Grammarless Parsing for Joint Inference

Jason Naradowsky
joint work with Tim Vieira[JHU] and David Smith[NE]

UMass Amherst & Macquarie University

Outline

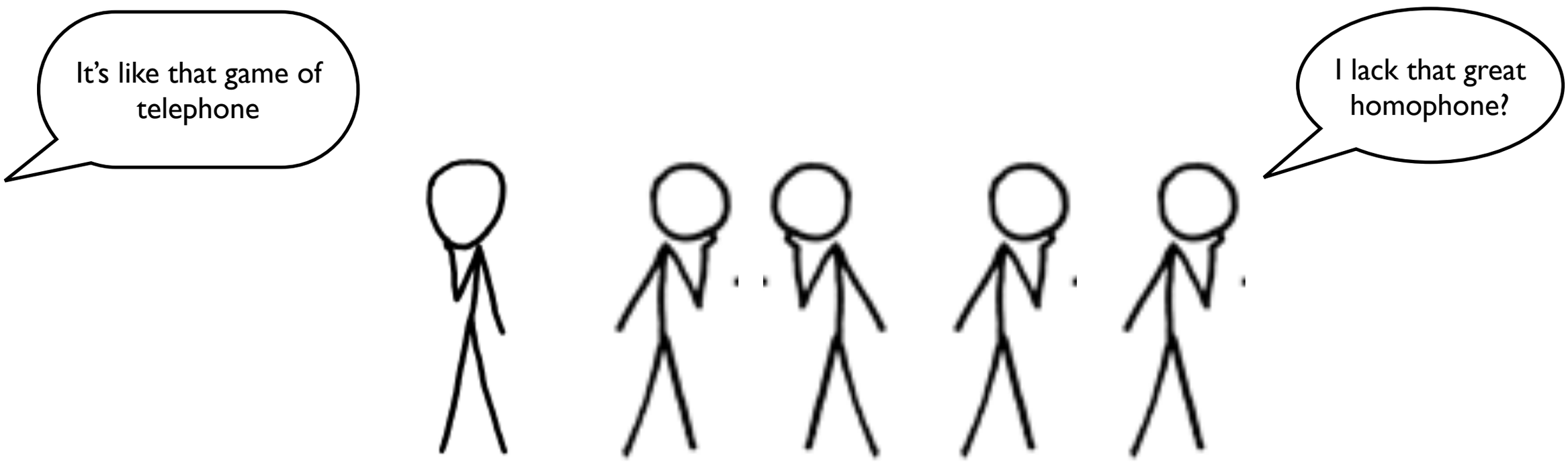
- Joint Inference
- Factor graphs and Message Passing Inference
- Modeling Syntax with Combinatorial Factors
- Joint Models of NER and Syntax
- Conclusions

Towards Sci-fi Futures:

- Goal: Construct complete NLP systems
- *But*, most models for NLP tasks are developed independently for pipelined system.
- The problem?

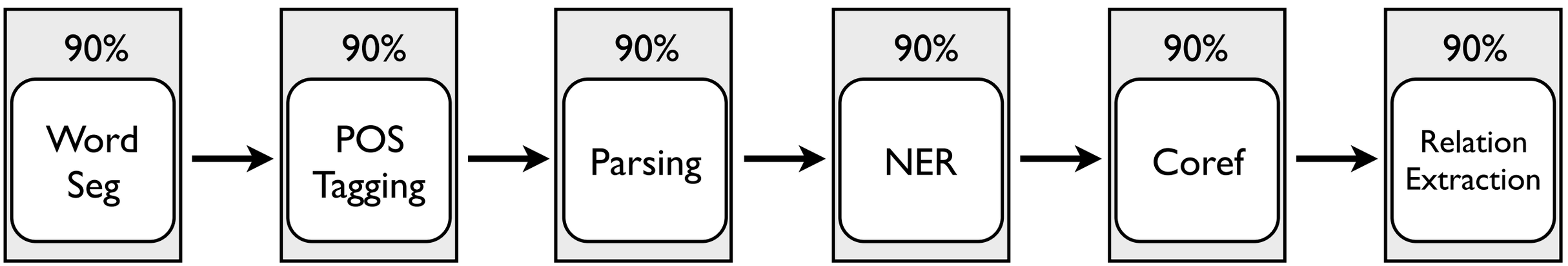
Towards Sci-fi Futures:

- Goal: Construct complete NLP systems
- *But*, most models for NLP tasks are developed independently for pipelined system.
- The problem? **Error Propagation**



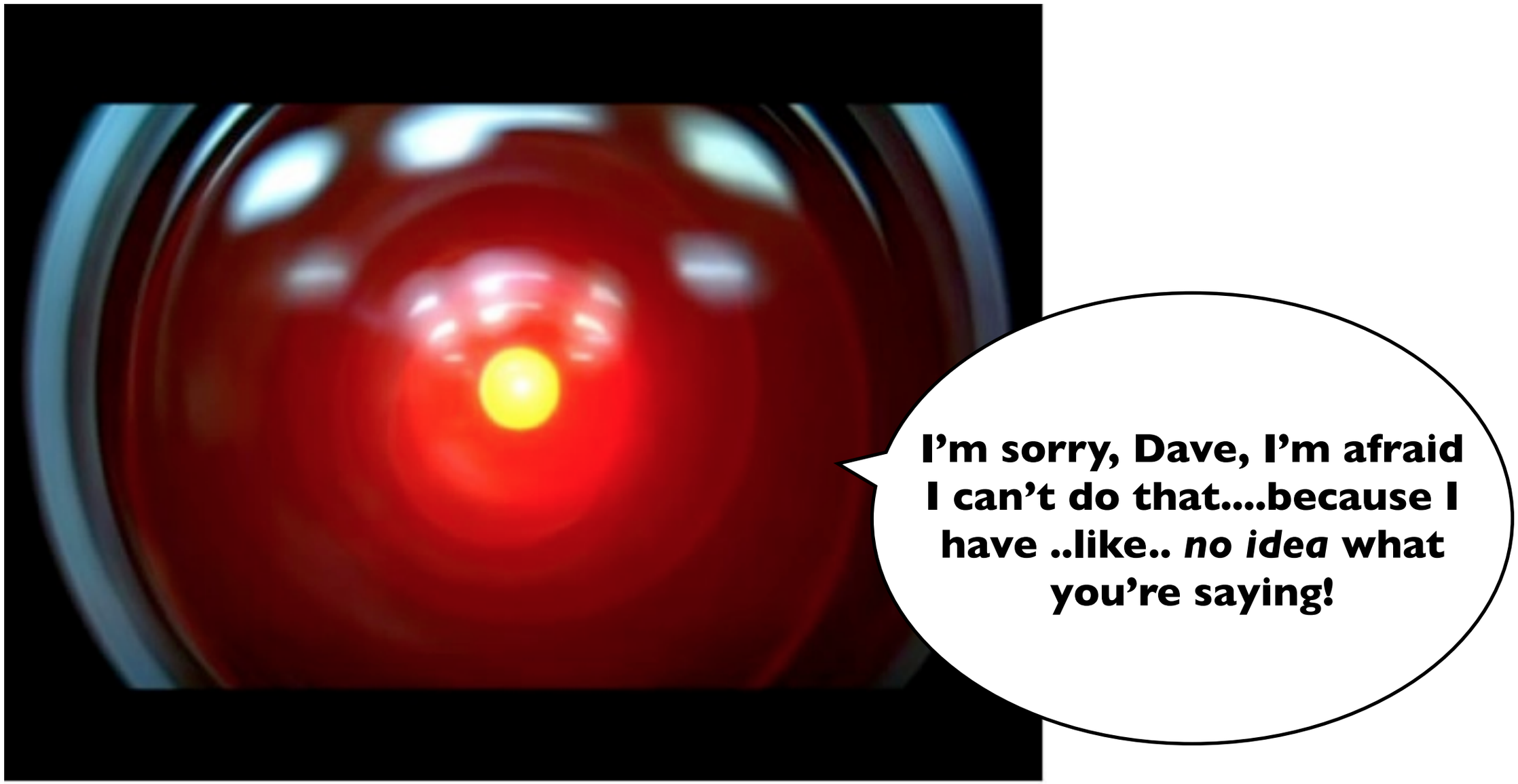
Towards Sci-fi Futures:

- Goal: Construct complete NLP systems
- *But*, most models for NLP tasks are developed independently for pipelined system.
- The problem? **Error Propagation**



= 53% Accuracy on End Task! [McCallum 09]

Towards Sci-fi Futures:



Can we do better?

Joint Inference [advantages]

- Combine the models for independent problems and perform inference over the combined model.
- Minimizes inconsistencies between models and prevents error propagation.
- Joint modeling has shown improvement in:
  - parsing and NER [Finkel & Manning 09]
  - parsing and morphology [Lee, et al 2011]
  - NER and coref [Haghighi 2010]
- Why then...don't we always use it?

Joint Inference [disadvantages]

- Complexity:
  - requires more coding, more troubleshooting, longer training
- Representation:
  - all models need to be implemented in a in a single framework
  - frameworks convenient for one problem are often inconvenient for another

