Joint Inference of Entities, Relations, and Coreference

Sameer Singh[§] Sebastian Riedel[†] Brian Martin[§] Jiaping Zheng[§] Andrew McCallum[§]
[§]School of Computer Science, University of Massachusetts, Amherst MA
[†]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.

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 [15, 6, 18] 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 [22] 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 information extraction 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 hence 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, these approaches obtain negligible (or minor) improvements on the joint tasks. For the task of joint relation and entity labeling, Roth and Yih [22] show negative improvements on entity labeling. Sutton and McCallum [27] 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 [26], the 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 [11], they are modest.

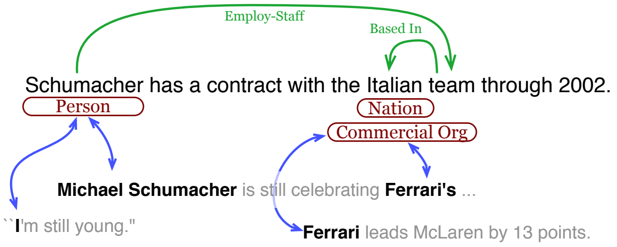

Employ-Staff
Based In
Schumacher has a contract with the Italian team through 2002.
Person
Nation
Commercial Org
Michael Schumacher is still celebrating Ferrari's ...
``I'm still young."
Ferrari leads McLaren by 13 points.

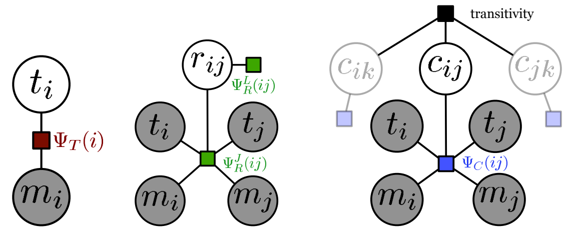
Figure 1: **Information Extraction:** 3 mentions labeled with entity types (red), relations (green), and coreference (blue links).

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 (since certain relations, such as EMPLOY can only occur between a PERSON and an ORGANIZATION) and by coreference resolution (as mentions which are coreferent have an identical entity 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 variables to a single, fixed value 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

AKBC CIKM 2013 Workshop on Automated Knowledge Base Construction
Copyright 2013 ACM X-XXXXX-XX-X/XX/XX ...$15.00.



(a) Tagging (b) Relations (c) Coreference

Figure 2: **Individual Classification Models:** where the observed/fixed variables are shaded in grey.

encoded in the graph structure. Factor graphs are commonly used to represent undirected graphical models [12]. Formally, a factor graph $\mathcal{G}$ is a bipartite graph with random variable $\mathbf{y}$, and factors $\Psi = \{\Psi_i\}$. The probability distribution can be written as $p(\mathbf{y}) = \frac{1}{Z}\prod_{\Psi_i\in\Psi}\Psi_i(\mathbf{y})$, where $Z = \sum_{\mathbf{y}}\prod_{\Psi_i}\Psi_i(\mathbf{y})$ 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 θ, i.e. $\Psi_i(\mathbf{y}) = \exp\left(\theta\cdot\vec{f_i}(\mathbf{y})\right)$. The parameters θ are learned from labeled data using maximum likelihood that uses inference as a sub-routine. Graphical models are a popular tool for representing uncertainty NLP tasks, using models such as classification[1], 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 $\mathcal{T}$. The set of labels used in newswire consist of PERSON, ORGANIZATION, GEO-POLITICAL, LOCATION, FACILITY, VEHICLE, and WEAPON. As an example of entity tagging, consider the sentence "*A cap for a depressurization valve floated away moments after* $\underbrace{\textit{Bill McArthur}}$ *emerged from* $\underbrace{\textit{space shuttle Discovery,}}$" where the correct labels are PERSON and VEHICLE, respectively. A common approach, and the approach that we follow, is to treat entity recognition as a classification task using a maximum entropy model [1], as shown in Figure 2a. This model may be written as a graphical model by defining 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 have leveraged research on a similar[2] task, Named Entity Recognition (NER), using features based on Ratinov and Roth [20].

