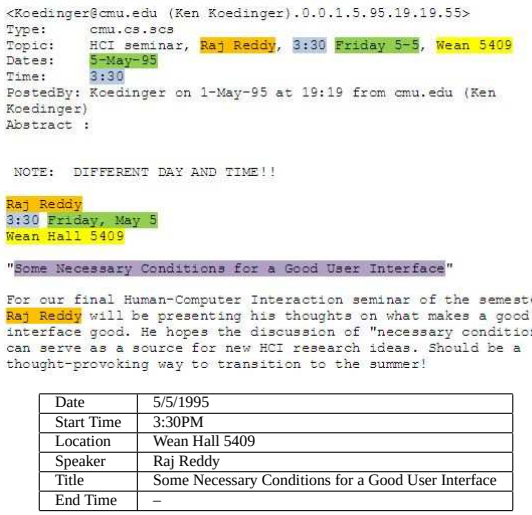
Discriminative Learning for Joint Template Filling

Einat Minkov
Information Systems
University of Haifa
Haifa 31905, Israel
einatm@is.haifa.ac.il

Luke Zettlemoyer
Computer Science & Engineering
University of Washington
Seattle, WA 98195, USA
lsz@cs.washington.edu

Abstract

This paper presents a joint model for template filling, where the goal is to automatically specify the fields of target relations such as seminar announcements or corporate acquisition events. The approach models mention detection, unification and field extraction in a flexible, feature-rich model that allows for joint modeling of interdependencies at all levels and across fields. Such an approach can, for example, learn likely event durations and the fact that start times should come before end times. While the joint inference space is large, we demonstrate effective learning with a Perceptron-style approach that uses simple, greedy beam decoding. Empirical results in two benchmark domains demonstrate consistently strong performance on both mention detection and template filling tasks.



<Koedinger@cmu.edu (Ken Koedinger).0.0.1.5.95.19.19.55>
Type: cmu.cs.scs
Topic: HCI seminar, Raj Reddy, 3:30 Friday 5-5, Wean 5409
Dates: 5-May-95
Time: 3:30
PostedBy: Koedinger on 1-May-95 at 19:19 from cmu.edu (Ken Koedinger)
Abstract :

NOTE: DIFFERENT DAY AND TIME!!

Raj Reddy
3:30 Friday, May 5
Wean Hall 5409

"Some Necessary Conditions for a Good User Interface"

For our final Human-Computer Interaction seminar of the semester, Raj Reddy will be presenting his thoughts on what makes a good interface good. He hopes the discussion of "necessary conditions" can serve as a source for new HCI research ideas. Should be a thought-provoking way to transition to the summer!

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	–

Figure 1: An example email and its template. Field mentions are highlighted in the text, grouped by color.

1 Introduction

Information extraction (IE) systems recover structured information from text. Template filling is an IE task where the goal is to populate the fields of a target relation, for example to extract the attributes of a job posting (Califf and Mooney, 2003) or to recover the details of a corporate acquisition event from a news story (Freitag and McCallum, 2000).

This task is challenging due to the wide range of cues from the input documents, as well as non-textual background knowledge, that must be considered to find the best joint assignment for the fields of the extracted relation. For example, Figure 1 shows an extraction from CMU seminar announcement corpus (Freitag and McCallum, 2000). Here, the goal is to perform mention detection and extraction, by finding all of the text spans, or *mentions*, that describe field values, unify these mentions by grouping them according to target field, and normalizing the results within each group to provide the final extractions. Each of these steps requires significant knowledge about the target relation. For example, in Figure 1, the mention "3:30" appears three times and provides the only reference to a time. We must infer that this is the starting time, that the end time is never explicitly mentioned, and also that the event is in the afternoon. Such inferences may not hold in more general settings, such as extraction for medical emergencies or related events.

