An Entity-Level Approach to Information Extraction

Aria Haghighi
UC Berkeley, CS Division
aria42@cs.berkeley.edu

Dan Klein
UC Berkeley, CS Division
klein@cs.berkeley.edu

Abstract

We present a generative model of template-filling in which coreference resolution and role assignment are jointly determined. Underlying template roles first generate abstract entities, which in turn generate concrete textual mentions. On the standard corporate acquisitions dataset, joint resolution in our entity-level model reduces error over a mention-level discriminative approach by up to 20%.

Template

(a)

SELLER	BUSINESS	ACQUIRED	PURCHASER
CSR Limited	Oil and Gas	Delhi Fund	Esso Inc.

Document

(b) [**S** CSR] has said that [**S** it] has sold [**S** its] [**B** oil interests] held in [**A** Delhi Fund]. [**P** Esso Inc.] did not disclose how much [**P** they] paid for [**A** Dehli].

Figure 1: Example of the corporate acquisitions role-filling task. In (a), an example template specifying the entities playing each domain role. In (b), an example document with coreferent mentions sharing the same role label. Note that pronoun mentions provide direct clues to entity roles.

1 Introduction

Template-filling information extraction (IE) systems must merge information across multiple sentences to identify all role fillers of interest. For instance, in the MUC4 terrorism event extraction task, the entity filling the *individual perpetrator* role often occurs multiple times, variously as proper, nominal, or pronominal mentions. However, most template-filling systems (Freitag and McCallum, 2000; Patwardhan and Riloff, 2007) assign roles to individual textual mentions using only local context as evidence, leaving aggregation for post-processing. While prior work has acknowledged that coreference resolution and discourse analysis are integral to accurate role identification, to our knowledge no model has been proposed which jointly models these phenomena.

In this work, we describe an entity-centered approach to template-filling IE problems. Our model jointly merges surface mentions into underlying entities (coreference resolution) and assigns roles to those discovered entities. In the generative process proposed here, document entities are generated for each template role, along with a set of non-template entities. These entities then generate mentions in a process sensitive to both lexical and structural properties of the mention. Our model outperforms a discriminative mention-level baseline. Moreover, since our model is generative, it can naturally incorporate unannotated data, which further increases accuracy.

2 Problem Setting

Figure 1(a) shows an example *template-filling* task from the corporate acquisitions domain (Freitag, 1998).[1] We have a template of K roles (PURCHASER, AMOUNT, etc.) and we must identify which entity (if any) fills each role (*CSR Limited*, etc.). Often such problems are modeled at the mention level, directly labeling individual mentions as in Figure 1(b). Indeed, in this data set, the mention-level perspective is evident in the gold annotations, which ignore pronominal references. However, roles in this domain appear in several locations throughout the document, with pronominal mentions often carrying the critical information for template filling. Therefore, Section 3 presents a model in which entities are explicitly modeled, naturally merging information across all mention types and explicitly representing latent structure very much like the entity-level template structure from Figure 1(a).

[1] In Freitag (1998), some of these fields are split in two to distinguish a full versus abbreviated name, but we ignore this distinction. Also we ignore the *status* field as it doesn't apply to entities and its meaning is not consistent.

Proceedings of the ACL 2010 Conference Short Papers, pages 291–295,
Uppsala, Sweden, 11-16 July 2010.

