Joint Inference of Entities, Relations, and Coreference

Sameer Singh\textsuperscript{1} Sebastian Riedel\textsuperscript{1} Brian Martin\textsuperscript{1} Jiaping Zheng\textsuperscript{1} Andrew McCallum\textsuperscript{1} \\
\textsuperscript{1}School of Computer Science, University of Massachusetts, Amherst MA \\
\textsuperscript{2}Department of Computer Science, University College London, UK \\
\{sameer, martin, jzheng, mccallum\}@cs.umass.edu, sriedel@cs.ucl.ac.uk

ABSTRACT

Although joint inference is an effective approach to avoid cascading of errors when inferring multiple natural language tasks, its application to information extraction has been limited to modeling only two tasks at a time, leading to modest improvements. In this paper, we focus on the three crucial tasks of automated extraction pipelines: entity tagging, relation extraction, and coreference. We propose a single, joint graphical model that represents the various dependencies between the tasks, allowing flow of uncertainty across task boundaries. Since the resulting model has a high tree-width and contains a large number of variables, we present a novel extension to belief propagation that sparsifies the domains of variables during inference. Experimental results show that our joint model consistently improves results on all three tasks as we represent more dependencies. In particular, our joint model obtains 12% error reduction on tagging over the isolated models.

Categories and Subject Descriptors

I.2.7 [Artificial Intelligence]: Natural Language Processing

Keywords

Information Extraction; Joint Inference; Coreference Resolution; Named Entity Recognition; Relation Extraction

1. INTRODUCTION

Most natural language processing tasks are decomposed into a number of subtasks, for example chunking, part-of-speech tagging, named entity recognition, parsing, semantic role labeling, relation extraction and coreference. Often, independently-trained models for each of these tasks are placed in a pipeline system, with the best output prediction of each model feeding into downstream modules as observed input. Since these pipeline systems are restricted to a uni-directional flow, they suffer from cascading errors. To address this concern, there has been some past and growing recent interest in joint inference across multiple NLP tasks [14, 6, 16] that allows bi-directional information flow to correct errors made earlier in the pipeline using later predictions. Recent work on joint inference has focused on information extraction. Finkel and Manning [5] perform named-entity recognition and parsing jointly, obtaining improvement on both tasks. Roth and Yih [20] perform joint inference for entity and relation classification. Haghighi and Klein [8] propose a generative model to jointly predict entity type and coreference. These approaches demonstrate the benefits of joint inference by improving the accuracy on the individual subtasks.

In spite of the obvious benefits of joint inference, previous work has faced difficulty in obtaining improvements from joint inference, as they are limited by the range of tasks they consider. Instead of representing the whole information extraction system with a single model, only two tasks are modeled jointly and cascading errors prevail for the remaining tasks. Further, accurate coreference is often crucial for improving other components in the pipeline, and most of systems do not consider coreference jointly with other tasks. Due to these reasons, existing approaches have obtained negligible (or minor) improvements on the joint tasks. For the task of joint relation and entity labeling, Roth and Yih [20] show worse accuracy on entity labeling. Sutton and McCallum [26] present a joint model of parsing and semantic role labeling that performs worse than the pipeline approach. In the CoNLL-2008 shared task on joint parsing and semantic role labeling [25], top five systems in the closed challenge consisted of pipeline approaches. Even when positive results are demonstrated, as in Finkel and Manning [5] and Kate and Mooney [10], they are modest.

In this paper, we propose a single, joint probabilistic graphical model for classification of entity mentions (entity tagging), clustering of mentions that refer to the same entity (coreference resolution), and identification of the relations between these entities (relation extraction). Figure 1 shows an annotated example sentence. We expect our joint model to reduce cascading errors by facilitating bi-directional information flow, allowing entity tags to be improved by better relation extraction (relations such as EMPLOY can only occur between a PERSON and an ORGANIZATION) and by coreference resolution (as mentions that are coreferent have the same type). To deal with our joint model’s high tree-width and large number of variables and factors, we introduce a modification to belief propagation.
that facilitates efficient inference. In particular, during the course of inference we examine the marginals of individual variables and fix the value of a variable if the entropy of the marginal becomes low.