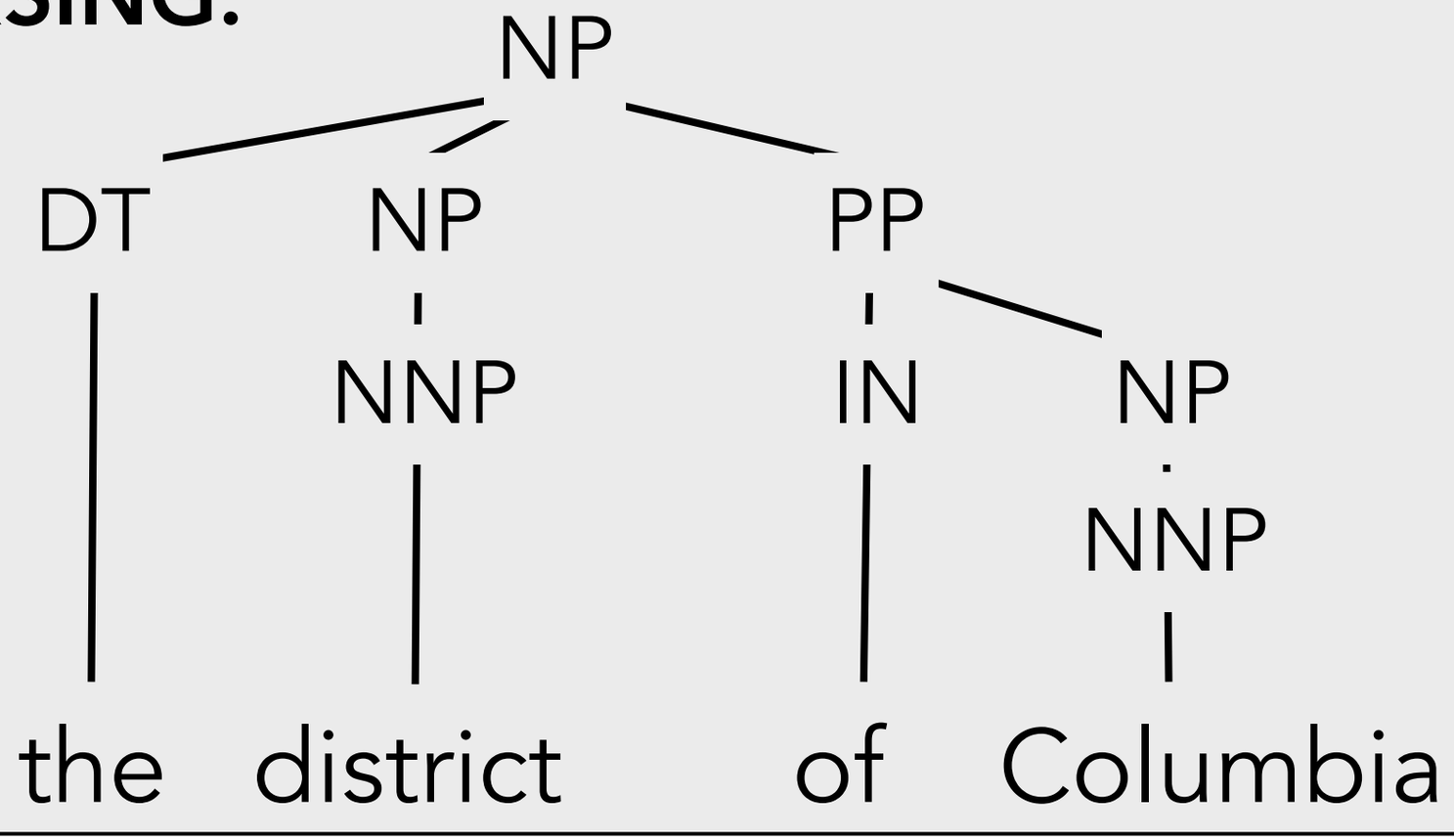
Joint Inference [example]

- Two Tasks:

NER:

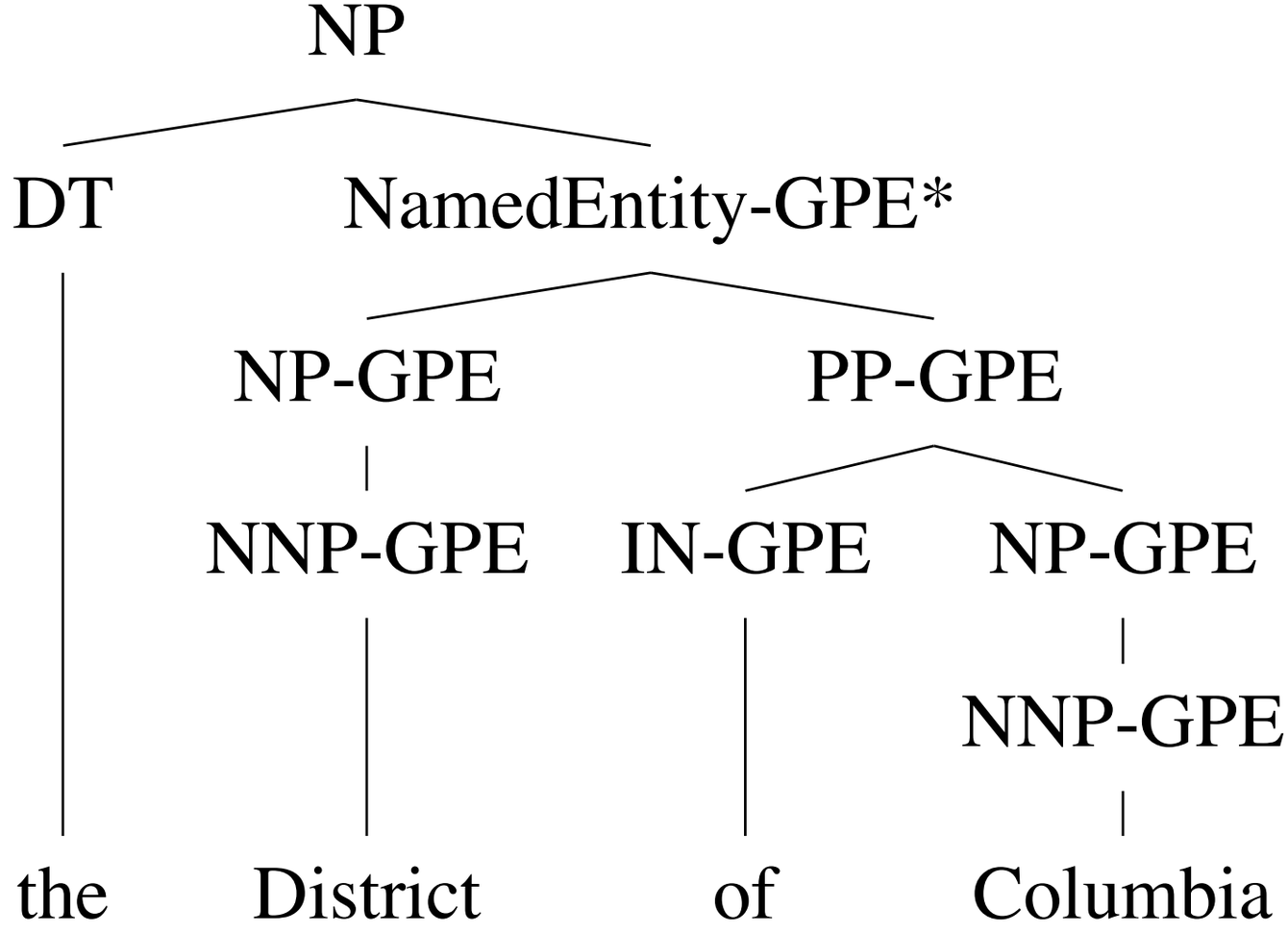
			GPE	
the	[district		of	Columbia]

PARSING:



Grammar Augmentation

- Conflate two tasks by combining the label sets of both tasks and making one prediction [Finkel & Manning 09]



Grammar Augmentation

- Conflate two tasks by combining the label sets of both tasks and making one prediction
- Disadvantages:
  - Label set size (G) increases multiplicatively with each task; $O(|G|n^3)$ decodes
  - Problems must be describable via parsing

An Alternative? Factor Graphs

- Factor Graphs are capable of representing common NLP models
- It pairs them automatically with efficient inference techniques via belief propagation
- Flexible connections between parts of models (just add a factor between them!)
- Label set growth between tasks is only additive (vs. multiplicative in grammar augmentation approach to joint inference)

Outline

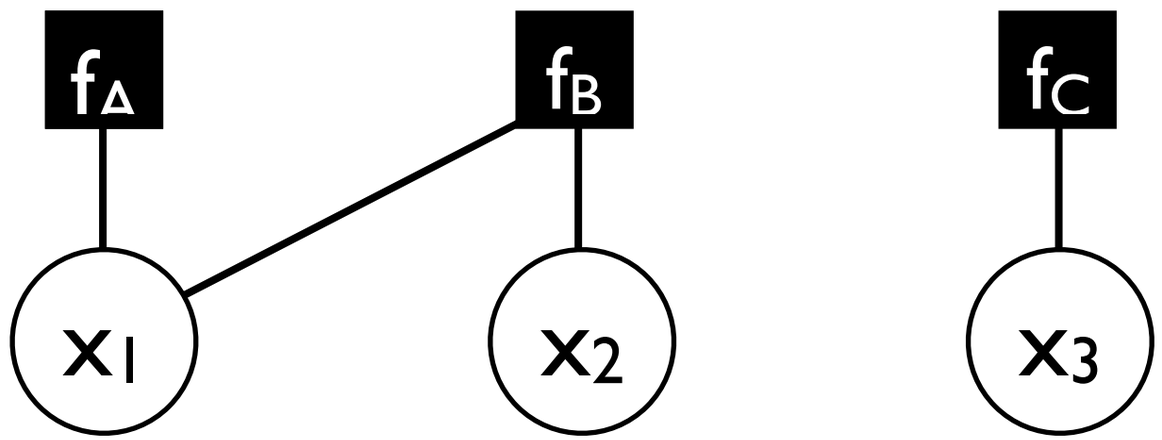
- Joint Inference
- Factor graphs and Message Passing Inference
- Modeling Syntax with Combinatorial Factors
- Joint Models of NER and Syntax
- Conclusions

Factor Graphs

- A probabilistic model
- Represents a factorization of a function over variables:

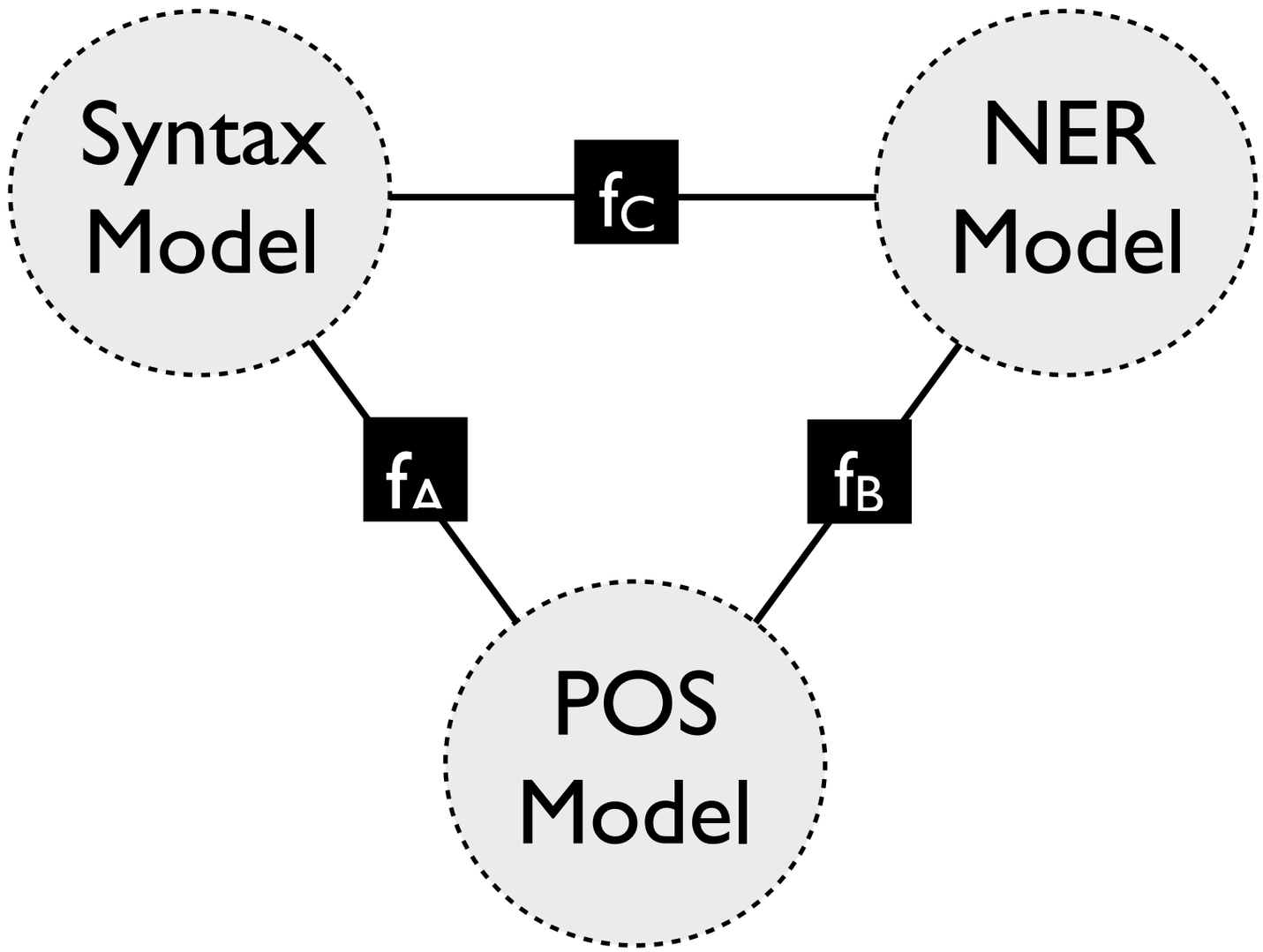
$$P(x_1,x_2,x_3) = f_A(x_1)f_B(x_1,x_2)f_C(x_3)$$