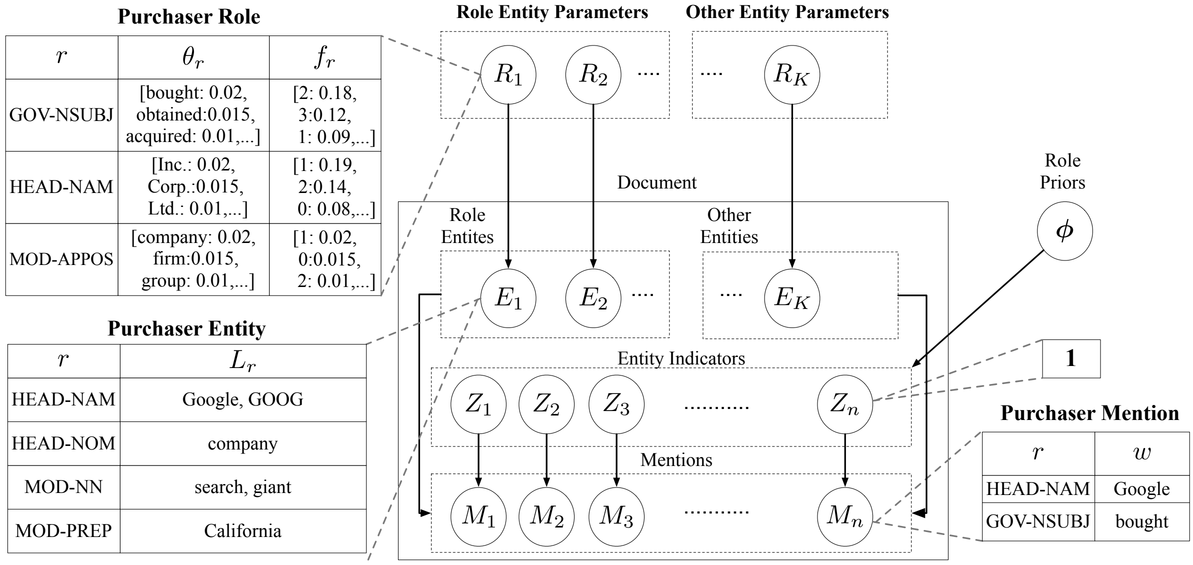

Figure 2: Graphical model depiction of our generative model described in Section 3. Sample values are illustrated for key parameters and latent variables.

3 Model

We describe our generative model for a document, which has many similarities to the coreference-only model of Haghighi and Klein (2010), but which integrally models template role-fillers. We briefly describe the key abstractions of our model.

Mentions: A mention is an observed textual reference to a latent real-world entity. Mentions are associated with nodes in a parse tree and are typically realized as NPs. There are three basic forms of mentions: proper (NAM), nominal (NOM), and pronominal (PRO). Each mention M is represented as collection of key-value pairs. The keys are called *properties* and the values are words. The set of properties utilized here, denoted $\mathcal{R}$, are the same as in Haghighi and Klein (2010) and consist of the mention head, its dependencies, and its governor. See Figure 2 for a concrete example. Mention types are trivially determined from mention head POS tag. All mention properties and their values are observed.

Entities: An entity is a specific individual or object in the world. Entities are always latent in text. Where a mention has a single word for each property, an entity has a *list* of signature words. Formally, entities are mappings from properties $r \in \mathcal{R}$ to lists L_r of "canonical" words which that entity uses for that property.

Roles: The elements we have described so far are standard in many coreference systems. Our model performs role-filling by assuming that each entity is drawn from an underlying role. These roles include the K template roles as well as 'junk' roles to represent entities which do not fill a template role (see Section 5.2). Each role R is represented as a mapping between properties r and pairs of multinomials (θ_r, f_r). θ_r is a unigram distribution of words for property r that are semantically licensed for the role (e.g., being the subject of "acquired" for the ACQUIRED role). f_r is a "fertility" distribution over the integers that characterizes entity list lengths. Together, these distributions control the lists L_r for entities which instantiate the role.

We first present a broad sketch of our model's components and then detail each in a subsequent section. We temporarily assume that all mentions belong to a template role-filling entity; we lift this restriction in Section 5.2. First, a semantic component generates a sequence of entities $\mathbf{E} = (E_1, \ldots, E_K)$, where each E_i is generated from a corresponding role R_i. We use $\mathbf{R} = (R_1, \ldots, R_K)$ to denote the vector of template role parameters. Note that this work assumes that there is a one-to-one mapping between entities and roles; in particular, at most one entity can fill each role. This assumption is appropriate for the domain considered here.