In this paper, we present a joint modeling and learning approach for the combined tasks of mention detection, unification, and template filling, as described above. As we will see in Section 2, previous work has mostly focused on learning tagging

Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, pages 845–853,
Jeju, Republic of Korea, 8-14 July 2012. ©2012 Association for Computational Linguistics

models for mention detection, which can be difficult to aggregate into a full template extraction, or directly learning template field value extractors, often in isolation and with no reasoning across different fields in the same relation. We present a simple, feature-rich, discriminative model that readily incorporates a broad range of possible constraints on the mentions and joint field assignments.

Such an approach allows us to learn, for each target relation, an integrated model to weight the different extraction options, including for example the likely lengths for events, or the fact that start times should come before end times. However, there are significant computation challenges that come with this style of joint learning. We demonstrate empirically that these challenges can be solved with a combination of greedy beam decoding, performed directly in the joint space of possible mention clusters and field assignments, and structured Perceptron-style learning algorithm (Collins, 2002).

We report experimental evaluations on two benchmark datasets in different genres, the CMU seminar announcements and corporate acquisitions (Freitag and McCallum, 2000). In each case, we evaluated both template extraction and mention detection performance. Our joint learning approach provides consistently strong results across every setting, including new state-of-the-art results. We also demonstrate, through ablation studies on the feature set, the need for joint modeling and the relative importance of the different types of joint constraints.

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, where the most likely entity is assigned to each slot (Roth and Yih, 2001; Siefkes, 2008). There has been little effort towards a comprehensive approach that includes mention unification, as well as considers the structure of the target relational schema to create semantically valid outputs.

Recently, Haghighi and Klein (2010) presented a generative semi-supervised 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. This work presents a discriminative approach to this 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 IE*, entities and relations may be inferred jointly (Roth and Yih, 2002; Yao et al., 2011). In this IE task, the target relation must agree with the entity types assigned to it; e.g., *born-in* relation requires a *place* as its argument. In addition, extracted relations may be required to be consistent with an existing ontology (Carlson et al., 2010). Compared with the extraction of tuples of entity mention pairs, template filling is associated with a more complex target relational schema.

Interestingly, several researchers have attempted to model label consistency and high-level relational constraints using state-of-the-art sequential models of named entity recognition (NER). Mainly, predetermined word-level dependencies were represented as links in the underlying graphical model (Sutton and McCallum, 2004; Finkel et al., 2005). Finkel *et al.* (2005) further modelled 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. Our approach is complimentary to NER methods, as it can consolidate noisy overlapping predictions from multiple systems into coherent sets.

3 Problem Setting

In the template filling task, a target relation r is provided, comprised of attributes (also referred to as

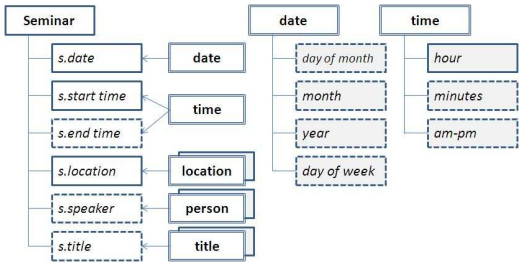


Figure 2: The relational schema for the seminars domain.

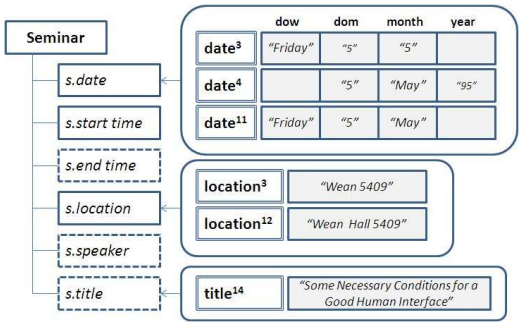


Figure 3: A record partially populated from text.

fields, or slots) $A(r)$. Given a document d, which is known to describe a tuple of the underlying relation, the goal is to populate the fields with values based on the text.

