Bi-directional Joint Inference for Entity Resolution and Segmentation Using Imperatively-Defined Factor Graphs

Sameer Singh Karl Schultz Andrew McCallum

Department of Computer Science
University of Massachusetts, Amherst

European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)
September 7-11, 2009

Outline

Outline

Citation Data

- drucker h., schapire r., and simard r. improving performance in neural networks using a boosting algorithm. advances in neural information processing systems 5, san mateo, ca. morgan kaufmann.1993 pages 42-49, in hanson, s. j., cowan, j. d., and giles, c. l., editors,
- yoav freund, and robert e. schapire. experiments with a new boosting algorithm. in proceedings of the 13th international conference on machine learning. morgan kaufmann, 1996
- freund y., schapire r.e. experiments with a new boosting algorithm, in saitta l.(ed.), proc of the thirteenth international conference on machine learning, san francisco, ca, pp.148-156, morgan kauf-mann, 1996
- drucker, schapire, and simard improving performance in neural networks using a boosting algorithm, advances in neural information processing systems 5, 1993, 42-49. 14

Segmentation

- drucker h., schapire r., and simard r. improving performance in neural networks using a boosting algorithm . advances in neural information processing systems 5 , san mateo, ca. morgan kaufmann.1993 pages 42-49, in hanson, s. j., cowan, j. d., and giles, c. l., editors.
- yoav freund, and robert e. schapire. experiments with a new boosting algorithm . in proceedings of the 13th international conference on machine learning . morgan kaufmann, 1996
- freund y., schapire r.e. experiments with a new boosting algorithm , in saitta l.(ed.), proc of the thirteenth international conference on machine learning , san francisco, ca, pp.148-156, morgan kauf-mann, 1996
- drucker, schapire, and simard improving performance in neural networks using a boosting algorithm , advances in neural information processing systems 5 , 1993, 42-49. 14

Entity Resolution

- drucker h., schapire r., and simard r. improving performance in neural networks using a boosting algorithm. advances in neural information processing systems 5, san mateo, ca. morgan kaufmann.1993 pages 42-49, in hanson, s. j., cowan, j. d., and giles, c. l., editors,
- yoav freund, and robert e. schapire. experiments with a new boosting algorithm. in proceedings of the 13th international conference on machine learning. morgan kaufmann, 1996
- freund y., schapire r.e. experiments with a new boosting algorithm, in saitta l.(ed.), proc of the thirteenth international conference on machine learning, san francisco, ca, pp.148-156, morgan kauf-mann, 1996
- drucker, schapire, and simard improving performance in neural networks using a boosting algorithm, advances in neural information processing systems 5, 1993, 42-49. 14

Entity Resolution

- drucker h., schapire r., and simard r. improving performance in neural networks using a boosting algorithm. advances in neural information processing systems 5, san mateo, ca. morgan kaufmann.1993 pages 42-49, in hanson, s. j., cowan, j. d., and giles, c. l., editors.
- yoav freund, and robert e. schapire. experiments with a new boosting algorithm. in proceedings of the 13th international conference on machine learning. morgan kaufmann, 1996
- freund y., schapire r.e. experiments with a new boosting algorithm, in saitta l.(ed.), proc of the thirteenth international conference on machine learning, san francisco, ca, pp.148-156, morgan kauf-mann, 1996
- drucker, schapire, and simard improving performance in neural networks using a boosting algorithm, advances in neural information processing systems 5, 1993, 42-49. 14

Pipeline Approach

- Each of the tasks can be solved independently

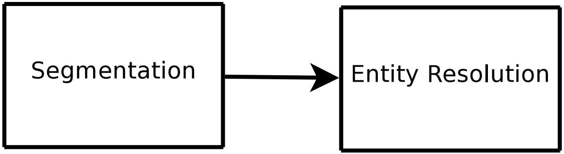
Pipeline Approach

- Each of the tasks can be solved independently
- However, sharing information can help
  - drucker h., schapire r., and simard r. improving performance in neural networks using a boosting algorithm. advances in neural information processing systems 5, san mateo, ca. morgan kaufmann.1993 pages 42-49, in hanson, s. j., cowan, j. d., and giles, c. l., editors,
  - drucker harris, schapire, robert, and simard patrice 1993. improving performance in neural networks using a boosting algorithm. in advances in neural informations processing systems 5, san ma-teo, ca. morgan kaufmann. 1993 42-49.

