in the MIT dataset. We found that there were less mentions, on average, of these field values in the MIT corpus compared with the CMU corpus: 1.8 versus 2.3 *date* mentions, and 1.3 versus 1.5 *title* mentions, per message. The average number of mentions of *time* field values is lower in the MIT corpus as well. In addition, unification may be less effective in this case due to the lower entity-level extraction performance. Nevertheless, in both settings, modeling unification improves field-level performance. Finally, applying joint inference, as opposed to predicting field values independently, is shown to be highly beneficial in this case of distributional data shift; the relative contribution of the joint prediction on the MIT dataset is 4.3% versus 1.9% on the CMU corpus.

Overall, this study supports our claim that joint prediction and modeling of domain semantics using the proposed framework result in better generalization of the record extraction process, generating models that are more robust to changes in the underlying data distribution. We therefore find that the proposed approach is beneficial in constructing seminar, and other task-specific, record extraction applications that are intended for use in different environments and over time, both being typical conditions of data shift. While it is hard to completely avoid a certain level of degradation in performance when data shift occurs, such degradation can be minimized if further knowledge of the target distribution is modeled. Such knowledge can be represented using lookup features in relevant available dictionaries (e.g., room names) in addition to plain lexical features, which are not transferrable in some cases. Another possibility is to model local conventions (e.g., as in MIT room names) as integrity constraints in the underlying relational schema (Section 3.1). Finally, due to high generalization of the proposed approach, a relatively small number of additional labeled examples would be sufficient if one wishes to retrain the system on the extended example set and achieve a desired performance improvement.

KLM TO TAKE 15 PCT STAKE IN AIR UK AMSTERDAM, March 3 – KLM Royal Dutch Airlines <KLM.A> said it agreed to take a 15 pct stake in Air U.K. Ltd, a subsidiary of British and Commonwealth Shipping Plc <BCOM.L>, in a transaction worth around two mln stg.

A KLM spokesman said KLM already cooperated closely with Air UK, which runs 111 flights a week to Amsterdam's Schipol airport from nine UK cities.

British and Commonwealth Shipping said last week it held preliminary talks about a KLM minority stake in Air U.K. But gave no further details. KLM said it hoped the move would attract more British feeder traffic to Amsterdam Airport.

REUTER

Purchaser name	KLM Royal Dutch Airlines
Purchaser abr.	KLM
Purchaser code	KLM.A
Acquired name	Air U.K. ltd
Acquired abr.	Air U.K.
Acquired code	-
Seller name	British and Commonwealth Shipping Plc
Seller abr.	British and Commonwealth Shipping
Seller code	BCOM.L
Amount	two mln stg

Fig. 14. An example of a news article from the corporate acquisitions corpus and the corresponding filled *acquisition event* template. Field mentions are highlighted in the text, grouped by color.

7. GENERALIZATION ACROSS DOMAIN AND GENRE

Thus far, we have focused on the design and evaluation of seminar event extraction from email announcements. The proposed framework is general and may be applied to various record extraction tasks, across domains and text genres. In this section, we focus on the engineering effort that is required for modeling domain- and genre-specific knowledge within the framework. The adaptation requirements are illustrated through another case study, the goal being to extract corporate acquisition events from news articles. As previously mentioned, newswire text greatly differs from informal text such as email, and should be processed differently. Following the introduction of the task, the various aspects of system adaptation to this different problem and text genre are described. First, we discuss the design of the relational schema (Section 7.1). We then discuss the adaptation of NER to newswire text (Section 7.2). Finally, the adaptation and design of features that capture domain knowledge, as well as informative document layout and textual regularities, is described (Section 7.3).

In the corporate acquisition extraction task, the target schema has to be populated with concrete details about an acquisition event described in a textual news report. Figure 14 presents an article included in the corporate acquisitions corpus [Freitag and McCallum 2000] and the target template, correctly populated with the respective values. As shown, the template includes the official names of the involved parties: *acquired*, *purchaser*, and *seller* company names, as well as their corresponding abbreviated names and company codes. In addition, the *amount* field denotes the price paid for the acquisition.[11] According to the news report shown in the figure, KLM Royal Dutch Airlines has agreed to acquire a 15% stake in Air U.K. Ltd from British and

[11]The corporate acquisition corpus [Freitag and McCallum 2000] also considers as fields the location of the acquired company and the status of negotiations. We ignore these fields, as we find them to be inconsistently defined, have low number of occurrences in the corpus, find them to be loosely semantically related to other fields, and therefore of lesser interest to this discussion.