The relational schema. In this work, we describe domain knowledge through an extended relational database schema R. In this schema, every field of the target relation maps to a tuple of another relation, giving rise to a hierarchical view of template filling. Figure 2 describes a relational schema for the seminar announcement domain. As shown, each field of the *seminar* relation maps to another relation; e.g., *speaker*'s values correspond to *person* tuples. According to the outlined schema, most relations (e.g., *person*) consist of a single attribute, whereas the *date* and *time* relations are characterised with multiple attributes; for example, the *time* relation includes the fields of *hour*, *minutes* and *ampm*.

We will make use of limited domain knowledge, expressed as relation-level constraints that are typically realized in a database. Namely, the following tests are supported for each relation.

Tuple validity. This test reflects data integrity. The attributes of a relation may be defined as *mandatory* or *optional*. Mandatory attributes are denoted with a solid boundary in Figure 2 (e.g., *seminar.date*), and optional attributes are denoted with a dashed boundary (e.g., *seminar.title*). Similar constraints can be posed on a set of attributes; e.g., either *day-of-month* or *day-of-week* must be populated in the *date* relation. Finally, a combination of field values may be required to be valid, e.g., the values of *day*, *month*, *year* and *day-of-week* must be consistent.

Tuple contradiction. This function checks whether two *valid* tuples v_1 and v_2 are inconsistent, implying a negation of possible unification of these tuples. In this work, we consider *date* and *time* tuples as contradictory if they contain semantically different values for some field; tuples of *location*, *person* and *title* are required to have minimal overlap in their string values to avoid contradiction.

Template filling. Given document d, the hierarchical schema R is populated in a bottom-up fashion. Generally, parent-free relations in the hierarchy correspond to generic entities, realized as entity mentions in the text. In Figure 2, these relations are denoted by double-line boundary, including *location*, *person*, *title*, *date* and *time*; every tuple of these relations maps to a named entity mention.[1]

Figure 3 demonstrates the correct mapping of named entity mentions to tuples, as well as tuple unification, for the example shown in Figure 1. For example, the mentions "Wean 5409" and "Wean Hall 5409" correspond to tuples of the *location* relation, where the two tuples are resolved into a unified set. To complete template filling, the remaining relations of the schema are populated bottom-up, where each field links to a unified set of populated tuples. For example, in Figure 3, the *seminar.location* field is linked to {"Wean Hall 5409","Wean 5409"}.

Value normalization of the unified tuples is another component of template filling. We partially address normalization: tuples of semantically detailed (multi-attribute) relations, e.g., *date* and *time*, are resolved into their semantic union, while textual tuples (e.g., *location*) are normalized to the longest string in the set. In this work, we assume that each template slot contains at most one value. This restriction can be removed, at the cost of increasing the size of the decoding search space.

[1] In the multi-attribute relations of *date* and *time*, each attribute maps to a text span, where the set of spans at tuple-level is required to be sequential (up to a small distance d).

4 Structured Learning

Next, we describe how valid candidate extractions are instantiated (Sec. 4.1) and how learning is applied to assess the quality of the candidates (Sec. 4.2), where beam search is used to find the top scoring candidates efficiently (Sec. 4.3).

4.1 Candidate Generation

Named entity recognition. A set of candidate mentions $S_d(a)$ is extracted from document d per each attribute a of a relation $r \in L$, where L is the set of parent-free relations in T. We aim at *high-recall* extractions; i.e., $S_d(a)$ is expected to contain the correct mentions with high probability. Various IE techniques, as well as an ensemble of methods, can be employed for this purpose. For each relation $r \in L$, *valid* candidate tuples $E_d(r)$ are constructed from the candidate mentions that map to its attributes.

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 candidate sets is exponential in the size of $E_d(t)$. Importantly, however, the tuples within a candidate unification set are required to be *non-contradictory*. 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 sets.

Candidate tuples. To construct the space of candidate tuples of the target relation, the remaining relations $r \in \{T-L\}$ are visited bottom-up, where each field $a \in A(r)$ is mapped in turn to a (possibly unified) populated tuple of its type. The valid (and non-overlapping) combinations of field mappings constitute a set of candidate tuples of r.