2. ISOLATED MODELS

Graphical Models: Graphical models define a family of probability distributions that factorize according to the dependencies encoded in the graph structure. Factor graphs are commonly used to represent undirected graphical models [11]. Formally, a factor graph $G$ is a bipartite graph with random variable $y$, and factors $\Psi = \{\Psi_f\}$. The probability distribution can be written as $p(y) = \sum_{\psi \in \Psi} \Psi(y)$, where $Z = \sum_y \prod_{\psi} \Psi(y)$ is the partition function that ensures the probabilities sum to one. Exact inference is NP-hard for models that are not trees, and the complexity depends on the tree-width or the “loopiness” of the model. Factors are often defined as a log-linear combination of feature functions $f_i$ and model parameters $\theta$, i.e. $\Psi_i(y) = \exp(\theta \cdot f_i(y))$.

The parameters $\theta$ are learned from labeled data using maximum likelihood that uses inference as a sub-routine. Graphical models are a popular tool for representing uncertainty for NLP, using models such as classification1, linear chains, etc.

Entity Tagging: Entity tagging is the task of classifying each entity mention according to the type of entity to which they refer. The input for this task is the set mention boundaries and the sentences of a document. For each mention $m_i$, the output of entity tagging is a label $t_i$ from a predefined set of labels $T$. The set of labels used in newswire consist of PERSON, ORGANIZATION, GEO-POLITICAL, LOCATION, FACILITY, VEHICLE, and WEAPON. A common approach, and the approach that we follow, is to treat entity tagging as a classification task using a maximum entropy model [1]. As shown in Figure 2a, the graphical model is defined using a factor $\Psi_T(m_i, t_i)$ for each entity tag variable $t_i$. We use several granularities for features: word-level, mention-level, and sentence-level. For this, we leverage research on a similar2 task, Named Entity Recognition (NER), using features based on Ratinov and Roth [18].

Relation Extraction: Relation extraction labels each entity mention pair in the same sentence with its relation as expressed in that sentence, or NONE if no relation is expressed. This task is often represented as variables $r_{ij}$ that represent the type of the relation where $m_i$ is the first argument, $m_j$ the second argument, and the type comes from a predefined set of labels $R$.[3] In the example in Figure 1, there are two relations: EMPLOY-STAFF between “Schumacher” and “the Italian team”, and BASEDIN between “the Italian team” and “Italian”. A common model for relation extraction is to independently label each entity mention pair with its type [9]. As shown in Figure 2b, this model is represented as factors $\Psi_R(r_{ij}, m_i, m_j, t_i, t_j)$ and $\Psi_T(m_i, t_i)$ over variables. The features we use are drawn upon Zhou et al. [34].

Coreference: Coreference is the task of linking mentions within a document that refer to the same real-word entity. Given the mentions in a document, and the coreference system predicts entities by identifying links between the mentions. In Figure 1 for example, “Schumacher,” “Michael Schumacher,” and “I” all refer to the same person, and should be all linked together. A common approach to the coreference task is to classify pairs of mentions as coreferent or not, i.e. for pairs of mentions $m_i$ and $m_j$ that appear in the same document, there is a variable $c_{ij} \in \{0, 1\}$. These decisions are symmetric ($c_{ij} \equiv c_{ji}$), and we only include one of these variables in the model. Coreference also requires the link decisions to be transitive. Transitivity could be captured using $O(n^3)$ deterministic factors, but this is often intractable. Instead, transitive closures of the coreferent pairs is computed as a post processing step. The parameters of the model are defined by a factor template $\Psi_C(c_{ij}, m_i, m_j, t_i, t_j)$, as shown in Figure 2c. The features are based on Soon et al. [23] and Bengston and Roth [2].