Acquisition	*Corp*
purchaser <*corp.*>	name <*s*>
acquired <*corp.*>	abbreviation <*s*>
seller <*corp.*>	code <*s*>
amount <*amount*>	

Fig. 15. An extended relational schema proposed for the *corporate acquisition* template-filling task.

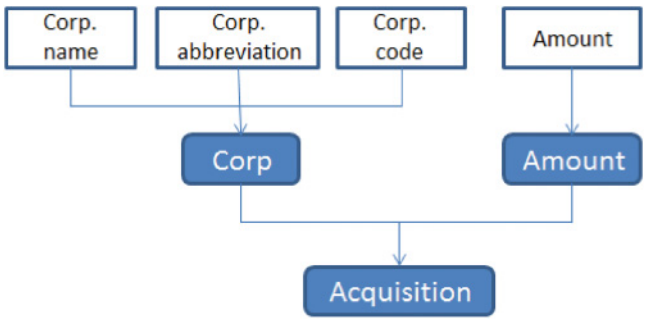


Fig. 16. The hierarchical dependency structure derived from the relational schema describing the corporate acquisition domain (Figure 15). The field values of the *corp* relation are populated with named entity mentions that are extracted from the given text. The unified sets of *corp* tuples map to different role fillers of the target *acquisition* relation.

Commonwealth Shipping Plc, a transaction worth about two million Sterling. Unlike email, news articles are formally written, strictly grammatical, and contain little structured information.

7.1. Design of the Relational Schema

We should first construct a relational schema that describes the target domain. This step involves turning the flat template representation into a relational schema–a simple normalization process.

Figure 15 shows the proposed schema for corporate acquisitions. It is comprised of a couple of relations. The *corp* relation describes a *corporate* entity, including its full name, abbreviated name and code as attributes. The target *acquisition* relation includes three role-designating attributes, each linking to a *corp* tuple. The hierarchical structure derived from this schema is shown in Figure 16. According to the outlined hierarchical dependencies, the proposed record extraction process requires first the extraction of candidate *corporate* and *amount* tuples. Unified sets of *corp* tuples will then be mapped to the different role fillers of the target *acquisition* relation. Performing joint inference across fields should allow us to find the most likely coherent assignment of the corporate entities to the different roles.

7.2. Extraction of Candidate Named Entity Mentions

According to Figure 16, four types of named entities need to be extracted using NER in the acquisitions domain, namely the corporate's *name*, *abbreviation*, and *code*, as well as mentions of *amount* values.

In our case study of seminar extraction, we considered text documents in the form of email messages. We manually crafted rules to extract named entity mentions from these email messages, which encoded both content and layout properties of the email documents. News reports, in contrast with informal genres such as email, are strictly grammatical; named entity mentions are typically capitalized in formal texts, as shown in Figure 14. It is therefore possible to apply a syntactic approach to NER in this case.

We followed Haghighi and Klein [2010], extracting candidate named entity mentions in the following manner. Each document was sentence-segmented using the OpenNLP toolkit, sentences were then parsed with the Berkeley Parser [Petrov et al. 2006], and the corresponding dependency structures obtained using the Stanford Dependency Extractor [de Marneffe et al. 2006]. The candidate name mentions are (possibly overlapping) *noun phrases* in the analyzed dependency structure of each sentence.

As discussed in Section 4.2.1, NER is complimentary to the proposed framework–one may use a variety of NER methods to identify candidate named entity mentions in a given document, which achieve high extraction recall rate. In the case studies described in this article, we experimented with rule-based NER from email messages, using overlapping rules that were tuned to yield high recall, and simply extracted all syntactic noun phrases as candidate named entities from newswire documents. Several recent works explore the task of *fine-grained* NER, in which spans of named entities are first identified and then associated with a large set of possible semantic types; for example, Ling and Weld [2012] consider more than 100 semantic tags, based on distant supervision from a large knowledge base. Such fine entity typing is suitable for record extraction settings, in which field types are diverse. In order to increase recall, it is possible to further use an ensemble of NER methods [Speck and Ngomo 2014]. Finally, we note that domain and genre adaptation of NER systems is an active research topic (e.g., Daumé III [2007]).

7.3. Features

The features used in the corporate acquisition domain differ from the feature sets described per the seminar extraction task in several respects. First, semantic features had to be designed per the acquisitions domain. We observed that there typically exists high similarity between the field values of the *corp* tuples. Dedicated features aim to capture this phenomenon, for instance, checking whether the *corp.abbreviation* forms the prefix, initials, or other string variant of the full corporate name, *corp.name*. A detailed description of the semantic features used is provided later. Further, since the underlying news articles are syntactically analyzed, context is modeled based on syntactic information, following previous work [Haghighi and Klein 2010]. Finally, newswire documents exhibit little structure compared with email. Limited structured information is available in this genre in the form of the article's header. Following is a detailed description of the different feature types.