as a bipartite graph of factors and variables:

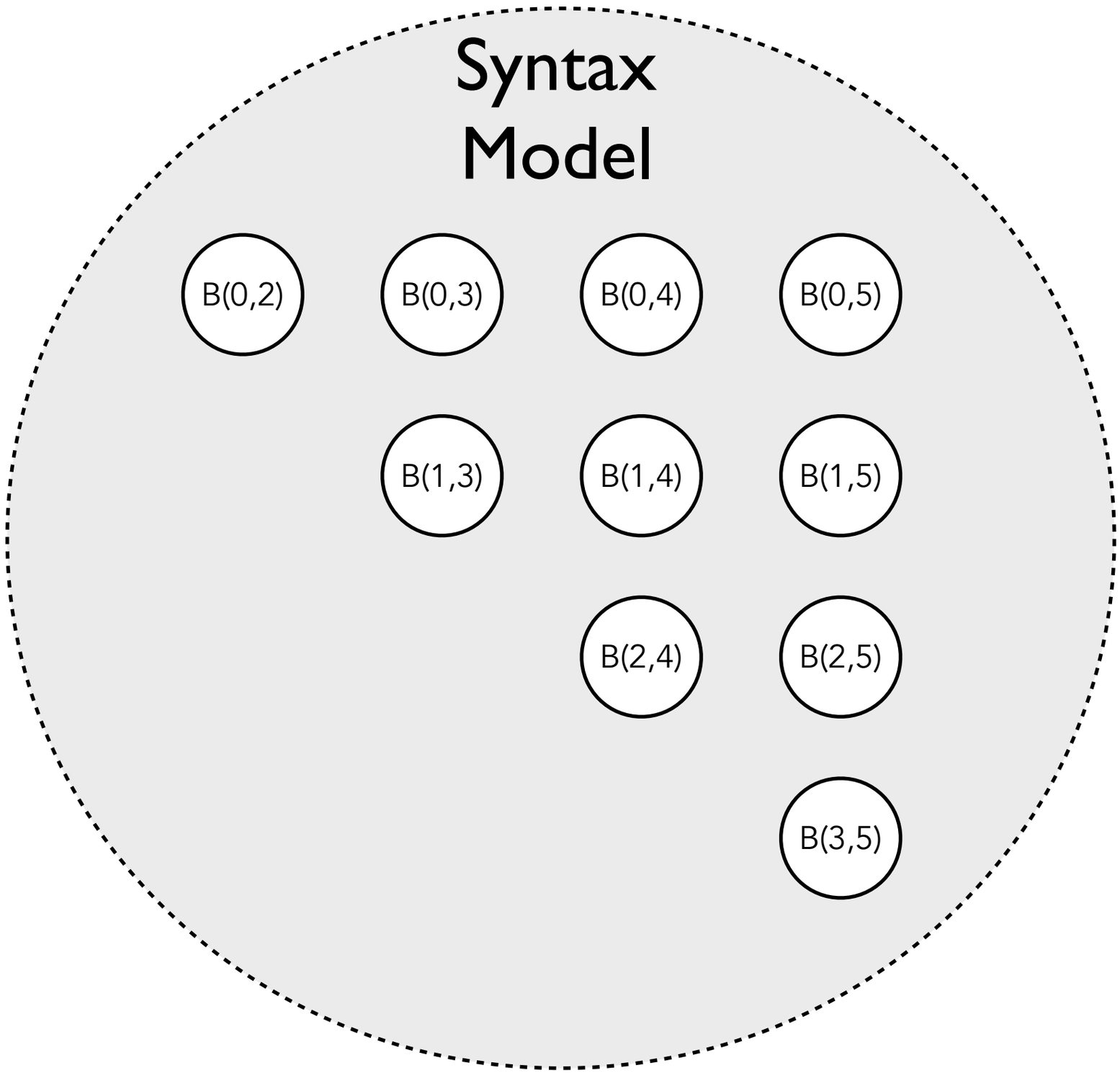


Factor Graphs

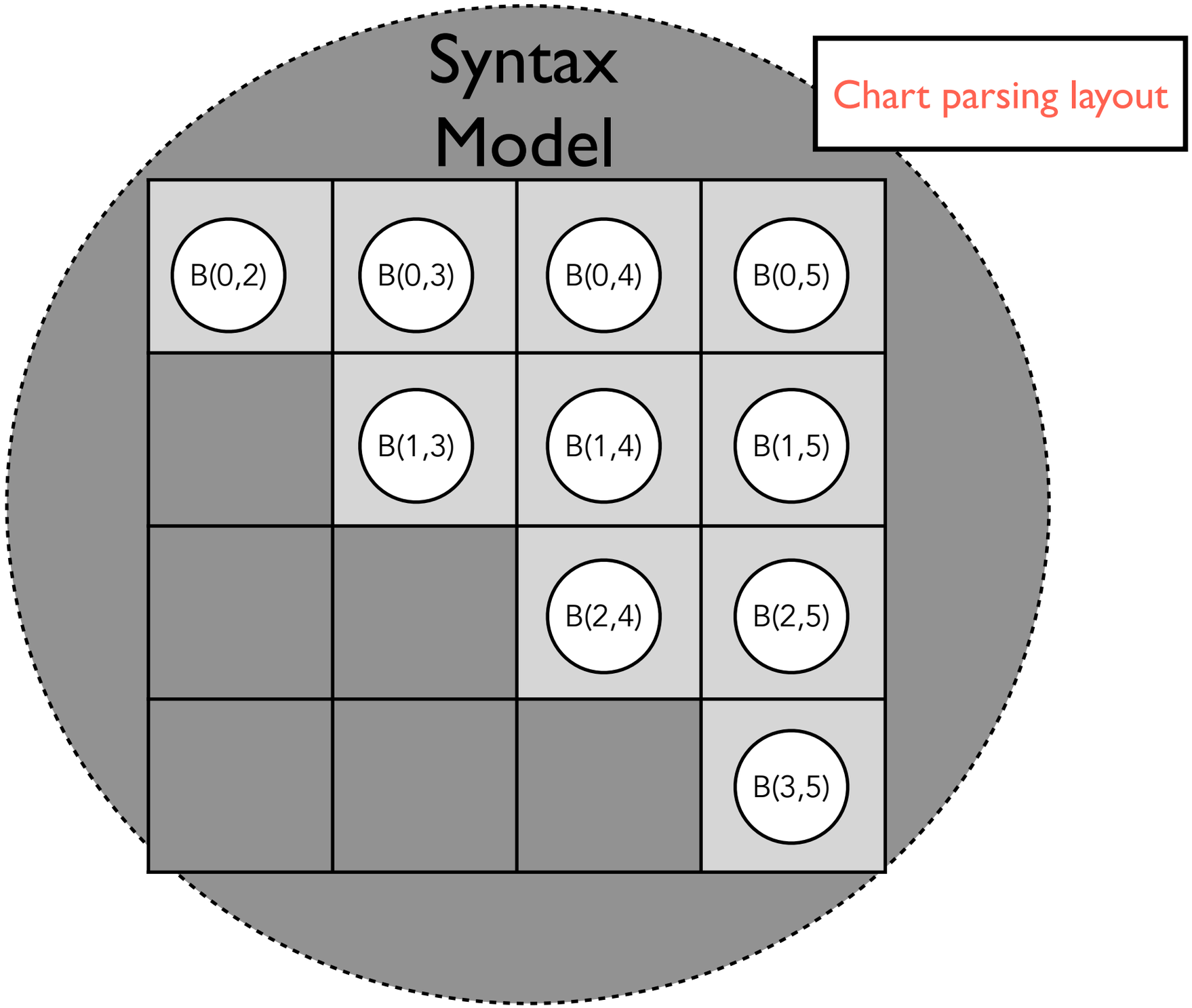
- Principled Method for Constructing Joint Models:



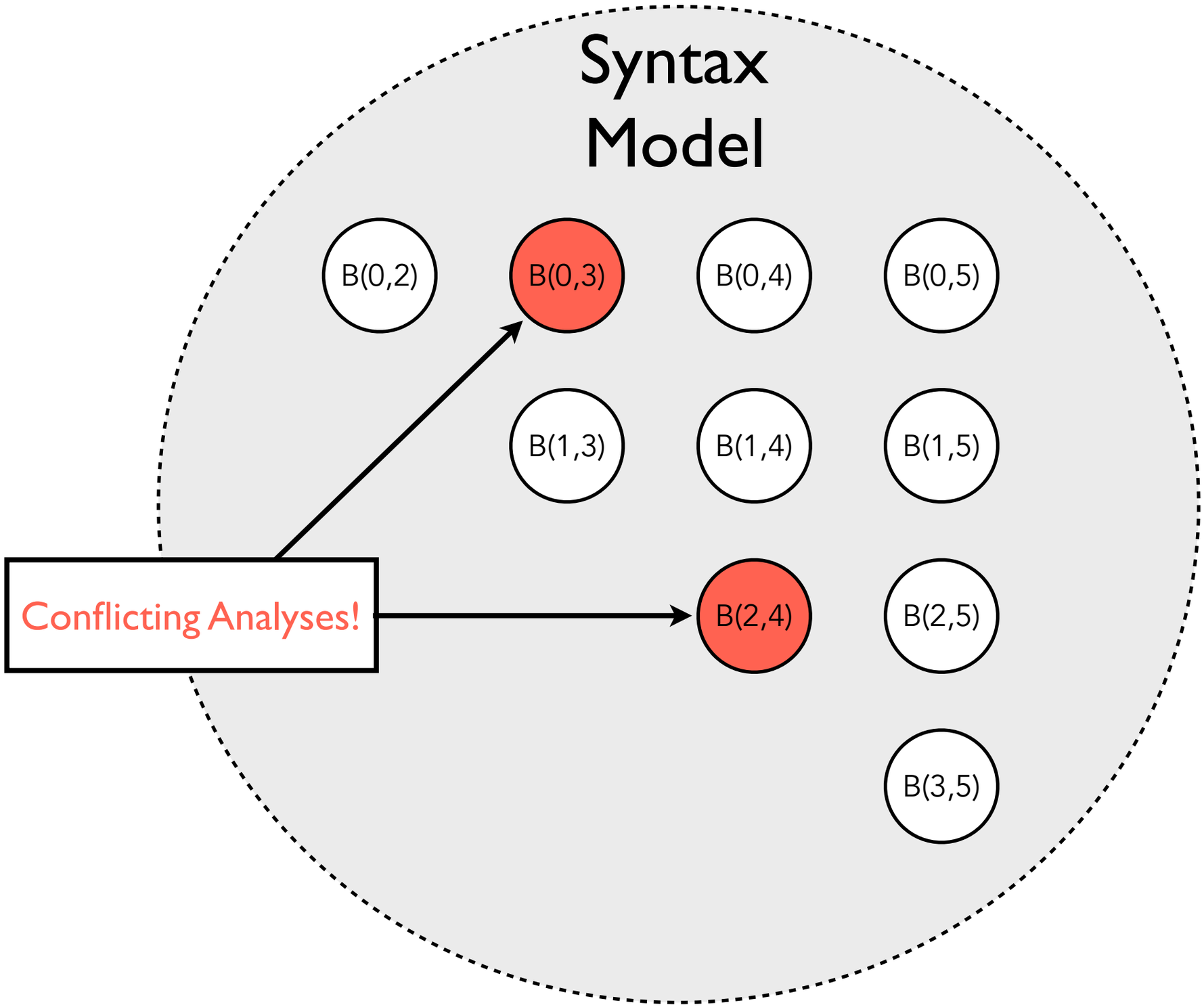
Factor Graphs



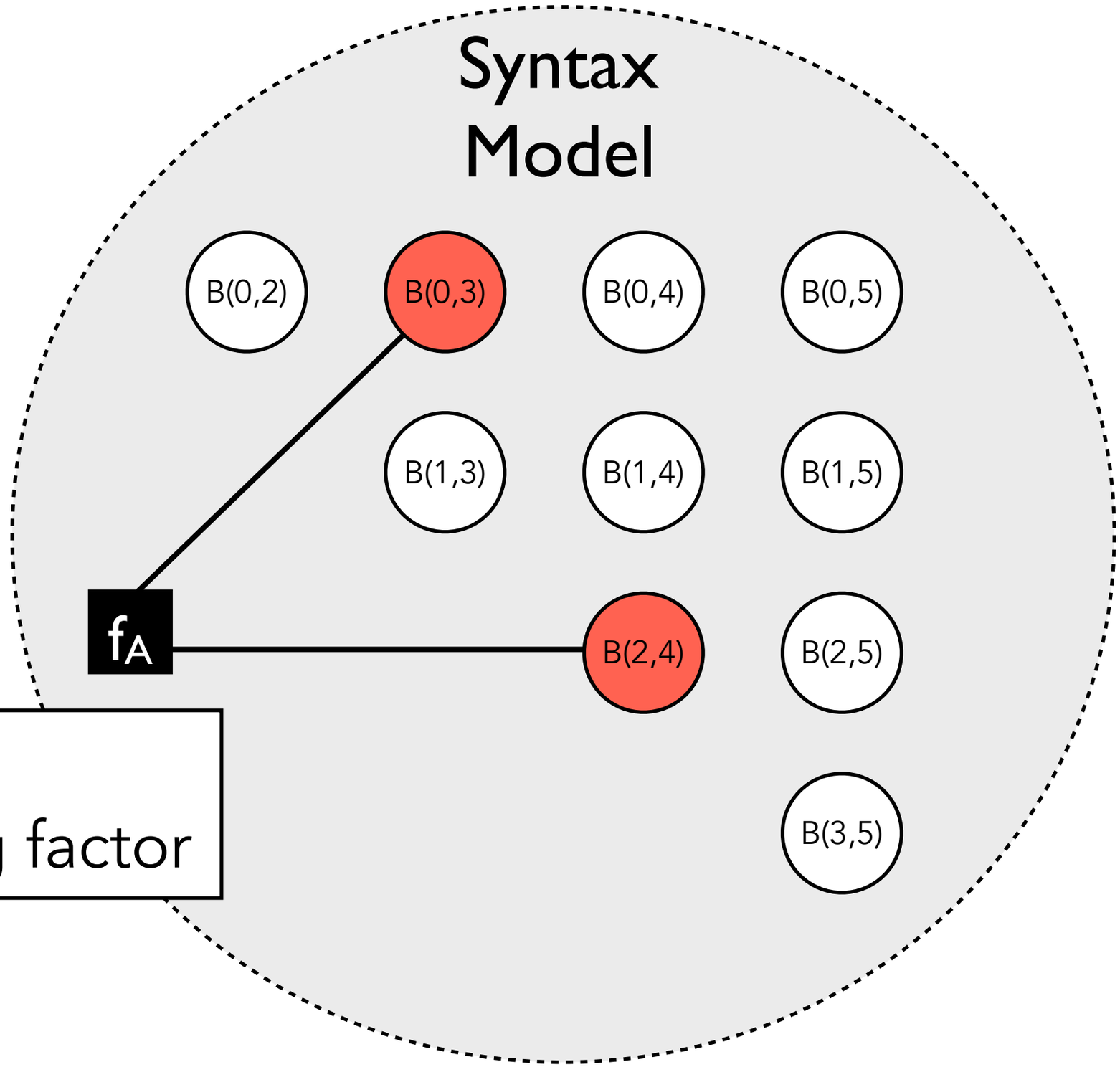
Factor Graphs



Factor Graphs

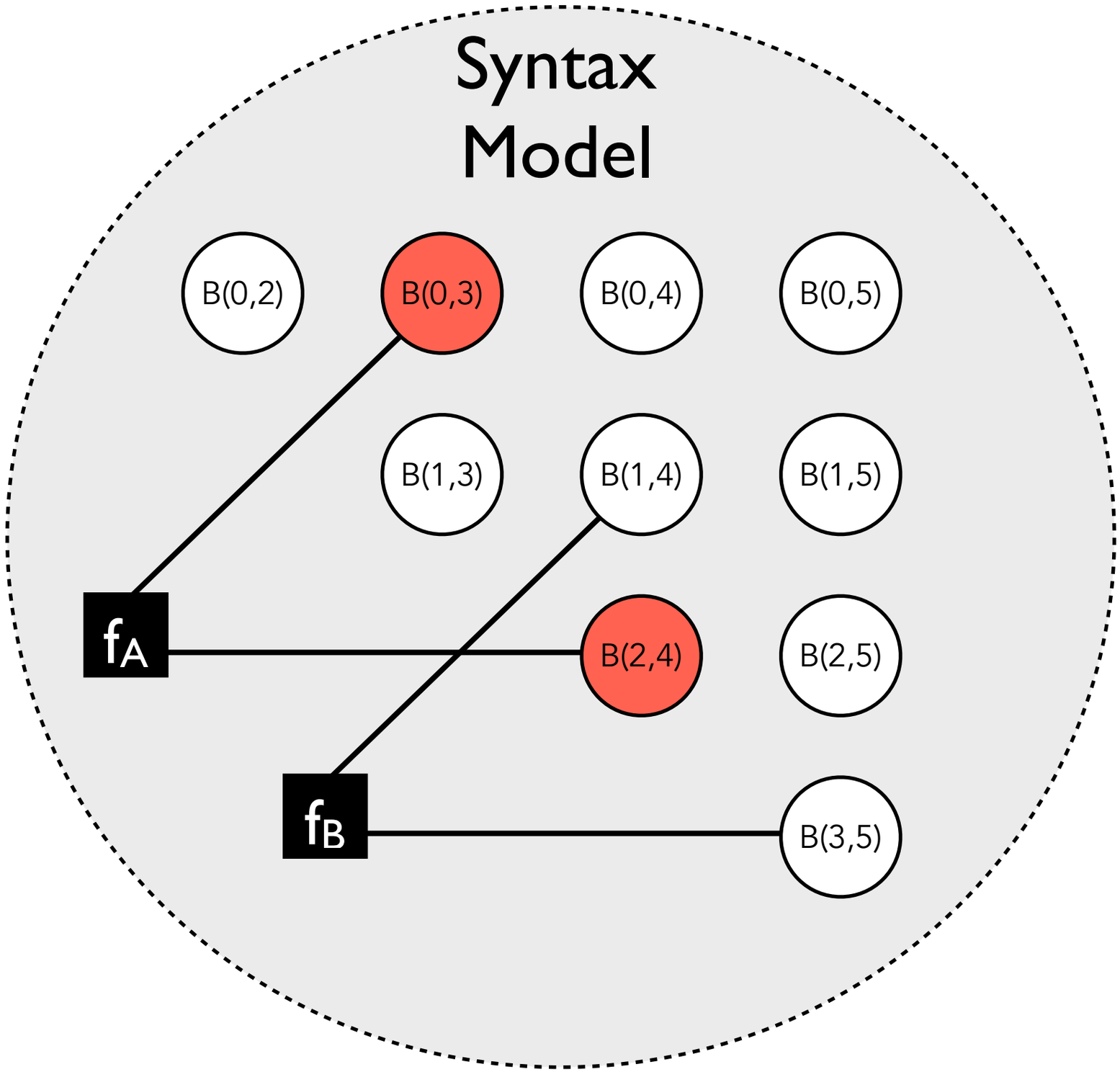


Factor Graphs

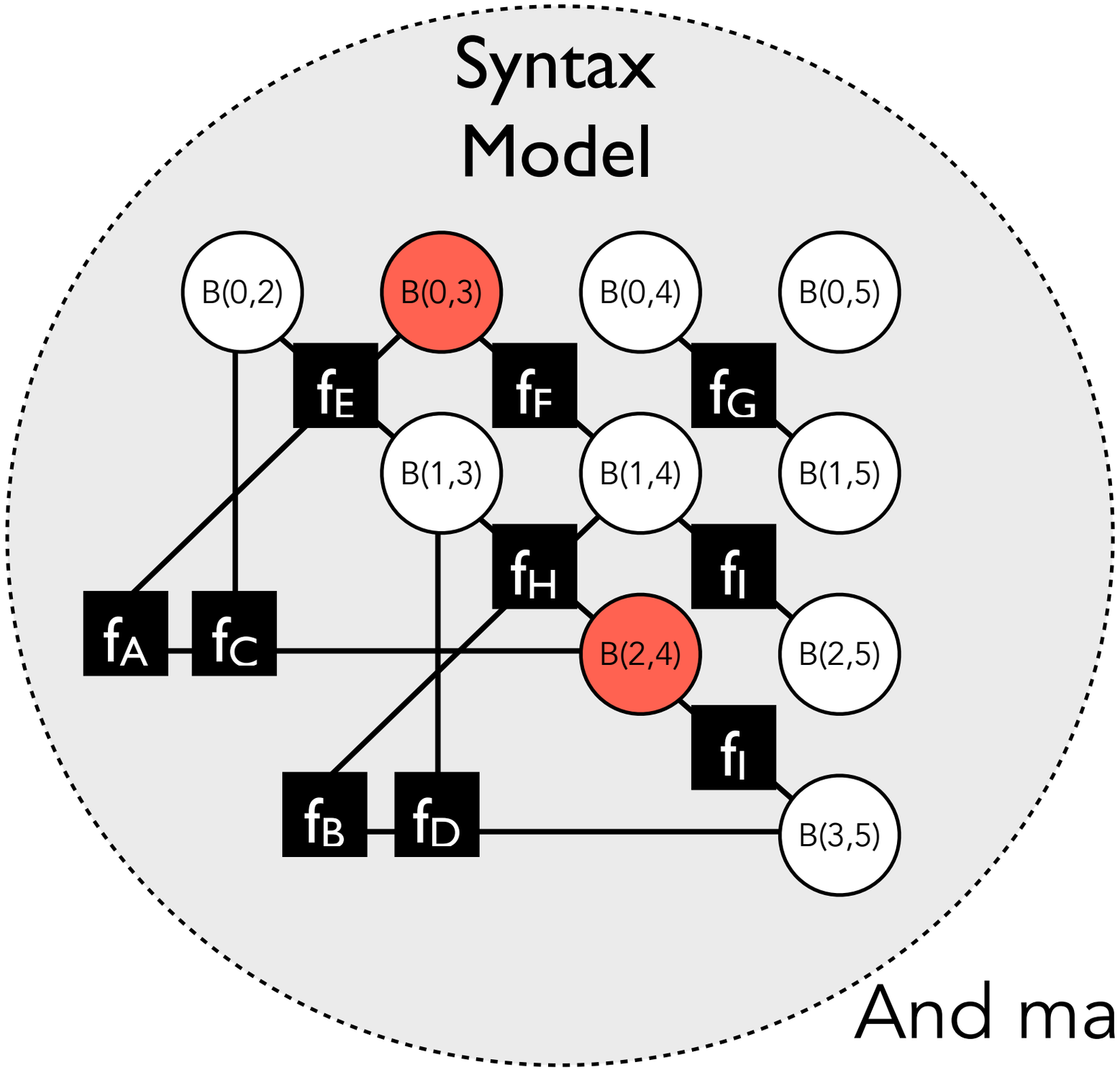


Prevent with coordinating factor

Factor Graphs



Factor Graphs



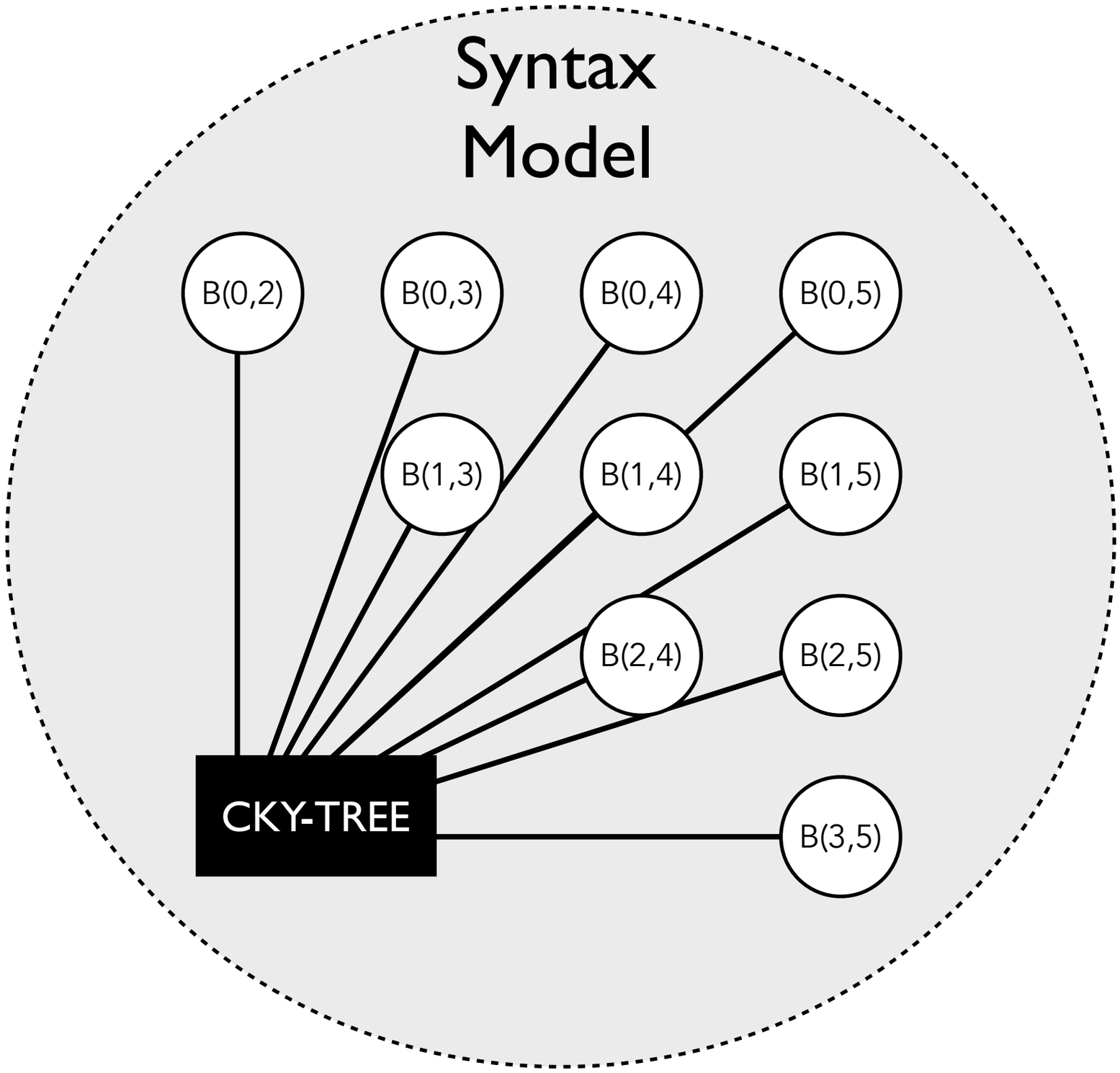
And many more!

Factor Graphs: problems?

- A densely connected clique creates a troublesome graphical model:
  - Exponential growth with sentence length
  - Lots of cycles = poor inference
  - Lots of messages = lots of computation
- Yet, there are many cubic time algorithms for constraining nested bracketings...

...can we do better?

Factor Graphs



Combinatorial Factors

- Sum-Product for Inference in Factor Graphs contains two types of messages:
  - Variables → Factors: $\prod_{\mathcal{N}(i)\setminus\{c\}} m_{\phi_{c'}\to Y_i}(y_i)$
  - Factors → Variables: $\sum_{y_{c\setminus\{i\}}} \phi_c(y_c) \prod_{i'\in c\setminus\{i\}} m_{Y_{i'}\to\phi_c}(y_{i'})$
- **Combinatorial Factors:**
  - Connect many variables (often globally)
  - Override default behavior
  - Implement efficient combinatorial algorithms