The candidate tuples generated using this procedure are structured entities, constructed using typed named entity recognition, unification, and hierarchical assignment of field values (Figure 3). We will derive features that describe local and global properties of the candidate tuples, encoding both surface and semantic information.

4.2 Learning

We employ a discriminative learning algorithm, following Collins (2002). Our goal is to find the candidate that maximizes:

$$F(y, \bar{\alpha}) = \sum_{j=1}^{m} \alpha_j f_j(y, d, T) \quad (1)$$

where $f_j(d, y, T)$, $j = 1, .., m$, are pre-defined feature functions describing a candidate record y of the target relation given document d and the extended schema T. The parameter weights α_j are to be learned from labeled instances. The training procedure involves initializing the weights $\bar{\alpha}$ to zero. Given $\bar{\alpha}$, an inference procedure is applied to find the candidate that maximizes Equation 1. If the top-scoring candidate is different from the correct mapping known, then: (i) $\bar{\alpha}$ is incremented with the feature vector of the correct candidate, and (ii) the feature vector of the top-scoring candidate is subtracted from $\bar{\alpha}$. This procedure is repeated for a fixed number of epochs. Following Collins, we employ the averaged Perceptron online algorithm (Collins, 2002; Freund and Schapire, 1999) for weight learning.

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 $S_d(a)$, $a \in A(r)$.
   - Group $E_d(r)$ into possibly overlapping unified sets, $\{C_d(r) \subseteq E_d(r)\}$.
2. Iterate bottom-up through relations $r \in \{T - L\}$:
   - 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 T. 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 $E_d(r^*)$, where r^* is the target relation.

4.3 Beam Search

Unfortunately, optimal local decoding algorithms (such as the Viterbi algorithm in tagging problems (Collins, 2002)) can not be applied to our problem. We therefore propose using beam search to efficiently find the top scoring candidate. This means

that rather than instantiate the full space of valid candidate records (Section 4.1), we are interested in instantiating only those candidates that are likely to be assigned a high score by F. Algorithm 1 outlines the proposed beam search procedure. As detailed, only a set of top scoring tuples of size k (beam size) is maintained per relation $r \in T$ during candidate generation. A given relation is populated incrementally, having each of its attributes $a \in A(r)$ map in turn to populated tuples of its type, and using Equation 1 to find the k highest scoring *partially* populated tuples; this limits the number of candidate tuples evaluated to k^2 per attribute, and to nk^2 for a relation with n attributes. While beam search is efficient, performance may be compromised compared with an unconstrained search. The beam size k allows controlling the trade-off between performance and cost. An advantage of the proposed approach is that rather than output a single prediction, a list of coherent candidate tuples may be generated, ranked according to Equation 1.

5 Seminar Extraction Task

Dataset The CMU seminar announcement dataset (Freitag and McCallum, 2000) includes 485 emails containing seminar announcements. The dataset has been originally annotated with text spans referring to four slots: *speaker*, *location*, *stime*, and *etime*. We have annotated this dataset with two additional attributes: *date* and *title*.[2] We consider this corpus as an example of semi-structured text, where some of the field values appear in the email header, in a tabular structure, or using special formatting (Califf and Mooney, 1999; Minkov et al., 2005).[3]

We used a set of rules to extract candidate named entities per the types specified in Figure 2.[4] The rules encode information typically used in NER, including content and contextual patterns, as well as lookups in available dictionaries (Finkel et al., 2005; Minkov et al., 2005). The extracted candidates are high-recall and overlapping. In order to increase recall further, additional candidates were extracted based on document structure (Siefkes, 2008). The recall for the named entities of type *date* and *time* is near perfect, and is estimated at 96%, 91% and 90% for *location*, *speaker* and *title*, respectively.

Features The categories of the features used are described below. All features are binary and typed.[5]

Lexical. These features indicate the value and pattern of words within the text spans corresponding to each field. For example, lexical features per Figure 1 include *location.content.word.wean*, *location.pattern.capitalized*. Similar features are derived for a window of three words to the right and to the left of the included spans. In addition, we observe whether the words that comprise the text spans appear in relevant dictionaries: e.g., whether the spans assigned to the location field include words typical of location, such as "room" or "hall". Lexical features of this form are commonly used in NER (Finkel et al., 2005; Minkov et al., 2005).