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 $\mathcal{R}$.[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 [10], as shown in Figure 2b. This model is represented as factor templates $\Psi_R^L(m_i, m_j, r_{ij})$ and $\Psi_R^J(r_{ij}, m_i, m_j, t_i, t_j)$ over variables.[4] The features we use are drawn upon Zhou et al. [35].

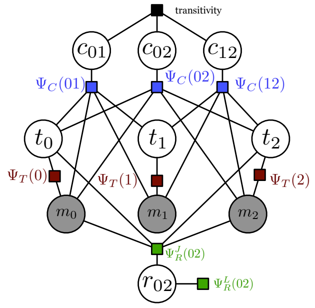


Figure 3: **Joint Model of Entity Tagging, Resolution and Relations:** over 3 observed mentions, two of which belong to the same sentence (m_0 and m_2). For brevity, we have only included (i, j) in the factor labels, and have omitted t_i, t_j, m_i and m_j.

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. As in Bengston and Roth [2], we include decisions only for mention pairs where the first mention is not pronominal. 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. [24] 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 of to predict the type of the relation or coreference between their arguments. 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. This also causes a cascade of errors, 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; relation type EMPLOY will likely have a PERSON and an ORGANIZATION as its arguments. 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. To elaborate, the resulting model has the same variables and features as the individual models. The factors of the joint models are also same as the set of factors instantiated by individual models, as described in section 2. See Figure 3 for an illustration of the joint model as defined over

[1] factors touch single *random* variables: $\Psi_i(\mathbf{y}) = \Psi_i(y_i)$

[2] In NER as defined by Tjong Kim Sang and De Meulder [29], mentions spans and labels need to be predicted.

[3] Note that r_{ij} and r_{ji} represent different relations.

[4] L signifies local template, while J signifies joint.

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 variables in a document is:

$$
\begin{aligned}
&p(\mathbf{t},\mathbf{r},\mathbf{c}|\mathbf{m}) \propto \\
&\quad \prod_{t_i\in\mathbf{t}} \Psi_T(m_i,t_i) \prod_{c_{ij}\in\mathbf{c}} \Psi_C(c_{ij},m_i,m_j,t_i,t_j) \\
&\quad \prod_{r_{ij}\in\mathbf{r}} \Psi_R^L(m_i,m_j,r_{ij})\Psi_R^J(r_{ij},m_i,m_j,t_i,t_j)
\end{aligned}
$$

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 [13]. 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, which, in our case, is coreference resolution. Third, the number of parameters for the joint factors is the full product of multiple domains, resulting in billions of parameters if not explicitly restricted.

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* [28] approach to learn our models. Instead of learning all the factors jointly, 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 an independent piece, separately learning the distribution over the neighboring unobserved variables given its neighboring observed variables 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 *supported* features, i.e. 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 [22]. 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 Ψ_R^J 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 [22]. 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. The ILP formulation of our model will result in a cubic number of binary variables. Further, (c) BP is often more efficient than linear programming solvers, let alone ILP [31].

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 *fixing* 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 ζ, 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 ζ directly controls the computational efficiency and accuracy trade-off, and we set the value for this parameter based on observing inference on the

Data	#Mentions	#Coreference	#Relation
Train	15,640	637,160	82,479
Dev	5,545	244,461	34,057
Test	6,598	342,942	38,270

Table 1: Number of variables in the various folds.

Model	Accuracy	Error Red.
Isolated Model	80.23	-
Joint w/ Coreference	81.24	5.1
Joint w/ Relations	81.77	7.8
Complete Joint	**82.69**	12.4

Table 2: **Entity Tagging:** Results for various models.

held out training data. We incorporate transitivity into the inference technique by directly propagating the sparse coreference decisions over their transitive closure after every iteration of message passing. We only propagate positive sparse coreferent decisions, while being consistent with negative sparse decisions.

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 B^3 F1 score of 76.34%, which is competitive with related approaches with predicted entity tags (for e.g. Haghighi and Klein [7] obtain 76.9).

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 the model defined over all the three tasks, we achieved further improvements for all three tasks, most significantly we achieved an error reduction of 12.4% for the entity tagging task.

Model	Prec	Rec	F1
Pipeline (w/ Tagging)	53.22	54.92	54.05
Joint w/ Tagging	54.93	54.02	54.47
Complete Joint	56.06	54.74	**55.39**

Table 3: **Relation Extraction:** Comparison using the F-measure.