Lexical features. The representation of populated fields uses similar features to those applied for seminar extraction. These features indicate the value and pattern of words within the text spans that are associated with each field; for example, a *corp.abbreviation* field that maps to the text span "Air UK" is assigned the features *corp.abbr.content.air*, *corp.abr.content.uk*, *corp.abbr.content.capitalized* and *corp.abbr.content.upper_case*. In addition, we perform lookups in some small domain-specific, hand-crafted dictionaries; for example, observing whether the spans assigned to the *corp.name* field include a known corporate suffix, such as "inc" or "ltd". Contextual information in this case is represented using syntactic rather than lexical features.

Syntactic features. We follow closely on Haghighi and Klein [2010] in modeling context information as lexico-syntactic neighborhoods. Since the *purchaser*, *seller*, and *acquired* role fillers are all of type *corp*, the specific role that each of the extracted *corp* entities play in the acquisition event must be identified based on the contexts in which they are mentioned in the document. The modeled lexico-syntactic context features describe the dependencies and governor of the entity mention heads. For example, the text "A KLM spokesman said. . ." contains a mention of the purchasing party "KLM" (Figure 14). The name "KLM" is linked in this sentence over a direct *noun phrase (nn)* relation with the word "spokesman", where "spokesman" is the head of the

dependent word pair; this relationship is described by the feature *purchaser.headRel.spokesman.nn*. Assuming that the purchasing party typically announces the acquisition event to the press, this feature can serve as positive evidence for the *purchaser* role. Similarly, the feature *acquired.depRel.subsidiary.appos* indicates that the text span "Air U.K. Ltd", which was assigned to the *acquisition.acquired* field, is linked over an *apposition* relation with the word "subsidiary" (based on the analysis of the sentence "Air U.K. Ltd, a subsidiary of . . ."). In our model, this feature is positively correlated with the *acquired* role. In order to compensate for parsing errors and to increase coverage, shallow syntactic features were added, representing the values of neighboring verbs and the preceding preposition [Cohen et al. 2005]; for instance, the following features are derived from the header of the document displayed in Figure 14: *purchaser.rightVerb.take*, *acquired.leftVerb.take*, and *acquired.leftVerbPreposition.take.in*.

Semantic features. These features pertain to *corp* tuples, modeling string similarity between the values of the full name, abbreviated name and code fields. Specifically, we apply the Jaro-Winkler similarity measure [Cohen et al. 2003] to determine whether a string pair is similar, where a threshold on the similarity score was empirically set to 0.8. Additional features model domain-specific similarity, namely, whether the abbreviated name has common tokens with the full name, whether the corporate code forms exact initials of the full or abbreviated names, and whether their string values have the same prefix. For example, consider the correctly extracted *corp* tuple {"KLM Royal Dutch Airlines", "KLM", "KLM.A"} (Figure 14). Semantic features that describe this tuple include *corp.abr.name.commonTokens*, *corp.abbr.code.samePrefix*, *corp.abbr.code.similar*, and so forth.

Structural features. While newswire documents are mostly unstructured, the articles in the corporate acquisition corpus all include a one-line uppercased header, followed by city and date coordinates (see Figure 14). Also, it is generally acknowledged that the first sentence of a newswire article typically gives a concise summary of the article's content [Mani 2001]. In this domain, we observed that the header, as well as the first line of content, often mention the names of the parties involved in the reported acquisition. We therefore devised features indicating whether any of the text spans that map to the *purchaser*, *acquired*, and *seller* fields appear in the article's header, or in the consecutive first line of its content. For example, in Figure 14, the *purchaser*, *acquired*, and *seller* parties are all mentioned in the first sentence of the article (where the purchaser and acquired parties are also mentioned in the header). This information is represented by the following features: *acq.purchaser.inHeader*, *acq.acquired.inHeader* and *acq.seller.inHeader*.

We further apply the features that describe structural properties of the unified entity sets that map to each field in the populated relational schema. These features are described in detail in Section 5.3 (see also Table 6, feature groups s8–s10).

Finally, we model cross-field features, encoding the shortest string between text spans that map to different roles of the *acquisition* relation. Similar features are often used in relation extraction from free text [Mintz et al. 2009], as they often indicate the semantic relationship that holds between the pair of entity mentions. Here, we consider word sequences. For example, in Figure 14, the shortest string between the text spans that map to the *acquired* and *seller* roles is ", a subsidiary of"; this information is indicated by the feature *acquired.seller.shortestPath., a subsidiary of*. Alternatively, it is possible to represent the lexical relation between mentions, if they reside within the same sentence, in terms of the connecting dependency path.