Structural. It has been previously shown that the structure available in semi-structured documents such as email messages is useful for information extraction (Minkov et al., 2005; Siefkes, 2008). As shown in Figure 1, an email message includes a header, specifying textual fields such as *topic*, *dates* and *time*. In addition, space lines and line breaks are used to emphasize blocks of important information. We propose a set of features that model correspondence between the text spans assigned to each field and document structure. Specifically, 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 space lines; or in a tabular format. In Figure 1, structural active features include *location.inHeader*, *location.fullLine*, *title.withinSpaceLines*, etc.

Semantic. These features refer to the semantic interpretation of field values. According to the relational schema (Figure 2), *date* and *time* include detailed attributes, whereas other relations are represented as strings. The semantic features encoded therefore refer to *date* and *time* only. Specifically, these features indicate whether a unified set of tuples defines a value for all attributes; for example, in Figure 1, the union of entities that map to the *date* field specify all of the attribute values of this relation, including *day-of-month*, *month*, *year*, and

[2] A modified dataset is available on the author's homepage.

[3] Such structure varies across messages. Otherwise, the problem would reduce to wrapper learning (Zhu et al., 2006).

[4] The rule language used is based on cascaded finite state machines (Minorthird, 2008).

[5] Real-value features were discretized into segments.

	Date	Stime	Etime	Location	Speaker	Title
Full model	96.1	99.3	98.7	96.4	87.5	69.5
No structural features	94.9	99.1	98.0	96.1	83.8	65.1
No semantic features	96.1	98.7	95.4	96.4	87.5	69.5
No unification	87.2	97.0	95.1	94.5	76.0	62.7
Individual fields	96.5	97.2	-	96.4	86.8	64.5

Table 1: Seminar extraction results (5-fold CV): Field-level F1

	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

Table 2: Seminar extraction results (5-fold CV, trained on 50% of corpus): Field-level F1

day-of-week. Another feature encodes the size of the most semantically detailed named entity that maps to a field; for example, the most detailed entity mention of type $stime$ in Figure 1 is "3:30", comprising of two attribute values, namely *hour* and *minutes*. Similarly, the total number of semantic units included in a unified set is represented as a feature. These features were designed to favor semantically detailed mentions and unified sets. Finally, domain-specific semantic knowledge is encoded as features, including the *duration* of the seminar, and whether a *time* value is round (minutes divide by 5).

In addition to the features described, 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 separating lines or sentences), based on the hypothesis that field values typically co-appear in the same segments 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.

Experiments We conducted 5-fold cross validation experiments using the seminar extraction dataset. As discussed earlier, we assume that a single record is described in each document, and that each field corresponds to a single value. These assumptions are violated in a minority of cases. In evaluating the template filling task, only exact matches are accepted as true positives, where partial matches are counted as errors (Siefkes, 2008). Notably, the annotated labels as well as corpus itself are not error-free; for example, in some announcements the date and day-of-week specified are inconsistent. Our evaluation is strict, where non-empty predicted values are counted as errors in such cases.

Table 1 shows the results of our full model using beam size $k = 10$, as well as model variants. In order to evaluate the contribution of the proposed features, we eliminated every feature group in turn. As shown in the table, removing the structural features hurt performance consistently across fields. In particular, structure is informative for the *title* field, which is otherwise characterised with low content and contextual regularity. Removal of the semantic features affected performance on the *stime* and *etime* 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 in evaluation is to what extent the joint modeling approach contributes to performance. In another experiment we therefore mimic the typical scenario of template filling, in which the value of the highest scoring named entity is assigned to each field. In our framework, this corresponds to a setting in which a unified set includes no more than a single entity. The results are shown in Table 1 ('no unification'). Due to reduced evidence given a single entity versus a a coreferent set of entities, this results in significantly degraded performance. 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 negative impact on the *title* field. This is largely due to erroneous assignments of named entities of other types (mainly, *person*) as titles; such errors are avoided in the full joint model, where tuple validity is enforced.