3. JOINT MODEL

In the previous section we describe the different models for the three tasks. The relation extraction and coreference models both use fixed entity tags to predict the type of the relation or coreference. Unfortunately, with such models it is not possible for entity tagging to benefit from the coreference or the relation extraction decisions, resulting in a uni-directional flow of information out of the entity tagging model. Futher, this causes a cascade of error; incorrect entity tags result in adverse effect on relations and coreference. However, evidence at the relation or coreference levels can improve the entity tagging task. Certain relations, for example, only appear between entities of a specific type. Similarly, for two mentions that are coreferent, by definition their entity types has to be the same. Isolated models do not have any direct mechanism to facilitate this bi-directional flow of information.

We need to define a model that directly represents the dependencies between the three tasks by modeling the joint distribution over the three tasks. Since the individual models defined in the previous section represent the individual tasks, we construct a joint model by combining all the variables and factors into a single graphical model, but do not fix the entity tags to be observed. See Figure 3 for an illustration of the joint model as defined over 3 mentions. Note that even with such a small set of mentions, the underlying joint model is quite complex and dense. Formally, the probability for a setting to all the document variables is:

$$p(t, r, c | m) \propto \prod_{i \in T} \Psi_T(m_i, t_i) \prod_{c_{ij} \in \mathcal{C}} \Psi_C(c_{ij}, m_i, m_j, t_i, t_j) \prod_{r_{ij} \in R} \Psi_R(r_{ij}, m_i, m_j, t_i, t_j)$$

The factors here denote different distributions than in Section 2. Instead of representing a distribution over the labels of a single task conditioned on the predictions from another task, these factors now directly represent the joint (unnormalized) distribution over the tasks that they are defined over. For example, in Section 2, each coreference factor defines a distribution over one pairwise boolean coreference variable conditioned on the entity tags of the mentions. In the joint model, however, this factor induces a distribution over
both the pairwise boolean variable and the entity tags of the two mentions, based on the observed features of the two mentions. When trained, this factor can capture the bi-directional information flow between the tasks and, for example, encourage the entity tags of two mentions to be the same if confident about them being coreferent. Similarly, the relation extraction factors also induce a distribution over the entity tags of their arguments.

The coreference resolution and relation extraction are not directly connected in the model, as the dependency between these two tasks is much weaker in practice. Nonetheless they are not independent. As part of the same graphical model, information can flow between the two via entity tags, resulting in indirect improvements to relation extraction when coreference improves, and vice versa.

4. LEARNING AND INFERENCE

Given the large size, strong dependencies, and complex structure of the joint model, we cannot directly apply existing approaches to inference and learning. Instead we propose modifications to pseudo-likelihood learning for efficient parameter estimation, and a novel extension to the belief propagation algorithm.

4.1 Piecewise Learning for Joint Models

Learning is used to identify the set of parameters that maximize the likelihood of the labeled data, which, for our joint model, will be the joint likelihood over all the three tasks. Common approaches to maximize the training objective, such as BFGS, unfortunately cannot be applied to our setting for several reasons. First, many of these techniques assume inference in their inner loop, which is NP-Hard for exact inference (and computationally expensive even when using approximations). Learning with approximate inference for such models can often diverge [12]. Second, the likelihood is defined over all the tasks simultaneously, and the optimization approaches face difficulty balancing between the different tasks, often biasing the learning for the task with most terms in the objective. Third, the number of parameters for the joint factors is the cross product of the domains, resulting in billions of parameters.

Our approach to learning attempts to address these concerns. Since joint training is intractable due to the complexity of inference, we use the piecewise training [27] approach to learn our models. This approach decomposes the model into pieces, and maximizes the piecewise likelihood by treating each piece independently. For our joint model, we treat each factor as a piece, separately learning the distribution over the neighboring unobserved variables given its neighbors and features. Further, to facilitate faster convergence, the predictions from entity tagging factors are incorporated during piecewise training of relation extraction and coreference as fixed incoming beliefs. To limit the number of parameters that occur in the joint model between entity tagging and relation extraction, we only include the features that appear at least once in the training data. These modifications to existing approaches enable tractable parameter estimation of the joint model.