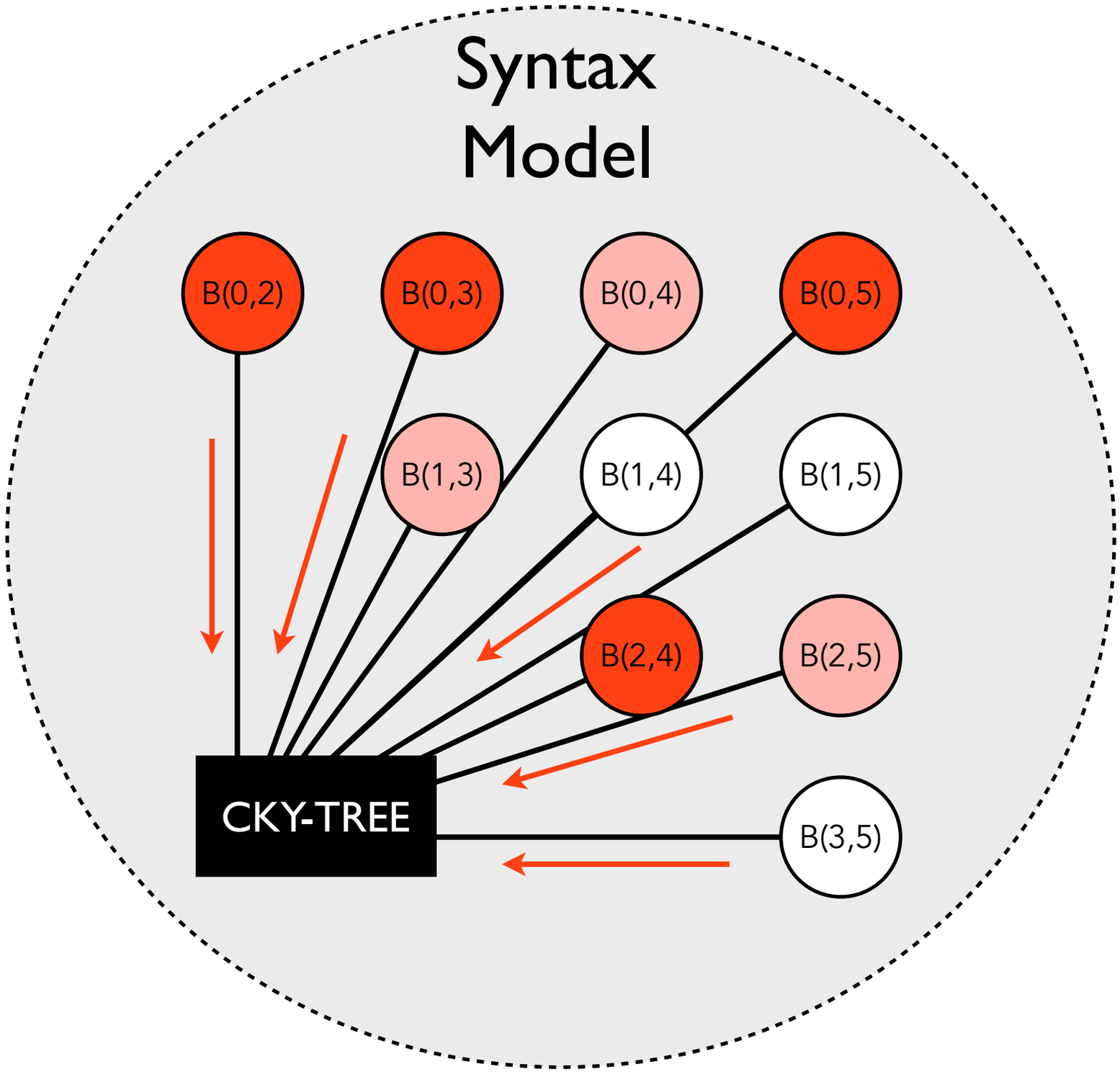
Outline

- Joint Inference
- Factor graphs and Message Passing Inference
- Modeling Syntax with Combinatorial Factors
- Joint Models of NER and Syntax
- Conclusions

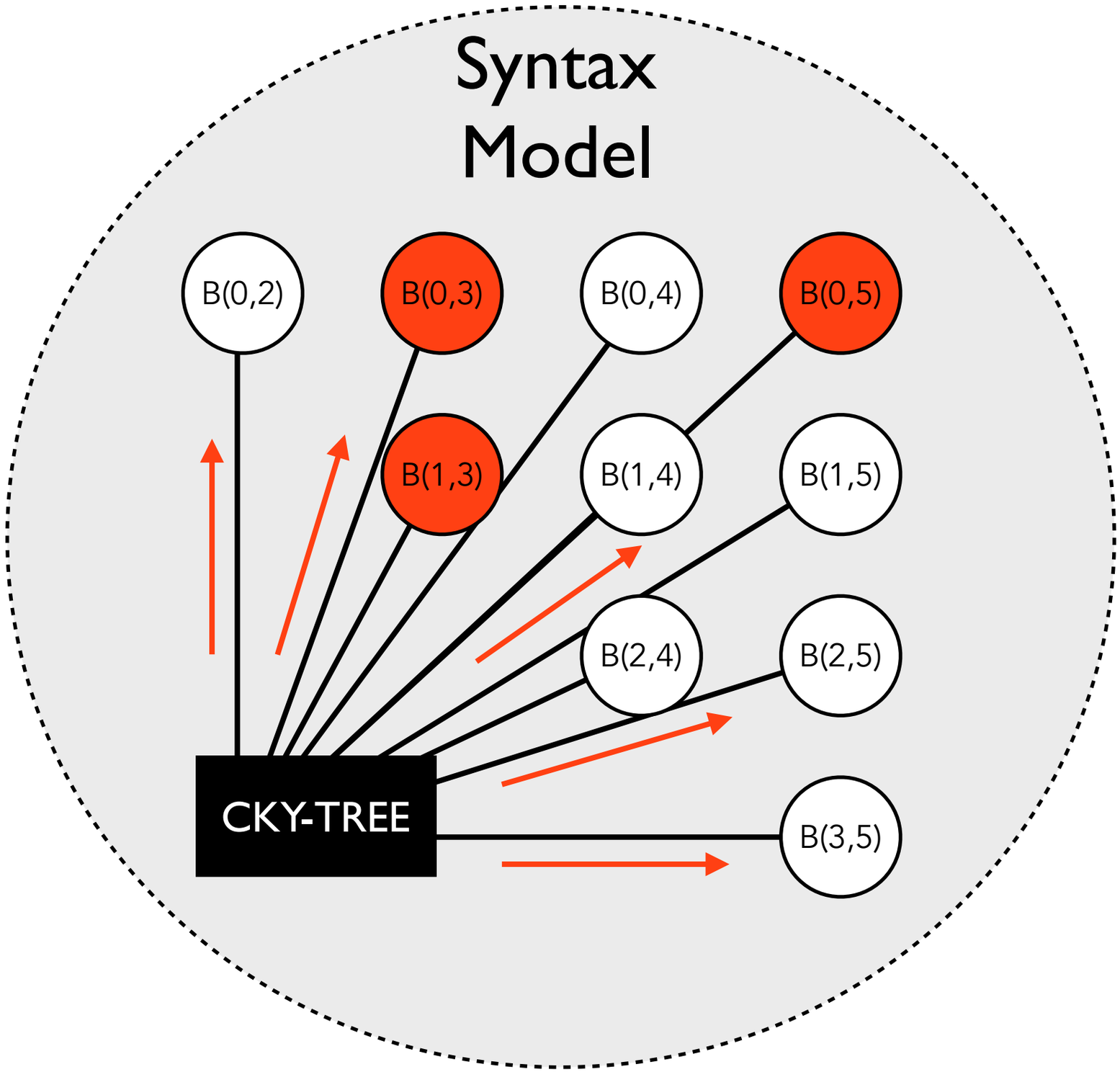
Parsing in Factor Graphs

- CKY-Tree
  - A Combinatorial Factor for constraining span variables to form a properly nested tree.
  - Implementation is simply inside-outside algorithm, computing expectations over spans without any grammatical rules.
    - Cubic time
- The global factor coordinating boolean variables produces a parser yielding *unlabeled* trees.