In general, as illustrated in our review of the two case studies, various task- and genre-specific features may be modeled using the proposed structured learning framework. Many of the feature templates that are described in this article are applicable across domain and genre. For example, the lexical content and pattern features that

describe field values (Table 6) are generic, and have been used in both event extraction tasks. (We did make use of small hand-crafted dictionaries, for example, of room names or suffixes of corporate names; the construction of such lists typically requires very little effort.) Similarly, strictly lexical, or lexico-syntactic features, may be used to encode contextual information for semistructured and structured texts, respectively, as demonstrated in this work. Some of the structural features included in Table 6 are generic (feature groups s8–s10 in the table), whereas the remaining structural features describe properties of field value mentions with respect to document layout, and are genre-related. In general, once features are designed that capture typical phenomena of a text genre of interest, they can be applied to other extraction tasks from documents of that genre. We believe that the feature sets designed for email and newswire that are described in this article can be readily used in other record extraction tasks from email messages and news articles. In contrast, semantic features describe relevant aspects of the subject domain, and may need to be redesigned given a new task. We find that domain knowledge that was encoded as semantic features per the two domains considered in this work is quite intuitive, and does not require special expertise. Nevertheless, if such expertise is needed, then relevant knowledge, reasoning about individual fields, or global relationships among multiple fields, may be represented in this framework.

7.4. A Summary of the Experimental Results

The empirical results that were obtained on the acquisition event extraction task using a benchmark dataset [Freitag and McCallum 2000], and their comparison against previous published results, are reported elsewhere [Minkov and Zettlemoyer 2012].

In summary, overall performance on the corporate acquisition extraction task is generally lower compared with seminar extraction. There are several reasons for this lower performance. First, corporate names typically correspond to complex multitoken names, so that recovering their correct boundaries is challenging. In addition, once *corporate* tuples are constructed, they need to be assigned to three different role fillers in the *acquisition* relation based solely on contextual evidence. As free text uses highly diverse language, the contextual evidence learned from the set of labeled examples is sparse. In comparison, semistructured text such as email uses more regular language. The learning of the contextual cues involved in *corp* role assignments should improve given a larger corpus of labeled examples.

An ablation study showed that the *semantic* features, which account for the semantic cohesiveness of the populated *corp* tuples, were highly effective in this case. These features allow joint prediction of the abbreviated names and the full corporate names, considering various types of similarities between the respective string values. These features were found to be particularly beneficial for the extraction of the abbreviated names, as the full corporate names are more regular, typically including a distinctive suffix, and so forth. The contribution of the *structural* features, most of which pertain to document layout, was found to be mild in this case. This is not surprising, as the underlying news articles consist mostly of free text. Finally, the empirical study showed that, inferring the roles jointly, versus populating the fields of the target relation individually, significantly improved performance.

8. BEAM SEARCH: PERFORMANCE AND SCALABILITY CONSIDERATIONS

In this section, we examine the effect of varying the beam size k on extraction performance. Using beam search allows one to restrict the search space explored, focusing solely on the most promising candidates. This procedure therefore promotes efficiency, possibly at the cost of accuracy. Tuning the beam size parameter k provides control over the trade-off between performance quality and computation cost.

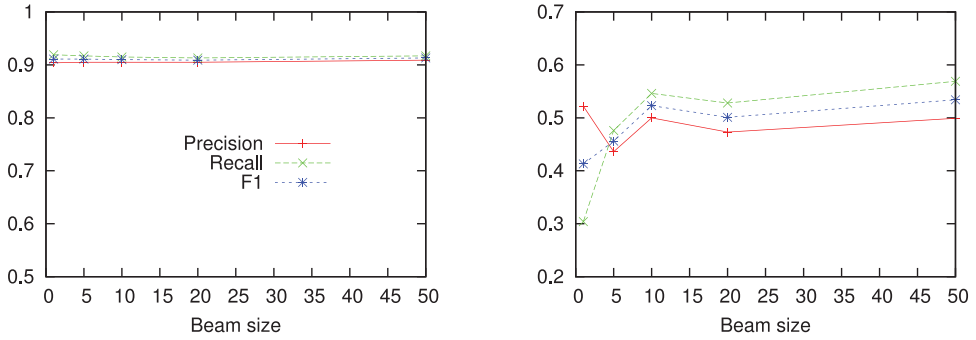


Fig. 17. Macro performance (averaged over all fields) at varying beam size k, presented for the CMU seminar-extraction (left) and corporate-acquisition (right) datasets.