Model	MUC	Pairwise	B^3
Pipeline (w/ Tagging)	**73.81**	53.94	76.34
Joint w/ Tagging	71.09	57.59	78.06
Complete Joint	73.00	**58.39**	**78.50**

Table 4: **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.

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 [35, 25] employ a variety of features, including lexical, syntactical and semantic ones. The other common approach is to use convolution tree kernel for similarity [34]. Zhou et al. [36] proposed a composite of the tree kernel and a linear kernel that outperformed the individual kernels. Jiang and Zhai [10] systematically explored the feature space and showed that using more than the basic features only yields small improvements. Nguyen and Moschitti [17] built a composite kernel similar to Zhou et al. [36] and found that distant supervision using Wikipedia data improved performance. Hoffmann et al. [9] considered cases when multiple relations hold on the same entity mentions.

A majority of systems cast the coreference resolution task as binary classifications [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. [24] used the most recent positive antecedent, and Ng and Cardie [16] 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 [14]. Most relevant to our work is the combination of entity labeling and relation extraction. Roth and Yih [22] use the ILP framework to enforce manually-specified constraints between the tasks. Yao et al. [32] accomplish this through distant supervision via Wikipedia. Similar to our work, Yu and Lam [33] also model entities and relations using a discriminative model. Others have combined parsing with NER [5] and semantic role labeling [27] 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 [30, 18, 23] and to BioNLP [19, 21].

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. Difficulty of inference is further compounded from transitivity. 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 [22]), 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.

References

[1] O. Bender, F. Och, and H. Ney. Maximum entropy models for named entity recognition. In *North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL HLT)*, pages 148–151. Association for Computational Linguistics, 2003.

[2] E. Bengston and D. Roth. Understanding the value of features for coreference resolution. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2008.

[3] A. Culotta, M. Wick, and A. McCallum. First-order probabilistic models for coreference resolution. In *North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL HLT)*, 2007.

[4] G. Doddington, A. Mitchell, M. Przybocki, L. Ramshaw, S. Strassel, and R. Weischedel. The Automatic Content Extraction (ACE) program–tasks, data, and evaluation. In *Proceedings of LREC*, volume 4, pages 837–840. Citeseer, 2004.

[5] J. R. Finkel and C. D. Manning. Joint parsing and named entity recognition. In *North American Chapter of the Association for Computational Linguistics (NAACL HLT)*, 2009.

[6] J. R. Finkel, C. D. Manning, and A. Y. Ng. Solving the problem of cascading errors: Approximate bayesian inference for linguistic annotation pipelines. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2006.

[7] A. Haghighi and D. Klein. Simple coreference resolution with rich syntactic and semantic features. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1152–1161, 2009.

[8] A. Haghighi and D. Klein. Coreference resolution in a modular, entity-centered model. In *North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL HLT)*, pages 385–393, 2010.

[9] R. Hoffmann, C. Zhang, X. Ling, L. Zettlemoyer, and D. S. Weld. Knowledge-based weak supervision for information extraction of overlapping relations. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 541–550, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[10] J. Jiang and C. Zhai. A systematic exploration of the feature space for relation extraction. In *North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL HLT)*, pages 113–120, Rochester, New York, April 2007. Association for Computational Linguistics.

[11] R. J. Kate and R. J. Mooney. Joint entity and relation extraction using card-pyramid parsing. In *Conference on Computational Natural Language Learning (CoNLL)*, 2010.

[12] F. R. Kschischang, B. J. Frey, and H. A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions of Information Theory*, 47(2):498–519, Feb 2001.

[13] A. Kulesza and F. Pereira. Structured learning with approximate inference. In *Neural Information Processing Systems (NIPS)*, 2008.

[14] A. McCallum and D. Jensen. A note on the unification of information extraction and data mining using conditional-probability, relational models. In *IJCAI Workshop on Learning Statistical Models from Relational Data*, 2003.

[15] A. McCallum, K. Nigam, J. Rennie, and K. Seymore. A machine learning approach to building domain-specific search engines. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 1999.