Pipeline Approach

- Each of the tasks can be solved independently
- However, sharing information can help
  - drucker h., schapire r., and simard r. improving performance in neural networks using a boosting algorithm. advances in neural information processing systems 5, san mateo, ca. morgan kaufmann.1993 pages 42-49, in hanson, s. j., cowan, j. d., and giles, c. l., editors,
  - drucker harris, schapire, robert, and simard patrice 1993. improving performance in neural networks using a boosting algorithm. in advances in neural informations processing systems 5, san ma-teo, ca. morgan kaufmann. 1993 42-49.
- Thus, entity resolution should use segmentation

Pipeline Approach

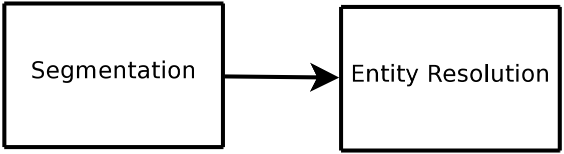


- Cascading error through the stages
  - Can be reduced by using N-best lists*, or sampling†

*Sutton & McCallum CoNLL 2005

†Finkel *et.al.* EMNLP 2006

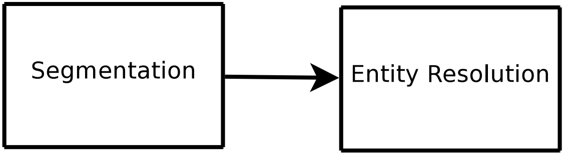
Pipeline Approach



- Cascading error through the stages
  - Can be reduced by using N-best lists*, or sampling†
- Unidirectional information flow
  - drucker, schapire, and simard improving performance in neural networks using a boosting algorithm, advances in neural information processing systems 5, 1993, 42-49. 14

*Sutton & McCallum CoNLL 2005
†Finkel *et.al.* EMNLP 2006

Pipeline Approach

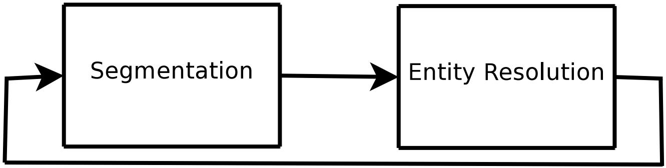


- Cascading error through the stages
  - Can be reduced by using N-best lists*, or sampling†
- Unidirectional information flow
  - drucker, schapire, and simard improving performance in neural networks using a boosting algorithm, advances in neural information processing systems 5, 1993, 42-49. 14
  - drucker harris, schapire, robert, and simard patrice 1993. improving performance in neural networks using a boosting algorithm. in advances in neural informations processing systems 5, san ma-teo, ca. morgan kaufmann. 1993 42-49.

*Sutton & McCallum CoNLL 2005

†Finkel *et.al.* EMNLP 2006

Iterated Pipeline Approach§

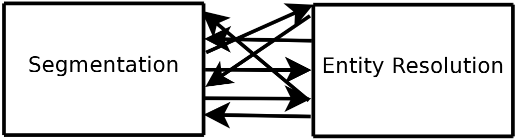


- Close the loop of the pipeline
  - Both tasks use information from each other
- Reduces cascading error
  - However, still not eliminated
  - N-best lists can be used to further reduce this error‡

‡Wellner *et.al.* UAI 2004

§Hollingshead & Roark ACL 2007

Our Approach To Joint Inference



Integrate models in a single, unified,“fully-joint” factor graph

- To **decrease cascading error** inference is performed simultaneously over both tasks
- **Increased complexity** is handled efficiently by using procedural hooks in model specification and inference

Outline

Factor graphs

- Undirected bipartite graph over variables $(\mathbf{x}, \mathbf{y})$ and factors (Ψ)
- A Factor computes a scalar value that represents the compatibility between neighboring variable values
- Parameters are tied using *factor templates*.
  $\mathbf{T_j}$: parameters $\{\theta_{jk}\}$, feature functions $\{f_{jk}\}$, set of tuples $\{(\mathbf{x}_j, \mathbf{y}_j)\}$
- Factors instantiated for each of these variable tuples share $\{\theta_{jk}\}$ and $\{f_{jk}\}$

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{T_j \in \mathcal{T}} \prod_{(\mathbf{x}_i, \mathbf{y}_i) \in T_j} \exp\left[\sum_{k=1}^{K_j} \theta_{jk} f_{jk}(\mathbf{x}_i, \mathbf{y}_i)\right]$$