Once entities have been generated, a discourse component generates which entities will be evoked in each of the n mention positions. We represent these choices using *entity indicators* denoted by $\mathbf{Z} = (Z_1, \ldots, Z_n)$. This component utilizes a learned global prior ϕ over roles. The Z_i indicators take values in $1, \dots, K$ indicating the entity number (and thereby the role) underlying the ith mention position. Finally, a mention generation component renders each mention conditioned on the underlying entity and role. Formally:

$$P(\mathbf{E}, \mathbf{Z}, \mathbf{M} | \mathbf{R}, \phi) =$$

$$\left(\prod_{i=1}^{K} P(E_i|R_i)\right) \quad \text{[Semantic, Sec. 3.1]}$$

$$\left(\prod_{j=1}^{n} P(Z_j|\mathbf{Z}_{<j}, \phi)\right) \quad \text{[Discourse, Sec. 3.2]}$$

$$\left(\prod_{j=1}^{n} P(M_j|E_{Z_j}, R_{Z_j})\right) \quad \text{[Mention, Sec. 3.3]}$$

3.1 Semantic Component

Each role R generates an entity E as follows: for each mention property r, a word list, L_r, is drawn by first generating a list length from the corresponding f_r distribution in R.[2] This list is then populated by an independent draw from R's unigram distribution θ_r. Formally, for each $r \in \mathcal{R}$, an entity word list is drawn according to,[3]

$$P(L_r|R) = P(\texttt{len}(L_r)|f_r) \prod_{w \in L_r} P(w|\theta_r)$$

3.2 Discourse Component

The discourse component draws the entity indicator Z_j for the jth mention according to,

$$P(Z_j|\mathbf{Z}_{<j}, \phi) = \begin{cases} P(Z_j|\phi), \text{ if non-pronominal} \\ \sum_{j'} \mathbf{1}[Z_j = Z_{j'}] P(j'|j), \text{ o.w.} \end{cases}$$

When the jth mention is non-pronominal, we draw Z_j from ϕ, a global prior over the K roles. When M_j is a pronoun, we first draw an antecedent mention position j', such that $j' < j$, and then we set $Z_j = Z_{j'}$. The antecedent position is selected according to the distribution,

$$P(j'|j) \propto \exp\{-\gamma \textsc{TreeDist}(j', j)\}$$

where $\textsc{TreeDist}(j',j)$ represents the tree distance between the parse nodes for M_j and $M_{j'}$.[4] Mass is restricted to antecedent mention positions j' which occur earlier in the same sentence or in the previous sentence.[5]

3.3 Mention Generation

Once the entity indicator has been drawn, we generate words associated with mention conditioned on the underlying entity E and role R. For each mention property r associated with the mention, a word w is drawn utilizing E's word list L_r as well as the multinomials (f_r, θ_r) from role R. The word w is drawn according to,

$$P(w|E, R) = (1 - \alpha_r) \frac{\mathbf{1}[w \in L_r]}{\texttt{len}(L_r)} + \alpha_r P(w|\theta_r)$$

For each property r, there is a hyper-parameter α_r which interpolates between selecting a word uniformly from the entity list L_r and drawing from the underlying role distribution θ_r. Intuitively, a small α_r indicates that an entity prefers to re-use a small number of words for property r. This is typically the case for proper and nominal heads as well as modifiers. At the other extreme, setting α_r to 1 indicates the property isn't particular to the entity itself, but rather always drawn from the underlying role distribution. We set α_r to 1 for pronoun heads as well as for the governor properties.

4 Learning and Inference

Since we will make use of unannotated data (see Section 5), we utilize a variational EM algorithm to learn parameters $\mathbf{R}$ and ϕ. The E-Step requires the posterior $P(\mathbf{E}, \mathbf{Z}|\mathbf{R}, \mathbf{M}, \phi)$, which is intractable to compute exactly. We approximate it using a surrogate variational distribution of the following factored form:

$$Q(\mathbf{E}, \mathbf{Z}) = \left(\prod_{i=1}^{K} q_i(E_i)\right) \left(\prod_{j=1}^{n} r_j(Z_j)\right)$$

Each $r_j(Z_j)$ is a distribution over the entity indicator for mention M_j, which approximates the true posterior of Z_j. Similarly, $q_i(E_i)$ approximates the posterior over entity E_i which is associated with role R_i. As is standard, we iteratively update each component distribution to minimize KL-divergence, fixing all other distributions:

$$q_i \leftarrow \operatorname*{argmin}_{q_i} KL(Q(\mathbf{E}, \mathbf{Z}) | P(\mathbf{E}, \mathbf{Z}|\mathbf{M}, \mathbf{R}, \phi)$$
$$\propto \exp\{\mathbb{E}_{Q/q_i} \ln P(\mathbf{E}, \mathbf{Z}|\mathbf{M}, \mathbf{R}, \phi))\}$$

[2]There is one exception: the sizes of the proper and nominal head property lists are jointly generated, but their word lists are still independently populated.