[16] V. Ng and C. Cardie. Improving machine learning approaches to coreference resolution. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 104–111, 2002.

[17] T.-V. T. Nguyen and A. Moschitti. Joint distant and direct supervision for relation extraction. In *International Joint Conference on Natural Language Processing (IJCNLP)*, pages 732–740, Chiang Mai, Thailand, November 2011. Asian Federation of Natural Language Processing. URL http://www.aclweb.org/anthology/I11-1082.

[18] H. Poon and P. Domingos. Joint inference in information extraction. In *AAAI Conference on Artificial Intelligence*, pages 913–918, 2007.

[19] H. Poon and L. Vanderwende. Joint Inference for Knowledge Extraction from Biomedical Literature. In *North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL HLT)*, pages 813–821, Los Angeles, California, June 2010. Association for Computational Linguistics. URL http://www.aclweb.org/anthology-new/N/N10/N10-1123.bib.

[20] L. Ratinov and D. Roth. Design challenges and misconceptions in named entity recognition. In *Conference on Computational Natural Language Learning (CoNLL)*, pages 147–155. Association for Computational Linguistics, 2009.

[21] S. Riedel and A. McCallum. Fast and robust joint models for biomedical event extraction. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2011.

[22] D. Roth and W. Yih. Global inference for entity and relation identification via a linear programming formulation. In L. Getoor and B. Taskar, editors, *Introduction to Statistical Relational Learning*. MIT Press, 2007.

[23] S. Singh, K. Schultz, and A. McCallum. Bi-directional joint inference for entity resolution and segmentation using imperatively-defined factor graphs. In *Machine Learning and Knowledge Discovery in Databases (Lecture Notes in Computer Science) and European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, pages 414–429, 2009.

[24] W. M. Soon, H. T. Ng, and D. C. Y. Lim. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544, Dec 2001.

[25] A. Sun, R. Grishman, and S. Sekine. Semi-supervised relation extraction with large-scale word clustering. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 521–529, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[26] M. Surdeanu, R. Johansson, A. Meyers, L. Màrquez, and J. Nivre. The CoNLL-2008 shared task on joint parsing of syntactic and semantic dependencies. In *Conference on Computational Natural Language Learning (CoNLL)*, 2008.

[27] C. Sutton and A. McCallum. Joint parsing and semantic role labeling. In *Conference on Computational Natural Language Learning (CoNLL)*, 2005.

[28] C. Sutton and A. McCallum. Piecewise training for structured prediction. *Machine Learning*, 77(2–3):165–194, 2009.

[29] E. F. Tjong Kim Sang and F. De Meulder. Introduction to the CoNLL-2003 shared task: language-independent named entity recognition. In *North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL HLT)*, pages 142–147. Association for Computational Linguistics, 2003.

[30] B. Wellner, A. McCallum, F. Peng, and M. Hay. An integrated, conditional model of information extraction and coreference with application to citation matching. In *Uncertainty in Artificial Intelligence (UAI)*, pages 593–601, 2004.

[31] C. Yanover, T. Meltzer, and Y. Weiss. Linear programming relaxations and belief propagation – an empirical study. *Journal of Machine Learning Research (JMLR)*, 7:1887–1907, Dec. 2006. ISSN 1532-4435.

[32] L. Yao, S. Riedel, and A. McCallum. Collective cross-document relation extraction without labelled data. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2010.

[33] X. Yu and W. Lam. Jointly identifying entities and extracting relations in encyclopedia text via a graphical model approach. In *International Conference on Computational Linguistics (COLING)*, pages 1399–1407, Beijing, China, August 2010. Coling 2010 Organizing Committee.

[34] M. Zhang, J. Zhang, and J. Su. Exploring syntactic features for relation extraction using a convolution tree kernel. In *North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL HLT)*, HLT-NAACL '06, pages 288–295, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics. doi: 10.3115/1220835.1220872. URL `http://dx.doi.org/10.3115/1220835.1220872`.

[35] G. Zhou, J. Su, J. Zhang, and M. Zhang. Exploring various knowledge in relation extraction. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 427–434, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

[36] G. Zhou, M. Zhang, D. Ji, and Q. Zhu. Tree kernel-based relation extraction with context-sensitive structured parse tree information. In *Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 728–736, 2007.