Imperatively Defined Factor Graphs (IDFs)

- IDFs provide a single framework for combining declarative and procedural domain knowledge
  - By leveraging imperative constructs (snippets of procedural code)
- A model written as an IDF *is* a factor graph with all the traditional semantics of factor graphs
  - E.g., factors, variables, possible worlds, scores, partition functions

MCMC Inference and Learning

- **Metropolis-Hastings inference on factor graphs**
  - A configuration of the variables is a sample for MCMC
  - To generate the next sample, the proposal function changes values of some variables
  - Acceptance probability uses the scores given by the parameters
- **Learning using SampleRank**[¶]
  - Parameters are updated when model *disagrees* with labeled truth
  - Shown to be efficient and achieve high-accuracy

[¶] Rohanimanesh *et al.* Tech Report 2009

Imperative Hooks in MCMC

- **Metropolis-Hastings inference on factor graphs**
  - A configuration of the variables is a sample for MCMC
  - To generate the next sample, the proposal function changes values of some variables
    - Customize proposal function to generate an initial set of changes
    - Expand the set of changes to other related variables
  - Acceptance probability uses the scores given by the parameters
    - Identify factors that neighbor these changed variables
    - Calculate the features for these factors
- **Learning using SampleRank**¶
  - Parameters are updated when model *disagrees* with labeled truth
  - Shown to be efficient and achieve high-accuracy

¶Rohanimanesh *et al.* Tech Report 2009

Implementing an IDF

- Specifying an IDF
  1. Identify a natural representation of the data (variables)
  2. Create factor templates to capture dependencies between variables
  3. Create features for each template
- Comparison to *Markov logic networks (MLNs)*
  - Both are Conditional Random Fields (CRFs)
  - IDFs use a *Turing-complete* language to specify graph structure, while MLNs use first-order logic
- Implemented in the FACTORIE toolkit
  - Available at `http://factorie.cs.umass.edu/`
- For more details,
  - Talk to us at the poster session
  - See upcoming publication at NIPS 2009

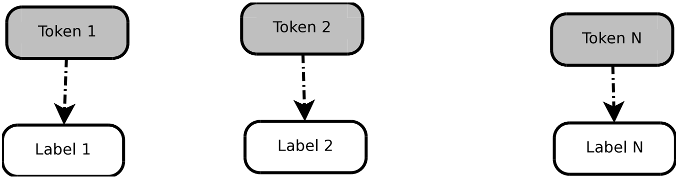
Outline

Bi-directional Joint Inference for Segmentation and Entity Resolution

- Objective:
  - **Input:** a set of mention strings (e.g., bibliographic citations)
  - **Output:**
    - A set of fields for each mention string (*segmentation*)
    - A clustering of the mention strings (*entity resolution*)
- Separate factor graphs are created for each task
- A unified factor graph is created to model both tasks
  - Contains variables for both tasks
  - Contains *joint factors*
    - Neighbor variables of different tasks
    - Capture dependencies between the tasks

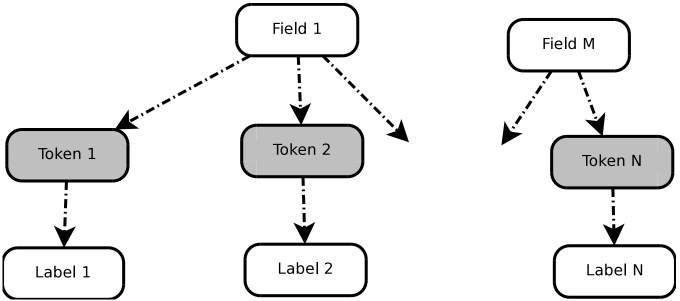
Segmentation

- **Variables**
  - `Token`: Observed variable representing a word in the mention
  - `Label`: Variable that can take any of the field types as a value



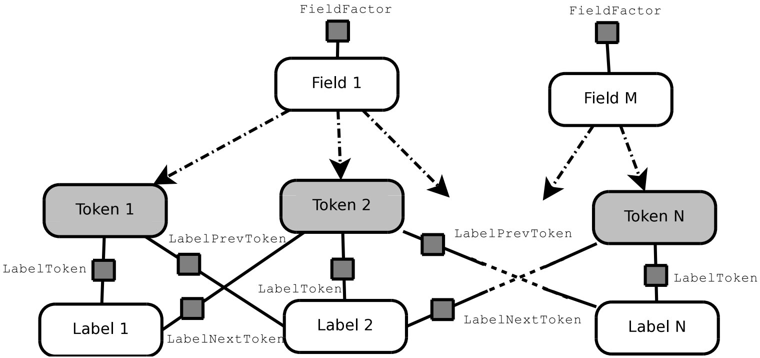
Segmentation

- **Variables**
  - `Token`: Observed variable representing a word in the mention
  - `Label`: Variable that can take any of the field types as a value
  - `Field`: Consecutive `Tokens` that have the same label type