Table 2 provides a comparison of the full model

	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

Table 3: Seminar extraction results: Token-level F1

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; the reported figures were averaged over five random splits. For comparison, we used 5-fold cross validation, where only a subset of each train fold that corresponds to 50% of the corpus was used for training. Due to the reduced training data, the results are slightly lower than in Table 1. (Note that we used the same test examples in both cases.) The best results per field are marked in boldface. The proposed approach yields the best or second-best performance on all target fields, and gives the best performance overall. While a variety of methods have been applied in previous works, none has modeled template filling in a joint fashion. As argued before, joint modeling is especially important for 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 to perform NER on this dataset, identifying named entities that pertain to the template slots. Both of these works incorporated coreference and high-level semantic information to a limited extent. We compare our approach to their work, having obtained and used the same 5-fold cross validation splits as both works. Table 3 shows results in terms of token F1. Our results evaluated on the named mention recognition task are superior overall, giving comparable or best performance on all fields. We believe that these results demonstrate the benefit of performing mention recognition as part of a joint model that takes into account detailed semantics of the underlying relational schema, when available.

Finally, we evaluate the *global* quality of the extracted records. Rather than assess performance at field-level, this stricter evaluation mode considers a whole tuple, requiring the values assigned to all of its fields to be correct. Overall, our full model (Table 1) extracts globally correct records for 52.6% of the examples. To our knowledge, this is the first work that provides this type of evaluation on this dataset. Importantly, an advantage of the proposed approach

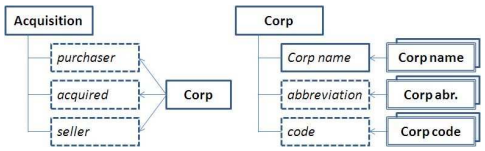


Figure 4: The relational schema for acquisitions.

is that it readily outputs a ranked list of coherent predictions. While the performance at the top of the output lists was roughly comparable, increasing k gives higher oracle recall: the correct record was included in the output k-top list 69.7%, 76.1% and 80.4% of the time, for $k = 5, 10, 20$ respectively.

6 Corporate Acquisitions

Dataset The corporate acquisitions corpus contains 600 newswire articles, describing factual or potential corporate acquisition events. The corpus has been annotated with the official names of the parties to an acquisition: *acquired*, *purchaser* and *seller*, as well as their corresponding abbreviated names and company codes.[6] We describe the target schema using the relational structure depicted in Figure 4. The schema includes two 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 linked to a *corp* tuple.

Candidate name mentions in this strictly grammatical genre correspond to *noun phrases*. Documents were pre-processed to extract noun phrases, similarly to Haghighi and Klein (2010).

Features We model *syntactic* features, following Haghighi and Klein (2010). In order to compensate for parsing errors, shallow syntactic features were added, representing the values of neighboring verbs and prepositions (Cohen et al., 2005). While newswire documents are mostly unstructured, *structural* features are used to indicate whether any of the *purchaser*, *acquired* and *seller* text spans appears in

[6] In this work, we ignore other fields annotated, as they are inconsistently defined, have low number of occurrences in the corpus, and are loosely inter-related semantically.

	purname	purabr	purcode	acqname	acqabr	acqcode	sellname	sellabr	sellcode
TIE (batch)	**55.7**	**58.1**	-	**53.5**	55.0	-	31.8	25.8	-
TIE (inc)	51.6	55.3	-	49.2	51.7	-	26.0	24.0	-
Full model	48.9	55.0	**70.2**	50.7	**55.2**	**67.2**	**33.2**	**36.8**	**55.4**
Model variants:									
No inter-type and struct. ftrs	45.1	50.5	66.8	49.8	53.9	66.4	34.9	42.2	56.0
No semantic features	42.6	38.4	58.1	40.5	36.5	44.8	32.2	26.6	46.6
Individual roles	43.9	48.7	62.5	45.0	47.2	52.7	34.1	40.3	47.8