Factor Graphs



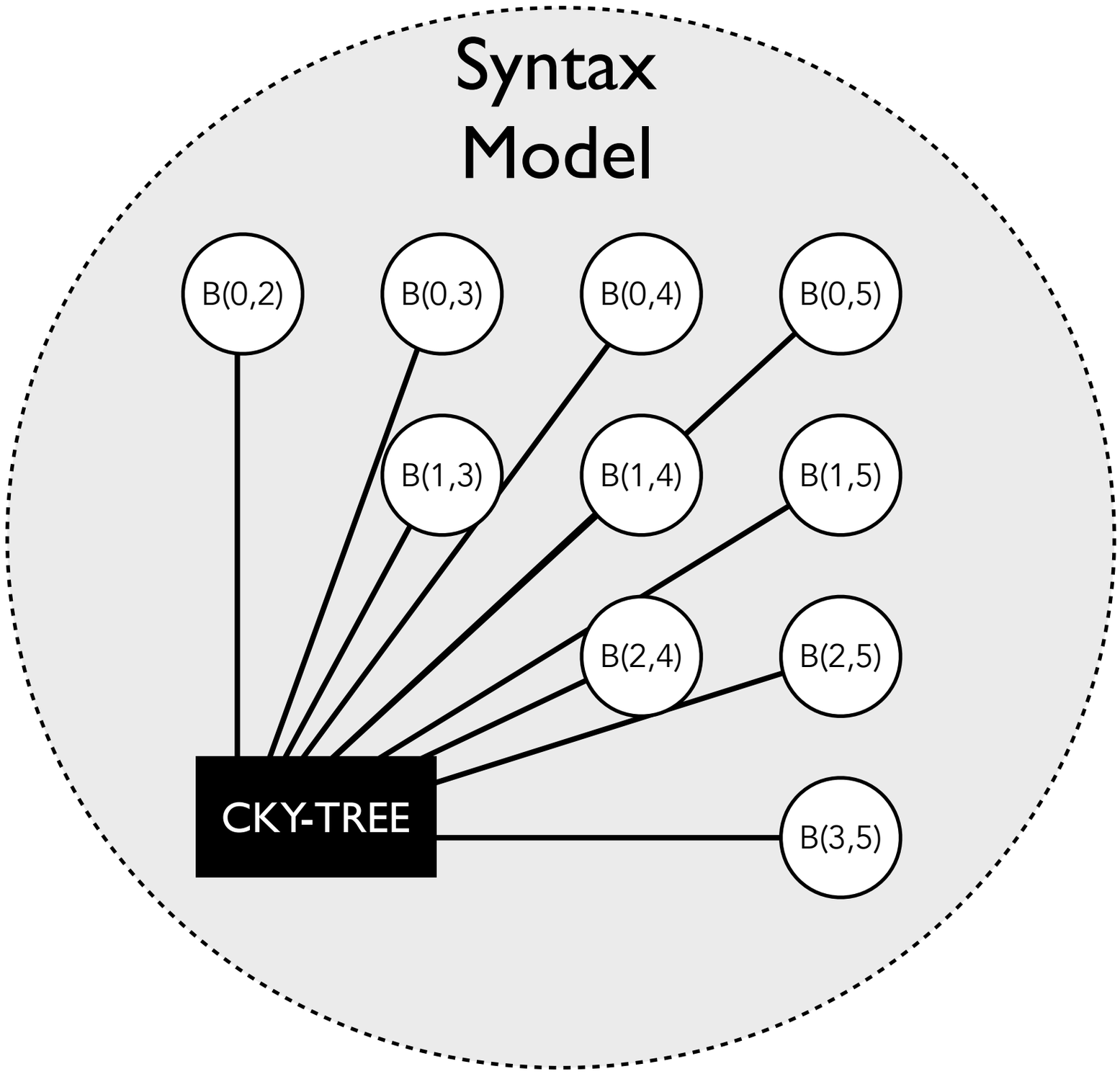
Factor Graphs



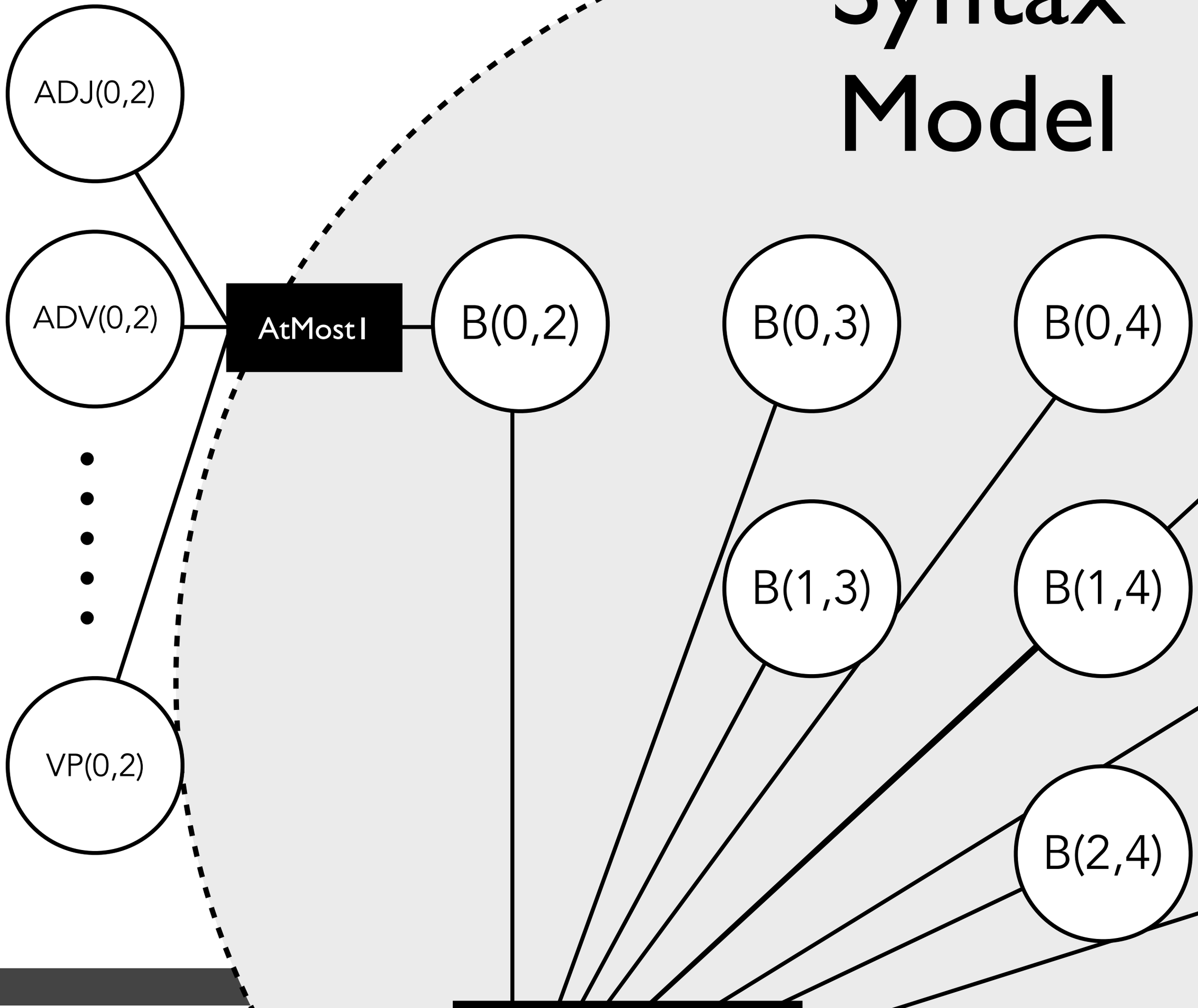
Extending to Labeled Parsing

- Bracket Model:
  - CKY-Tree
  - n^2 boolean span variables
- Labeled Bracket Model (LBM)
  - + |L| n^2 boolean label variables
  - + n^2 AtMost1 factors connecting label variables to span variables:
    - When span variable is on, assign a label
    - When span variable is off, don't