Learning a joint model using piecewise training is different from combining independent models using manually-specified constraints, as in Roth and Yih [20]. Our factors represent the complete joint distributions over the multiple tasks they touch. From the constraints perspective, we are learning soft constraints between tasks, instead of manually enumerating constraints and hand-tuning the weights; we can also incorporate such constraints if available.

4.2 Sparsity for Efficient Inference

Due to the number of variables, non-trivial domain sizes, strong dependencies, and a loopy structure, common approximate inference techniques, such as belief propagation and sampling, cannot be applied directly. Belief propagation converges to accurate marginals in a few iterations when the model is mostly cycle-free, and is fast when marginalization of each factor is quick. Unfortunately, the joint model is incredibly loopy due to the large number of factors that connect variables across the whole document. Further, even marginalization of individual factors is non-trivial due to the large domain involved (for example, the neighborhood of $\Psi^r$, consists of all combinations of the relation label along with the entity tags for the argument entities). These reasons prevent the direct use of belief propagation (BP) in our model.

A few alternatives to belief propagation exist in the literature. MCMC-based sampling often scales to models such as ours, however faces local minima issues, and often requires designing customized proposal functions. Yet another option for inference would be to frame the problem as an integer linear program (ILP), as in Roth and Yih [20]. BP style inference is preferred over ILP for many reasons (a) BP provides marginals while ILP does not, (b) our joint factors assign different scores to each value in the joint domains, which results in a cubic number of binary variables in the ILP formulation. Further, (c) BP is often more efficient than linear programming solvers, let alone ILP [30].

Since belief propagation is not directly applicable, we adapt the algorithm for inference on our model. Our main extension stems from the insight that during inference in NLP models, most of the variable marginals often peak during the initial stages of inference, without changing substantially during the rest of the course of inference. Detecting these low-entropy marginals in earlier phases and freezing to their high-probability values provides benefits to belief propagation. First, since the domain now contains only a single value, the factors that neighbor the variable can marginalize much more efficiently. Second, these fixed variables result in fewer cycles in the model and allow decomposition of the model into independent inference problems by partitioning at these fixed variables. Lastly, factors that only neighbor fixed variables can be effectively removed during inference, reducing the amount of messages that are passed.

To employ these benefits of value sparsity in belief propagation, we examine the marginals of all the variables after every iteration of message passing. When the probability of a value for a variable goes above a predetermined probability threshold $\zeta$, we set the value of the variable to its maximum probability value, treating it as a fixed variable for the rest of inference. The parameter $\zeta$ directly controls the computational efficiency and accuracy trade-off, and we set the value for this parameter based on observing inference on the held out training data\footnote{This algorithm and tradeoff is studied in Singh et al. [22].}. We incorporate transitivity into the inference technique by directly propagating the positive coreference decisions over their transitive closure after every iteration of message passing.
Tables 2, 3 and 4. Allowing uncertainty in entity tags improves the accuracies of both the tasks, demonstrating the importance of propagating uncertainty along the pipeline. Further, there are significant error reductions for entity tagging, corroborating the need for flow of information from relations and coreference to the tagging model. When performing inference together on all the three tasks, we achieved further improvements for all of them, most significantly an error reduction of 12.4% for the entity tagging task.

## 5. EXPERIMENTS

We use the Automatic Content Extraction (ACE) 2004 English dataset for the experiments, a standard labeled corpus for the three tasks that we are studying [4]. ACE consists of 443 documents from 4 distinct news domains, with 7, 790 sentences and 172, 506 tokens. Counts of each type of variable are shown in Table 1. For these experiments, we use gold mention boundaries, and the coarse-grained labels for tagging and relations (7 and 8 respectively). We run 4 iterations of inference with $\zeta = 0.8$.

### Isolated Models: We train the isolated models using the features described in Section 2. Our model for entity tagging achieves an accuracy of 80.2%, which is impressive considering many of the mentions are pronouns and pronominals with little evidence in the context. Our relation extraction model achieves an F1 score of 54.05% which is comparable to existing research that uses only predicted entity tags (we obtain 61% F1 with gold tags). The coreference model achieves a macro $F^3$ F1 score of 76.34%, which is competitive with related approaches.