Figure 17 displays our results using the beam size values $k = \{1, 5, 10, 20, 50\}$ on the CMU seminar-announcement (left) and corporate-acquisitions (right) datasets. In each individual experiment, a ranked list of k extracted records is generated. The figure shows precision, recall, and $F1$ scores of the top-scoring candidate at macro level, that is, averaged across all of the fields of the target template. The results for seminar extraction correspond to 5-fold cross-validation, and the results for corporate acquisition are evaluated based on the fixed test set; the curve for seminars is therefore smoother. As shown, increasing the beam size contributes little to performance on the seminar extraction problem. In contrast, a beam size of $k = 10$ or more is required to reach the reported level of performance in the corporate acquisition domain. The main reason for this is that a greater level of interaction exists between the fields of the acquisition template; in order to find the best joint assignment of its three *corp* role fillers, it is beneficial to examine a larger number of possible field assignment combinations. The semantic correspondence that is modeled between the seminar template slots is limited; concretely, it describes the precedence relationship between the *stime* and *etime* field values. The extraction performance of these two fields is near-perfect also when using a beam size $k = 1$, and the selection of the top-scoring candidate is hardly affected by increasing the beam size. Overall, the improvements beyond $k = 10$ are small in both cases. We set the beam size to $k = 10$ in our experiments.

An advantage of the beam search inference process is that it yields a k-ranked list of candidate tuples. Our ultimate goal is to obtain coherent predictions that are globally accurate. Given a high-quality ranked list of candidates–that is, a list that contains a globally correct candidate with high probability–one may apply techniques such as *reranking* to improve the output rankings [Cohen et al. 1999; Collins and Koo 2005; Minkov and Cohen 2010]. Or, assuming a semiautomatic scenario, high-quality ranked lists produced may be further processed manually, as in information retrieval settings.

We also assess the *global* quality of the ranked candidates. As opposed to the field-level performance reported so far, this stricter evaluation mode requires the values assigned to *all* of the fields of the populated template to be correct [Cohen et al. 2005]. Figure 18 shows global recall at different ranks of the output lists, varying the beam size $k = 1, 5, 10, 20, 50$, for the CMU seminar-announcement (left) and corporate-acquisition (right) datasets. As reflected in the figure, it is often the case that the globally correct candidate is ranked at the top of the list, albeit not in the first rank. For example, using a beam search of size $k = 1$ on the CMU seminar extraction dataset, the top rank includes the prediction of the gold standard record in 52% of the cases. Recall increases to 70% when the top-5-ranked candidates are considered, and to 76% among the top 10 candidates. Recall increases up to a high of 88% using beam size $k = 50$.

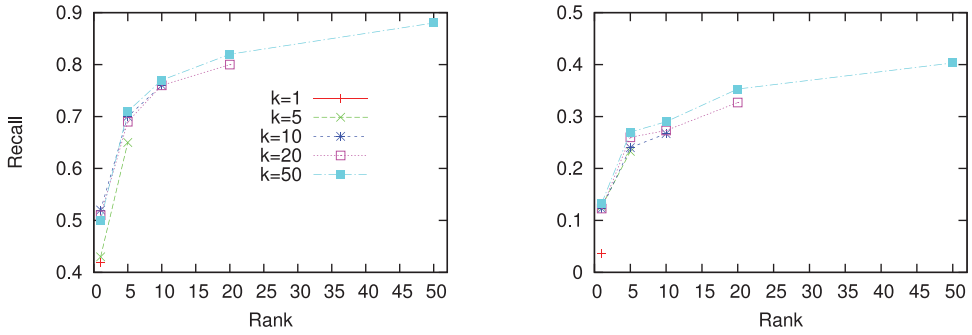


Fig. 18. Global recall versus rank curves at varying beam size k, presented for the CMU seminar-extraction (left) and corporate-acquisition (right) datasets.

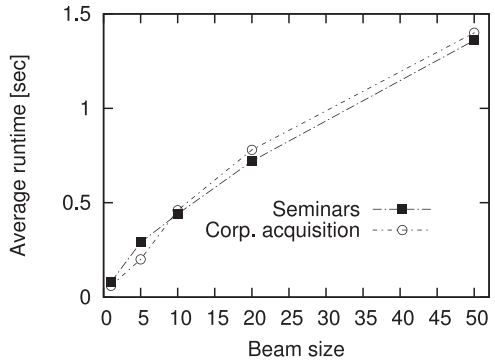


Fig. 19. Average test runtime per example at varying beam size k, presented for the CMU seminar-extraction and corporate-acquisition datasets.