[3]While, in principle, this process can yield word lists with duplicate words, we constrain the model during inference to not allow that to occur.

[4]Sentence parse trees are merged into a right-branching document parse tree. This allows us to extend tree distance to inter-sentence nodes.

[5]The sole parameter γ is fixed at 0.1.

	Ment Acc.	Ent. Acc.
INDEP	60.0	43.7
JOINT	64.6	54.2
JOINT+PRO	**68.2**	**57.8**

Table 1: Results on corporate acquisition tasks with given role mention boundaries. We report mention role accuracy and entity role accuracy (correctly labeling all entity mentions).

For example, the update for a non-pronominal entity indicator component $r_j(\cdot)$ is given by:[6]

$$\begin{aligned}\ln r_j(z) &\propto \mathbb{E}_{Q/r_j} \ln P(\mathbf{E}, \mathbf{Z}, \mathbf{M}|\mathbf{R}, \phi) \\ &\propto \mathbb{E}_{q_z} \ln \left(P(z|\phi) P(M_j|E_z, R_z)\right) \\ &= \ln P(z|\phi) + \mathbb{E}_{q_z} \ln P(M_j|E_z, R_z)\end{aligned}$$

A similar update is performed on pronominal entity indicator distributions, which we omit here for space. The update for variational entity distribution is given by:

$$\begin{aligned}\ln q_i(e_i) &\propto \mathbb{E}_{Q/q_i} \ln P(\mathbf{E}, \mathbf{Z}, \mathbf{M}|\mathbf{R}, \phi) \\ &\propto \mathbb{E}_{\{r_j\}} \ln \left(P(e_i|R_i) \prod_{j:Z_j=i} P(M_j|e_i, R_i) \right) \\ &= \ln P(e_i|R_i) + \sum_j r_j(i) \ln P(M_j|e_i, R_i)\end{aligned}$$

It is intractable to enumerate all possible entities e_i (each consisting of several sets of words). We instead limit the support of $q_i(e_i)$ to several sampled entities. We obtain entity samples by sampling mention entity indicators according to r_j. For a given sample, we assume that E_i consists of the non-pronominal head words and modifiers of mentions such that Z_j has sampled value i.

During the E-Step, we perform 5 iterations of updating each variational factor, which results in an approximate posterior distribution. Using expectations from this approximate posterior, our M-Step is relatively straightforward. The role parameters R_i are computed from the $q_i(e_i)$ and $r_j(z)$ distributions, and the global role prior ϕ from the non-pronominal components of $r_j(z)$.

5 Experiments

We present results on the corporate acquisitions task, which consists of 600 annotated documents split into a 300/300 train/test split. We use 50 training documents as a development set. In all documents, proper and (usually) nominal mentions are annotated with roles, while pronouns are not. We preprocess each document identically to Haghighi and Klein (2010): we sentence-segment using the OpenNLP toolkit, parse sentences with the Berkeley Parser (Petrov et al., 2006), and extract mention properties from parse trees and the Stanford Dependency Extractor (de Marneffe et al., 2006).

5.1 Gold Role Boundaries

We first consider the simplified task where role mention boundaries are given. We map each labeled token span in training and test data to a parse tree node that shares the same head. In this setting, the role-filling task is a collective classification problem, since we know each mention is filling some role.

As our baseline, INDEP, we built a maximum entropy model which independently classifies each mention's role. It uses features as similar as possible to the generative model (and more), including the head word, typed dependencies of the head, various tree features, governing word, and several conjunctions of these features as well as coarser versions of lexicalized features. This system yields 60.0 mention labeling accuracy (see Table 1). The primary difficulty in classification is the disambiguation amongst the acquired, seller, and purchaser roles, which have similar internal structure, and differ primarily in their semantic contexts. Our entity-centered model, JOINT in Table 1, has no latent variables at training time in this setting, since each role maps to a unique entity. This model yields 64.6, outperforming INDEP.[7]

During development, we noted that often the most direct evidence of the role of an entity was associated with pronoun usage (see the first "it" in Figure 1). Training our model with pronominal mentions, whose roles are latent variables at training time, improves accuracy to 68.2.[8]

5.2 Full Task

We now consider the more difficult setting where role mention boundaries are not provided at test time. In this setting, we automatically extract mentions from a parse tree using a heuristic ap-

[6] For simplicity of exposition, we omit terms where M_j is an antecedent to a pronoun.