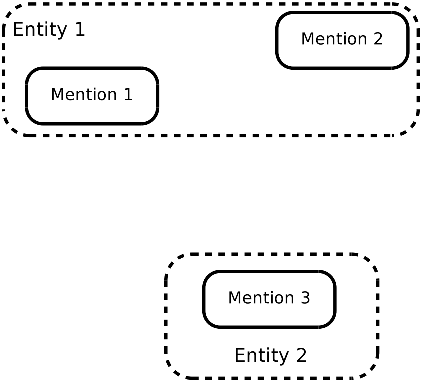
Segmentation

- **Variables**
  - `Token`: Observed variable representing a word in the mention
  - `Label`: Variable that can take any of the field types as a value
  - `Field`: Consecutive `Tokens` that have the same label type
- **Factors:** `LabelToken`,`LabelPrev/NextToken`, `FieldFactor`



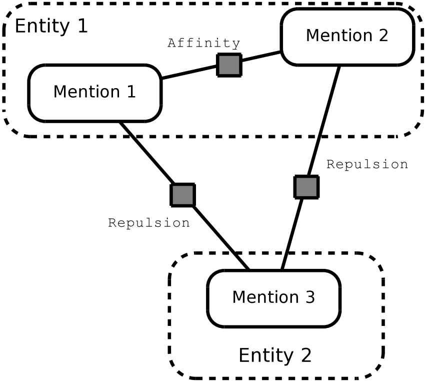
Entity Resolution

- **Variables**
  - `Mention`: Variable that takes a single `Entity` as its value
  - `Entity`: Set of `Mentions` that are coreferent



Entity Resolution

- **Variables**
  - `Mention`: Variable that takes a single `Entity` as its value
  - `Entity`: Set of `Mentions` that are coreferent
- **Factors:** `Affinity` and `Repulsion`