Global recall on the corporate acquisition datasets is substantially lower, reflecting the lower accuracy in recovering the values of the individual fields in this domain, as well as the higher complexity of the underlying relational schema. However, similar trends are observed in this case: using $k = 1$, a fully correct record is extracted at the top rank in 12.3% of the examples. Recall doubles at rank 5 to 24% and reaches 26.7% at rank 10. Maximum recall on this dataset using $k = 50$ is as high as 40.3%. We find these results to be very encouraging, as we believe that postprocessing of the ranked lists using additional high-level information may further promote the rankings of the correctly extracted tuples in these lists. We discuss this possibility in more detail in Section 9.

While increasing the beam size improves performance and overall success in generating a globally correct record candidate, this comes with a cost. Figure 19 shows the average processing (inference) time of individual examples for the two datasets, varying the beam size. These runtimes were obtained using a commodity PC with 8GB of RAM. A fraction of a second was required, on average, using a beam of size of 5 or under, rising to about 1.5s using a beam of size 50, on both datasets. The space of candidates considered per example is polynomial with respect to the beam size, at the order of $O(k^2)$ (Section 4.3). Memory requirements must therefore be met in order to further increase the beam size. Memory as well as runtime limitations may be alleviated by means of parallel computing, however [McDonald et al. 2010].

Overall, beam search is effective in the evaluated settings, yielding high performance at a manageable cost. We identified two main sources of errors that negatively affected

performance in our experiments. First, beam search is a pipelined inference procedure; if a correctly populated candidate is not included within the top k scoring candidates at one of the inference steps, this results in error propagation. Further, the input to the process of record extraction using beam search is a noisy set of named entity mentions extracted from the given document using independent NER techniques. If NER fails to identify the correct field mentions, this may result in failure of record extraction. As discussed earlier, we recommend addressing this latter weakness by tweaking NER to yield high recall.

9. CONCLUSION

We described a discriminative approach for record extraction that models mention detection, unification, and record extraction as a structured prediction problem. Using this framework, it is possible to consider complex semantic features, defined over an extended relational schema describing the domain at hand. In addition, high-level features that pertain to document structure and discourse may be incorporated. Importantly, this approach enables the modeling of interdependencies at all levels and across fields.

We argued and demonstrated empirically that this approach produces record extraction models that are characterized with improved generalization. The task of seminar extraction from email announcements was considered as a case study in this article. Models learned and evaluated on the benchmark CMU datasets were shown to outperform other methods. We have further shown, through the evaluation of the learned models on a different set of seminar announcements at MIT, that the framework is especially advantageous in real-world conditions, in which data shift occurs.

Special attention was dedicated to practical considerations concerning the design procedure and cost involved in applying this framework to multiple tasks; in particular, the encoding of domain knowledge and the design of semantic and structural features require manual intervention. We outlined and discussed the representation of domain knowledge and feature encodings designed for the seminar-extraction task. We further discussed the adaptation of the framework to the task of corporate acquisition-event extraction from news articles, a task that involves different domain knowledge and text genre.

While the design of domain-specific relational knowledge in the form of relational schema and semantic features involves human effort, it has been clearly demonstrated that the representation of such domain knowledge results in performance gains and improved model generalization in conditions of data shift. We find that the proposed approach forms a cost-effective alternative to human effort that must be invested otherwise in annotating a large number of examples of the target data distribution in order to reach similar levels of performance.

We further examined in this article the computational cost that is involved in applying this approach. Importantly, despite the computational challenges of the large inference space considered, we obtained effective learning with a perceptron-style approach. The application of relational constraints in the framework serve to limit the search space as well, while ensuring that the generated candidates are coherent and valid. Finally, applying simple beam decoding enables control of the explored search space. A set of experimental results was provided, illustrating the effectiveness and efficiency of the approach computationally.

There are several directions of future research that we are interested in pursuing. First, modeling feature combinations or additional features in this framework, as well as experimenting with effective feature selection or improved parameter estimation together with the perceptron model [Crammer et al. 2009], may boost performance. Second, as mentioned in Section 8, we believe that further processing of the ranked

list of candidate records that is output by the beam search procedure, using additional evidence, may improve performance. Specifically, relevant evidence may be sought on the Web, using techniques that are commonly used by open information extraction systems [Banko et al. 2008]. For example, given candidate-populated templates, by which a (directed) *acquisition* relation holds between *company A* and *company B*, or vice versa, one can obtain Web statistics that support, or defy, these beliefs [Samadi et al. 2013]. Finally, it is worth exploring scaling the approach to unrestricted event extraction, and jointly model extracting more than one relation per document.

REFERENCES

Harith Alani, Sanghee Kim, David E. Millard, Mark J. Weal, Wendy Hall, Paul H. Lewis, and Nigel R. Shadbolt. 2003. Automatic ontology-based knowledge extraction from Web documents. *IEEE Intelligent Systems* 18, 1.