[7] We use the mode of the variational posteriors $r_j(Z_j)$ to make predictions (see Section 4).

[8] While this approach incorrectly assumes that all pronouns have antecedents amongst our given mentions, this did not appear to degrade performance.

	ROLE ID			OVERALL		
	P	R	F_1	P	R	F_1
INDEP	79.0	65.5	71.6	48.6	40.3	44.0
JOINT+PRO	**80.3**	69.2	74.3	53.4	46.4	49.7
BEST	80.1	**70.1**	**74.8**	**57.3**	**49.2**	**52.9**

Table 2: Results on corporate acquisitions data where mention boundaries are not provided. Systems must determine which mentions are template role-fillers as well as label them. ROLE ID only evaluates the binary decision of whether a mention is a template role-filler or not. OVERALL includes correctly labeling mentions. Our BEST system, see Section 5, adds extra unannotated data to our JOINT+PRO system.

proach. Our mention extraction procedure yields 95% recall over annotated role mentions and 45% precision.[9] Using extracted mentions as input, our task is to label some subset of the mentions with template roles. Since systems can label mentions as non-role bearing, only recall is critical to mention extraction. To adapt INDEP to this setting, we first use a binary classifier trained to distinguish role-bearing mentions. The baseline then classifies mentions which pass this first phase as before. We add 'junk' roles to our model to flexibly model entities that do not correspond to annotated template roles. During training, extracted mentions which are not matched in the labeled data have posteriors which are constrained to be amongst the 'junk' roles.

We first evaluate role identification (ROLE ID in Table 2), the task of identifying mentions which play some role in the template. The binary classifier for INDEP yields 71.6 F_1. Our JOINT+PRO system yields 74.3. On the task of identifying and correctly labeling role mentions, our model outperforms INDEP as well (OVERALL in Table 2). As our model is generative, it is straightforward to utilize totally unannotated data. We added 700 fully unannotated documents from the mergers and acquisitions portion of the Reuters 21857 corpus. Training JOINT+PRO on this data as well as our original training data yields the best performance (BEST in Table 2).[10]

To our knowledge, the best previously published results on this dataset are from Siefkes (2008), who report 45.9 weighted F_1. Our BEST system evaluated in their slightly stricter way yields 51.1.

[9] Following Patwardhan and Riloff (2009), we match extracted mentions to labeled spans if the head of the mention matches the labeled span.

[10] We scaled expected counts from the unlabeled data so that they did not overwhelm those from our (partially) labeled data.

6 Conclusion

We have presented a joint generative model of coreference resolution and role-filling information extraction. This model makes role decisions at the entity, rather than at the mention level. This approach naturally aggregates information across multiple mentions, incorporates unannotated data, and yields strong performance.

Acknowledgements: This project is funded in part by the Office of Naval Research under MURI Grant No. N000140911081.

References

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

Dayne Freitag and Andrew McCallum. 2000. Information extraction with hmm structures learned by stochastic optimization. In *Association for the Advancement of Artificial Intelligence (AAAI)*.

Dayne Freitag. 1998. Machine learning for information extraction in informal domains.

A. Haghighi and D. Klein. 2010. Coreference resolution in a modular, entity-centered model. In *North American Association of Computational Linguistics (NAACL)*.

P. Liang and D. Klein. 2007. Structured Bayesian nonparametric models with variational inference (tutorial). In *Association for Computational Linguistics (ACL)*.

S. Patwardhan and E. Riloff. 2007. Effective information extraction with semantic affinity patterns and relevant regions. In *Joint Conference on Empirical Methods in Natural Language Processing*.

S. Patwardhan and E Riloff. 2009. A unified model of phrasal and sentential evidence for information extraction. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 433–440, Sydney, Australia, July. Association for Computational Linguistics.

Christian Siefkes. 2008. *An Incrementally Trainable Statistical Approach to Information Extraction: Based on Token Classification and Rich Context Model*. VDM Verlag, Saarbrücken, Germany, Germany.