### Joint Inference Results: We first present joint inference between pairs of tasks. In particular, we separately evaluate the result of joint inference between entity tagging and each of the other two tasks. The results, when compared to the isolated models, are shown in Tables 2, 3 and 4. Allowing uncertainty in entity tags improves the accuracies of both the tasks, demonstrating the importance of propagating uncertainty along the pipeline. Further, there are significant error reductions for entity tagging, corroborating the need for flow of information from relations and coreference to the tagging model. When performing inference together on all the three tasks, we achieved further improvements for all of them, most significantly an error reduction of 12.4% for the entity tagging task.

## 6. RELATED WORK

### Individual Tasks: There has been considerable research on the individual tasks covered in this paper.

Relation extraction systems generally fall into two categories. Feature-based systems [34, 24] employ a variety of features, including lexical, syntactical and semantic ones. The other common approach is to use convolution tree kernel for similarity [33]. Zhou et al. [35] proposed a composite of the tree kernel and a linear kernel that outperformed the individual kernels. Jiang and Zhai [9] system-

### Coreference Resolution: MUC metric has been provided for comparison to exiting work; it is much less informative compared to Pairwise and B$^3$ since simple baselines attain high scores.

Majority of the coreference resolution systems use binary classification [2]. Haghighi and Klein [7] designed a deterministic system based on a rich set of syntactic and semantic compatibilities. Unlike our model that computes the transitive closure from all the pairwise predictions, Soon et al. [23] used the most recent positive antecedent, and Ng and Cardie [15] linked to the best antecedent among all candidates for each mention. Culotta et al. [3] argued that the mention-pair method cannot capture features of sets of noun phrases, and hence use entity-based features.

### Joint Inference: In recent years, there has been an increasing interest in approaches to joint representations of multiple information extraction and natural language processing tasks [13]. Most relevant to our work is the combination of entity labeling and relation extraction. Roth and Yih [20] use the ILP framework to enforce manually-specified constraints between the tasks. Yao et al. [31] accomplish this through distant supervision via Wikipedia. Similar to our work, Yu and Lam [32] also model entities and relations using a discriminative model. Others have combined parsing with NER [5] and semantic role labeling [26] with mixed success. Finkel et al. [6] represents a pipeline of NLP tasks as a Bayesian network where each variable represents one stage of the pipeline. Joint inference has also been applied to various information extraction tasks such as citation segmentation and matching [29, 16, 21] and to BioNLP [17, 19].

Our approach to joint inference differs significantly from these. First, we model three crucial information extraction tasks, including coreference, which have not been modeled together before. Coreference as the third task requires document-level joint inference, as opposed to sentence-level joint inference in related work. Second, our resulting model is significantly more loopy than a number of existing joint inference techniques. Third, as opposed to some of the related work, we learn both hard and soft constraints between tasks instead of setting them by hand (as in Roth and Yih [20]), and our inference provides marginals. Due to the dependencies represented in our model, and our inference technique, we are able to obtain consistent improvements in all the three tasks, improving accuracy as we include more dependencies.

## 7. CONCLUSIONS

This paper introduces a novel, fully-joint model of three crucial information extraction tasks, entity tagging, relation extraction, and coreference. The model contains factors that represent the different dependencies that lie between the tasks, resulting in a high tree-width structure containing all the variables of a document. To facilitate efficient inference, we introduce a novel extension to belief propagation that sparsifies variables during inference, effectively eliminating the need to compute a majority of the messages. The combination of a joint model, and an accompanying inference technique, results in improvements to all three tasks. These results add substantially to our understanding of the joint inference, providing additional support that the improved representation of multiple tasks in the same model is beneficial to all the tasks.
Acknowledgments

We would like to thank the anonymous reviewers at http://openreview.net for their helpful comments. This work was supported in part by the Center for Intelligent Information Retrieval and in part by JARPA via DoI/NBC contract #D11PC20152. The U.S. Government is authorized to reproduce and distribute reprint for Governmental purposes notwithstanding any copyright annotation thereon. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

References