Galia Angelova. 2010. Use of domain knowledge in the automatic extraction of structured representations from patient-related texts. *Conceptual Structures: From Information to Intelligence, Lecture Notes in Computer Science,* vol. 6208, Springer, Berlin, 14–27.

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. 2008. Open information extraction from the web. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

Mikhail Bilenko, Raymond Mooney, William Cohen, Pradeep Ravikumar, and Stephen Fienberg. 2003. Adaptive name matching in information integration. *IEEE Intelligent Systems* 18, 5.

Mary Elaine Califf and Raymond J. Mooney. 1999. Relational learning of pattern-match rules for information extraction. In *AAAI/IAAI*.

Mary Elaine Califf and Raymond J. Mooney. 2003. Bottom-up relational learning of pattern matching rules for information extraction. *Journal of Machine Learning Research* 4.

Andrew Carlson, Justin Betteridge, Richard C. Wang, Estevam R. Hruschka Jr., and Tom M. Mitchell. 2010. Coupled semi-supervised learning for information extraction. In *Proceedings of WSDM*.

Ming-Wei Chang, Lev Ratinov, and Dan Roth. 2012. Structured learning with constrained conditional models. *Machine Learning* 88, 3.

William W. Cohen, Einat Minkov, and Anthony Tomasic. 2005. Learning to understand web site update requests. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

William W. Cohen, Pradeep Ravikumar, and Stephen Fienberg. 2003. A comparison of string distance metrics for name-matching tasks. In *IIWEB*.

William W. Cohen, Robert E. Schapire, and Yoram Singer. 1999. Learning to order things. *Journal of Artificial Intelligence Research (JAIR)* 10, 243–270.

Michael Collins. 2002. Discriminative training methods for Hidden Markov Models: Theory and experiments with perceptron algorithms. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Michael Collins and Terry Koo. 2005. Discriminative reranking for natural language parsing. *Computational Linguistics* 31, 1, 25–69.

Michael Collins and Brian Roark. 2004. Incremental parsing with the perceptron algorithm. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL)*.

Christopher Cox, Jamie Nicolson, Jenny Finkel, Christopher Manning, and Pat Langley. 2005. Template sampling for leveraging domain knowledge in information extraction. In *First PASCAL Challenges Workshop*.

Koby Crammer, Alex Kulesza, and Mark Dredze. 2009. Adaptive regularization of weight vectors. In *Advances in Neural Information Processing Systems (NIPS)*.

Jeremiah Crim, Ryan McDonald, and Fernando Pereira. 2005. Automatically annotating documents with normalized gene lists. *BMC Bioinformatics* 6, 1.

Nikolai Daraselia, Anton Yuryev, Sergei Egorov, Svetalana Novichkova, Alexander Nikitin, and Ilya Mazo. 2004. Extracting human protein interactions from MedLine using a full-sentence parser. *BioInformatics* 20, 5.

Hal Daumé III. 2007. Frustratingly easy domain adaptation. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL)*.

M. C. de Marneffe, B. Maccartney, and C. D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *LREC*.

Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S. Weld. 2008. Open information extraction from the web. *Communications of the ACM* 51, 12.

Jenny Rose Finkel, Trond Grenager, and Christopher D. Manning. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of ACL*.

Aidan Finn. 2006. A multi-level boundary classification approach to information extraction. In *PhD thesis*. University College Dublin, Ireland.

Dayne Freitag. 2000. Machine learning for information extraction in informal domains. *Machine Learning* 39, 2/3.

Dayne Freitag and Andrew McCallum. 2000. Information extraction with HMM structures learned by stochastic optimization. In *AAAI/IAAI*.

Yoav Freund and Rob Schapire. 1999. Large margin classification using the perceptron algorithm. *Machine Learning* 37, 3.

Ralph Grishman and Beth Sundheim. 1996. Message understanding conference-6: A brief history. In *16th International Conference on Computational Linguistics (COLING)*.

Aria Haghighi and Dan Klein. 2010. An entity-level approach to information extraction. In *Proceedings of ACL*.

Liang Huang, Suphan Fayong, and Yang Guo. 2012. Structured perceptron with inexact search. In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Zhenzhen Kou, William W. Cohen, and Robert F. Murphy. 2005. High-recall protein entity recognition using a dictionary. *BIOINFORMATICS* 21, 1.

Alberto Lavelli, Mary Elaine Califf, Fabio Ciravegna, Dayne Freitag, Claudio Giuliano, Nicholas Kushmerick, Lorenza Romano, and Neil Ireson. 2008. Evaluation of machine learning-based information extraction algorithms: Criticisms and recommendations. *Language Resources and Evaluation* 42, 4.