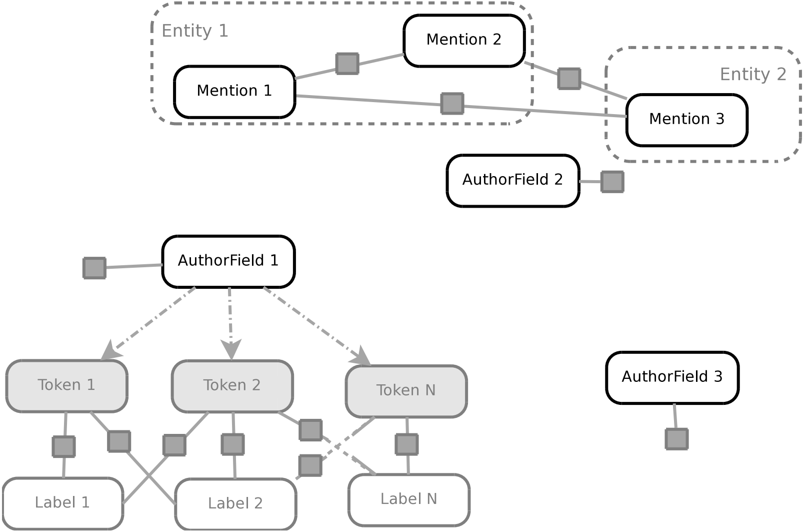


Integrating the Two Tasks

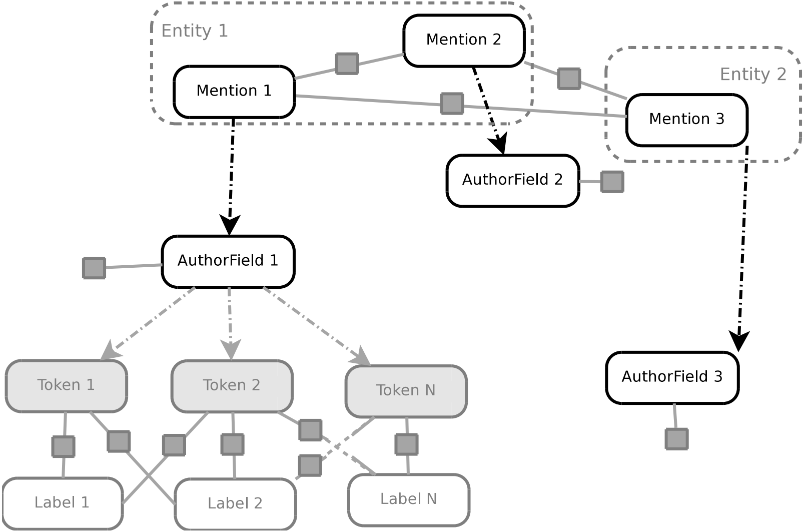
- **Variables**
  - No additional variables are required
  - `Field` variables are added as members of `Mention` variables
- **Joint Factors:** connect variables of different tasks
  - `JointInfBased:`
    - Connect identical trigrams of `Tokens` between two `Mentions` where the trigram is preceded by punctuation in only one of the `Mentions`[‖]
    - Forms a weak connection between the tasks since it is sparse, and does not take the entire predicted `Field` into account
  - `JointAffinity, JointRepulsion:`
    - Connect corresponding `Fields` between pairs of `Mentions`
    - Utilize features computed over the full predicted `Fields` between `Mention` pairs (e.g., string similarity, number of matching `Tokens`)

[‖] Poon & Domingos AAAI 2007

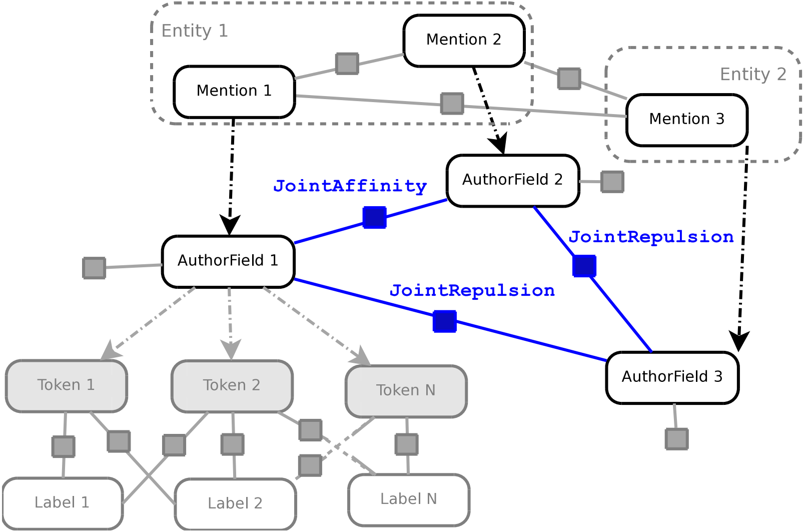
Example Model



Example Model



Example Model



The Advantages of IDFs

1. Joint factor templates can make inference intractable
   - `JointAffinity` and `JointRepulsion` factor templates have $O(m^2 n^4)^{**}$ instances in a fully unrolled graph

** $m =$ number of mentions, $n =$ number of tokens in a mention

The Advantages of IDFs

1. Joint factor templates can make inference intractable
   - `JointAffinity` and `JointRepulsion` factor templates have $O(m^2 n^4)^{**}$ instances in a fully unrolled graph
   - IDFs allow such factors through imperative structure definition and on-the-fly feature calculation
   - Evaluating a new sample requires re-scoring only m such factors

$^{**}m =$ number of mentions, $n =$ number of tokens in a mention

The Advantages of IDFs

1. Joint factor templates can make inference intractable
   - `JointAffinity` and `JointRepulsion` factor templates have $O(m^2 n^4)^{**}$ instances in a fully unrolled graph
   - IDFs allow such factors through imperative structure definition and on-the-fly feature calculation
   - Evaluating a new sample requires re-scoring only m such factors
2. The proposal function utilizes domain knowledge to implicitly define and efficiently explore the feasible region

** m = number of mentions, n = number of tokens in a mention

The Advantages of IDFs

1. Joint factor templates can make inference intractable
   - `JointAffinity` and `JointRepulsion` factor templates have $O(m^2 n^4)^{**}$ instances in a fully unrolled graph
   - IDFs allow such factors through imperative structure definition and on-the-fly feature calculation
   - Evaluating a new sample requires re-scoring only m such factors
2. The proposal function utilizes domain knowledge to implicitly define and efficiently explore the feasible region
3. Factor templates leverage the flexible separation of data representation and parameterization provided by IDFs
   - E.g., a `Field` is most naturally represented as a range over `Tokens`, and the compatibility between `Field` pairs is easily parameterized by a `JointAffinity` factor