Table 4: Corp. acquisition extraction results: Field-level F1

	purname	purabr	purcode	acqname	acqabr	acqcode	sellname	sellabr	sellcode
TIE (batch)	**52.6**	40.5	-	**49.2**	43.7		28.7	16.4	-
TIE (inc)	48.4	38.6	-	44.7	42.7	-	23.6	14.5	-
Full model	45.0	**48.3**	**69.8**	46.4	**59.5**	**66.9**	**31.6**	**33.0**	**55.0**

Table 5: Corp. acquisition extraction results: Entity-level F1

the article's header. *Semantic* features are applied to *corp* tuples: we model whether the abbreviated name is a subset of the full name; whether the corporate code forms exact initials of the full or abbreviated names; or whether it has high string similarity to any of these values. Finally, *cross-type features* encode the shortest string between spans mapping to different roles in the *acquisition* relation.

Experiments We applied beam search, where *corp* tuples are extracted first, and *acquisition* tuples are constructed using the top scoring *corp* entities. We used a default beam size $k = 10$. The dataset is split into a 300/300 train/test subsets.

Table 4 shows results of our full model in terms of field-level F1, compared against TIE, a state-of-the-art discriminative system (Siefkes, 2008). Unfortunately, we can not directly compare against a generative joint model evaluated on this dataset (Haghighi and Klein, 2010).[7] The best results per attribute are shown in boldface. Our full model performs better overall than TIE trained incrementally (similarly to our system), and is competitive with TIE using batch learning. Interestingly, the performance of our model on the *code* fields is high; these fields do not involve boundary prediction, and thus reflect the quality of role assignment.

Table 4 also shows the results of model variants. Removing the *inter type* and *structural* features mildly hurt performance, on average. In contrast, the *semantic* features, which account for the semantic cohesiveness of the populated *corp* tuples, are shown to be necessary. In particular, removing them degrades the extraction of the abbreviated names; these features allow prediction of abbreviated names jointly with the full corporate names, which are more regular (e.g., include a distinctive suffix). Finally, we show results of predicting each role filler individually. Inferring the roles jointly ('full model') significantly improves performance.

Table 5 further shows results on NER, the task of recovering the sets of named entity mentions pertaining to each target field. As shown, the proposed joint approach performs overall significantly better than previous results reported. These results are consistent with the case study of seminar extraction.

7 Summary and Future Work

We presented a joint approach for template filling that models mention detection, unification, and field extraction in a flexible, feature-rich model. This approach allows for joint modeling of interdependencies at all levels and across fields. Despite the computational challenges of this joint inference space, we obtained effective learning with a Perceptron-style approach and simple beam decoding.

An interesting direction of future research is to apply reranking to the output list of candidate records using additional evidence, such as supporting evidence on the Web (Banko et al., 2008). Also, modeling additional features or feature combinations in this framework as well as effective feature selection or improved parameter estimation (Crammer et al., 2009) may boost performance. Finally, it is worth exploring scaling the approach to unrestricted event extraction, and jointly model extracting more than one relation per document.

[7] They report average performance on a different set of fields; in addition, their results include modeling of pronouns and nominal mentions, which are not considered here.

References

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. 2008. Open information extraction from the web. In *Proceedings of IJCAI*.

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*.

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)*.

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)*.

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

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*.

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).

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

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*.

Minorthird. 2008. Methods for identifying names and ontological relations in text using heuristics for inducing regularities from data. `http://http://minorthird.sourceforge.net`.

Leonid Peshkin and Avi Pfeffer. 2003. Bayesian information extraction network. In *Proceedings of the international joint conference on Artificial intelligence (IJCAI)*.

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*.

Christian Siefkes. 2008. In *An Incrementally Trainable Statistical Approach to Information Extraction*. VDM Verlag.

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*.

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

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 *Proc. of the ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining (KDD)*.