Xiao Ling and Daniel S. Weld. 2012. Fine-grained entity recognition. In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*.

Inderjeet Mani. 2001. *Automatic Summarization*. John Benjamins Publishing Co., Philadelphia, PA.

Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Online large-margin training of dependency parsers. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*.

Ryan McDonald, Keith Hall, and Gideon Mann. 2010. Distributed training strategies for the structured perceptron. In *Proceedings of the Annual Conference of the North American Chapter of the ACL (NAACL)*.

Einat Minkov, Ben Charrow, Jonathan Ledlie, Seth Teller, and Tommi Jaakkola. 2010. Collaborative future event recommendation. In *Proceedings of the ACM International Conference on Information Management and Knowledge Management (CIKM)*.

Einat Minkov and William W. Cohen. 2010. Improving graph-walk-based similarity with reranking: Case studies for personal information management. *Transactions on Information Systems (TOIS)* 29, 1.

Einat Minkov, Richard C. Wang, and William W. Cohen. 2005. Extracting personal names from emails: Applying named entity recognition to informal text. In *HLT/EMNLP*.

Einat Minkov, Richard C. Wang, Anthony Tomasic, and William W. Cohen. 2006. NER systems that suit user's preferences: Adjusting the recall-precision trade-off for entity extraction. In *HLT/NAACL*.

Einat Minkov and Luke S. Zettlemoyer. 2012. Discriminative learning for joint template filling. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Minorthird. 2008. Methods for identifying names and ontological relations in text using heuristics for inducing regularities from data. Retrieved November 4, 2015 from http://sourceforge.net/projects/minorthird/.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*.

Raymond J. Mooney and Razvan C. Bunescu. 2005. Mining knowledge from text using information extraction. *SIGKDD Explorations* 7, 1.

Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering* 22, 10.

Jun-geun Park, Ben Charrow, Dorothy Curtis, Jonathan Battat, Einat Minkov, Jamey Hicks, Seth Teller, and Jonathan Ledlie. 2010. Growing an organic indoor location system. In *Proceedings of the International Conference on Mobile Systems, Applications and Services (MobiSys)*.

Leonid Peshkin and Avi Pfeffer. 2003. Bayesian information extraction network. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of COLING-ACL*.

Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence. 2009. *Dataset Shift in Machine Learning*. MIT Press, Cambridge, MA.

Frank Rosenblatt. 1958. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review* 65, 6, 386–408.

Dan Roth and Wen-tau Yih. 2001. Relational learning via propositional algorithms: An information extraction case study. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

Dan Roth and Wen-tau Yih. 2002. Probabilistic reasoning for entity and relation recognition. In *COLING*.

Mehdi Samadi, Manuela Veloso, and Manuel Blum. 2013. OpenEval: Web information query evaluation. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence (AAAI)*.

Sunita Sarawagi. 2008. Information extraction. *Foundations and Trends in Databases* 1, 2.

Karl Michael Schneider. 2006. Information extraction from calls for papers with conditional random fields and layout features. *Artificial Intelligence Review* 25, 1–2.

Christian Siefkes. 2008. *An Incrementally Trainable Statistical Approach to Information Extraction*. VDM Verlag, Saarbrücken, Germany.

René Speck and Axel-Cyrille Ngonga Ngomo. 2014. Ensemble learning for named entity recognition. In *Proceedings of the International Semantic Web Conference, Lecture Notes in Computer Science*.

Charles Sutton and Andrew McCallum. 2004. Collective segmentation and labeling of distant entities in information extraction. In Technical Report no. 04-49, University of Massachusetts, Amherst MA, USA.

Jordi Turmo, Alicia Ageno, and Neus Català. 2006. Adaptive information extraction. *Computing Surveys* 38, 2.

Limin Yao, Aria Haghighi, Sebastian Riedel, and Andrew McCallum. 2011. Structured relation discovery using generative models. In *Proceedings of EMNLP*.

Luke S. Zettlemoyer and Michael Collins. 2009. Learning context-dependent mappings from sentences to logical form. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP (ACL-AFNLP)*.

Kaixu Zhang, Jinsong Su, and Changle Zhou. 2014. Regularized structured perceptron: A case study on Chinese word segmentation, POS tagging and parsing. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Yue Zhang and Stephen Clark. 2011. Syntactic processing using the generalized perceptron and beam search. *Computational Linguistics* 37, 1.

Jun Zhu, Zaiqing Nie, Ji-Rong Wen, Bo Zhang, and Wei-Ying Ma. 2006. Simultaneous record detection and attribute labeling in web data extraction. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*.

Received September 2014; revised May 2015; accepted July 2015