** m = number of mentions, n = number of tokens in a mention

Outline

Experimental Setup

- Cora citation dataset[††]
  - 1,295 mentions, 134 clusters, 36,487 tokens
  - Evaluated using three-fold cross-validation
- **Isolated Models**
  - Each task is completely independent of the other
  - Learn with 5 loops of 100,000 MCMC samples each
  - Inference for 300,000 MCMC samples per task
- **Joint Models**
  - Single model over both the tasks
  - Learn with 5 loops of 250,000 MCMC samples each
  - Inference for 750,000 MCMC samples
- Results are compared to Poon and Domingos' previous state-of-the-art isolated and joint Markov logic networks

[††] Available at http://alchemy.cs.washington.edu/papers/poon07

Model Performance

Table: Cora Entity Resolution: Pairwise F1 and Cluster Recall

Method	Prec/Recall	F1	Cluster Rec.
Fellegi-Sunter	78.0/97.7	86.7	62.7
Joint MLN	94.3/97.0	95.6	78.1
Isolated IDF	**97.09**/95.42	96.22	86.01
Joint IDF	95.34/**98.25**	**96.71**	**94.62**

25.2%

Table: Cora Segmentation: Tokenwise F1

Method	Author	Title	Venue	Total
Isolated MLN	99.3	97.3	98.2	98.2
Joint MLN	99.5	97.6	98.3	98.4
Isolated IDF	99.35	97.63	98.58	98.51
Joint IDF	**99.42**	**97.99**	**98.78**	**98.72**

20.0%

Model Performance

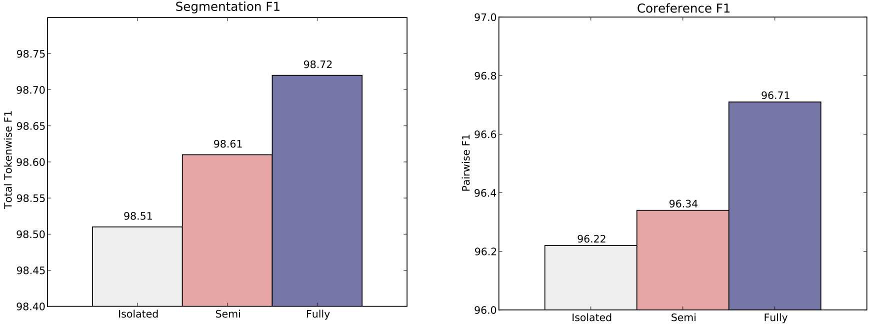
Table: Cora Entity Resolution: Pairwise F1 and Cluster Recall

Method	Prec/Recall	F1	Cluster Rec.	
Fellegi-Sunter	78.0/97.7	86.7	62.7	
Joint MLN	94.3/97.0	95.6	78.1	50-90 mins
Isolated IDF	**97.09**/95.42	96.22	86.01	~3 mins
Joint IDF	95.34/**98.25**	**96.71**	**94.62**	~18 mins

Table: Cora Segmentation: Tokenwise F1

Method	Author	Title	Venue	Total	
Isolated MLN	99.3	97.3	98.2	98.2	} 50-90 mins
Joint MLN	99.5	97.6	98.3	98.4	
Isolated IDF	99.35	97.63	98.58	98.51	~3 mins
Joint IDF	**99.42**	**97.99**	**98.78**	**98.72**	~18 mins

Bidirectionality



(a) Segmentation F1 (b) Entity Resolution F1

Figure: F1 of the joint model as different types of factors are added, starting with the base model containing only isolated model factors. "Semi-Joint" refers to the model containing *weakly* joint factors while the "Fully-Joint" model consists of bi-directional highly-coupled factors.

Summary

- Introduce Imperatively Defined Factor graphs (IDFs)
  - efficient learning and inference on complex factor graphs
- Utilize IDFs for increased influence between tasks
- Demonstrate significant error reduction and time improvements

- Future Work:
  - Joint model for more than two tasks
  - Extend to non-MCMC based inference
  - Other applications

Thanks!

Sameer Singh, Karl Schultz, Andrew McCallum
University of Massachusetts, Amherst

{sameer,kschultz,mccallum}@cs.umass.edu

Visit us at the poster session