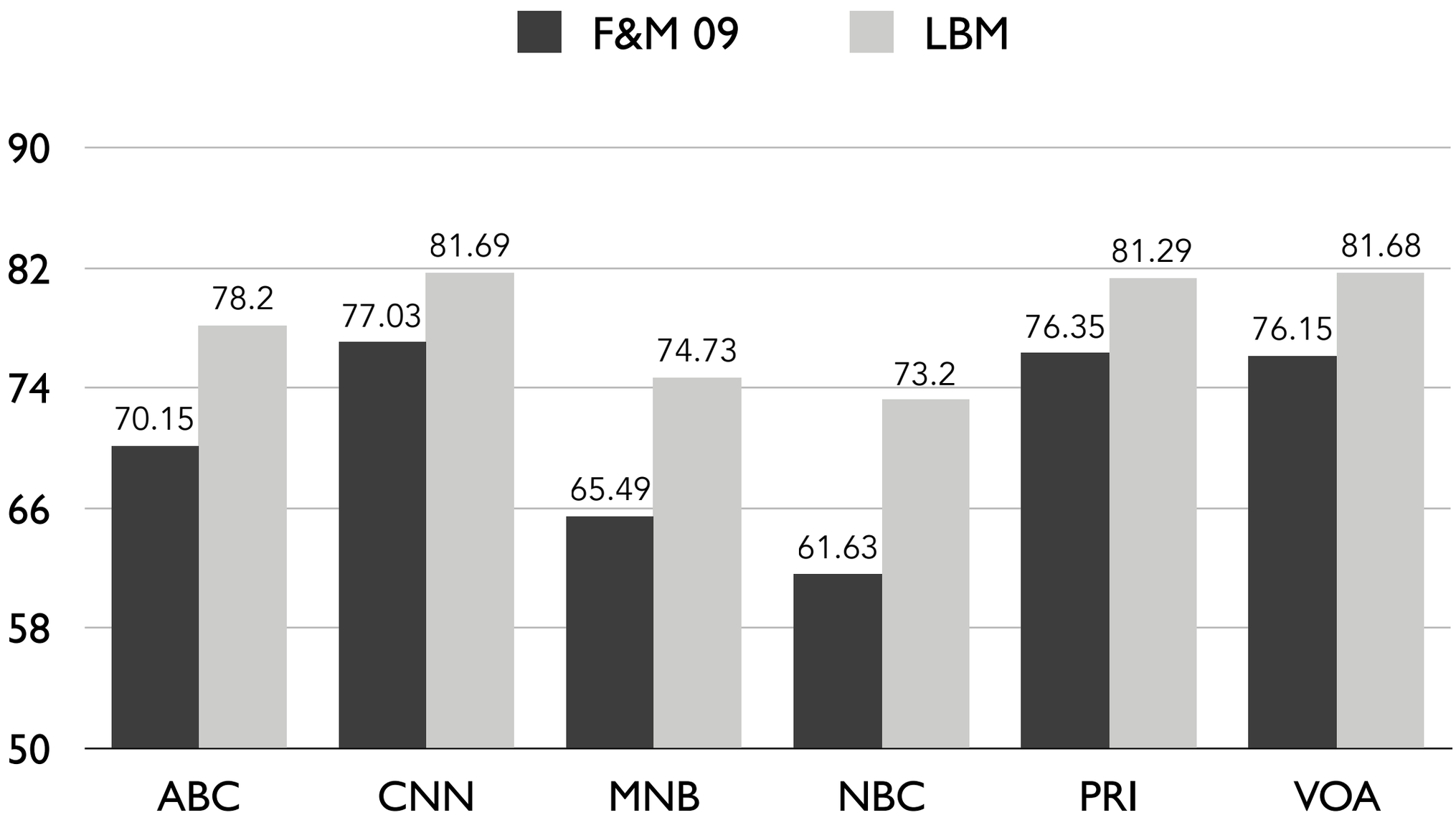
Factor Graphs



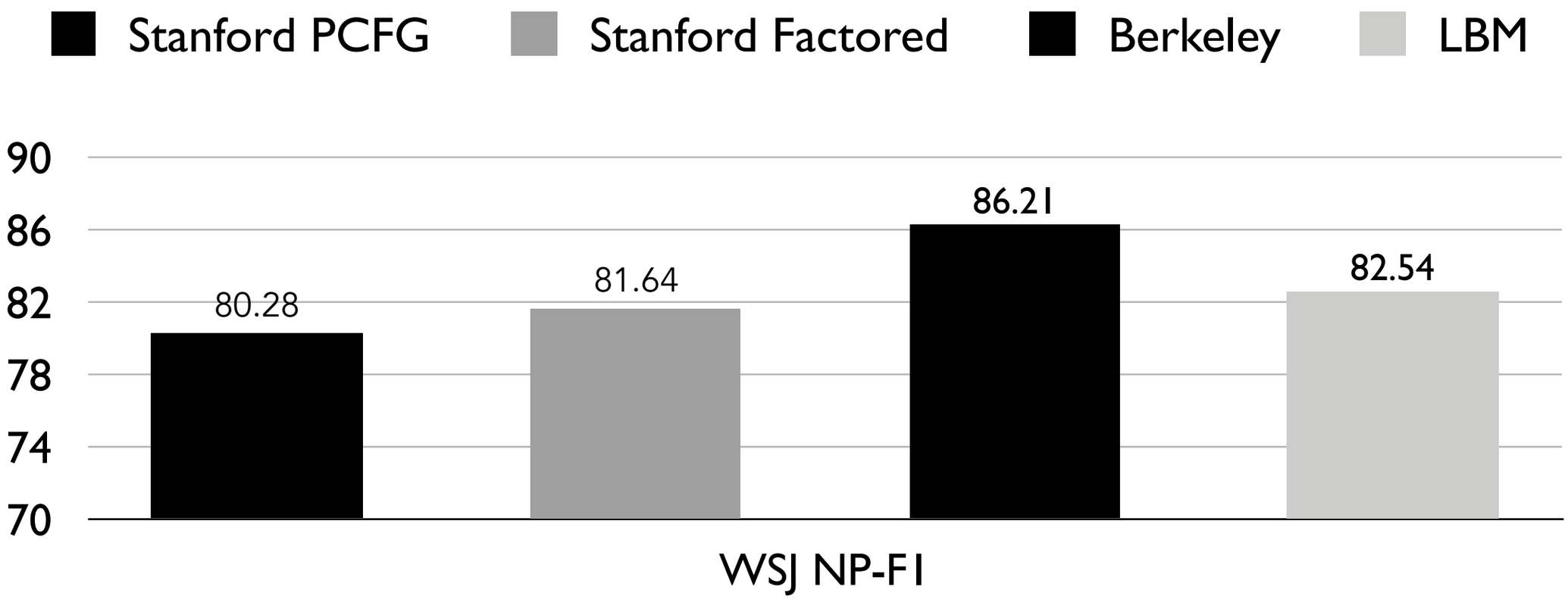
Syntax Model



Parsing Results



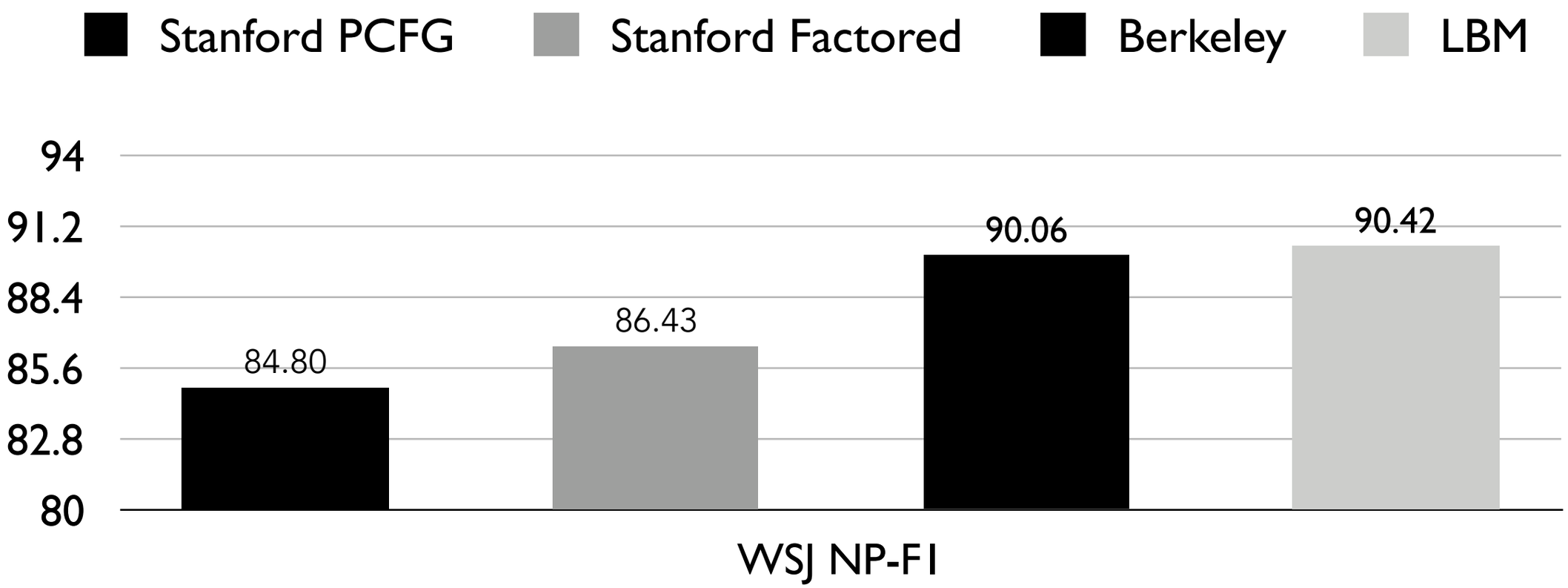
Accuracy on WSJ*



- Stanford parsers with horizontal and vertical markov order of 1

* Results from Ontonotes distribution of WSJ, which differs from historical release.

NP Accuracy on WSJ*



- Why care about NP accuracy in particular?

* Results from Ontonotes distribution of WSJ, which differs from historical release.

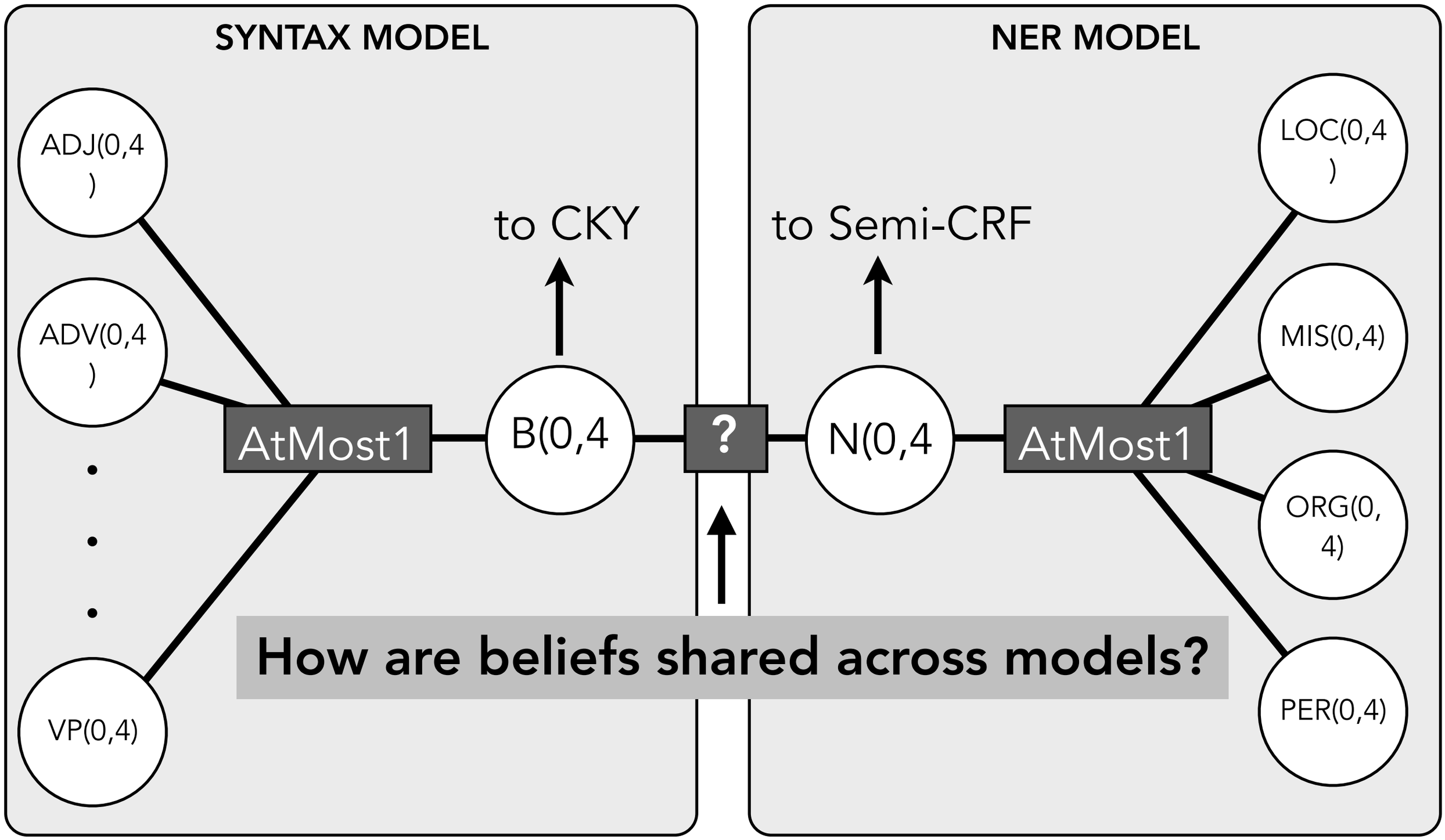
Outline

- Joint Inference
- Factor graphs and Message Passing Inference
- Modeling Syntax with Combinatorial Factors
- Joint Models of NER and Syntax
- Conclusions

Joint Parsing & NER

- A Strong Correlation:
  - Whenever we find a named-entity, we know there is a constituent NP span in the syntax
  - Whenever we find a non-NP constituent span, we know a named-entity cannot exist for that span
- Named entities cannot cross syntactic spans

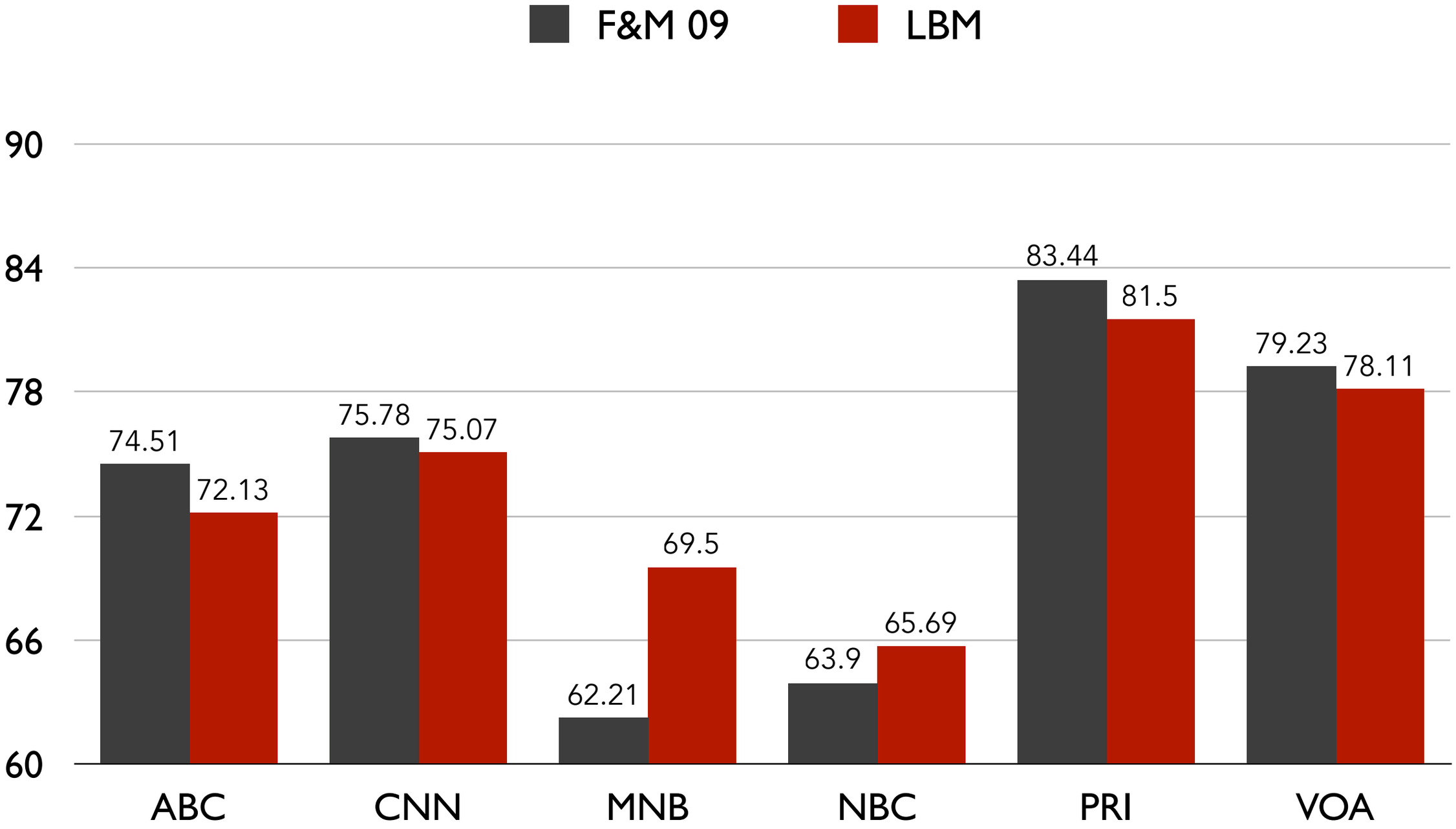
Joint Parsing & NER



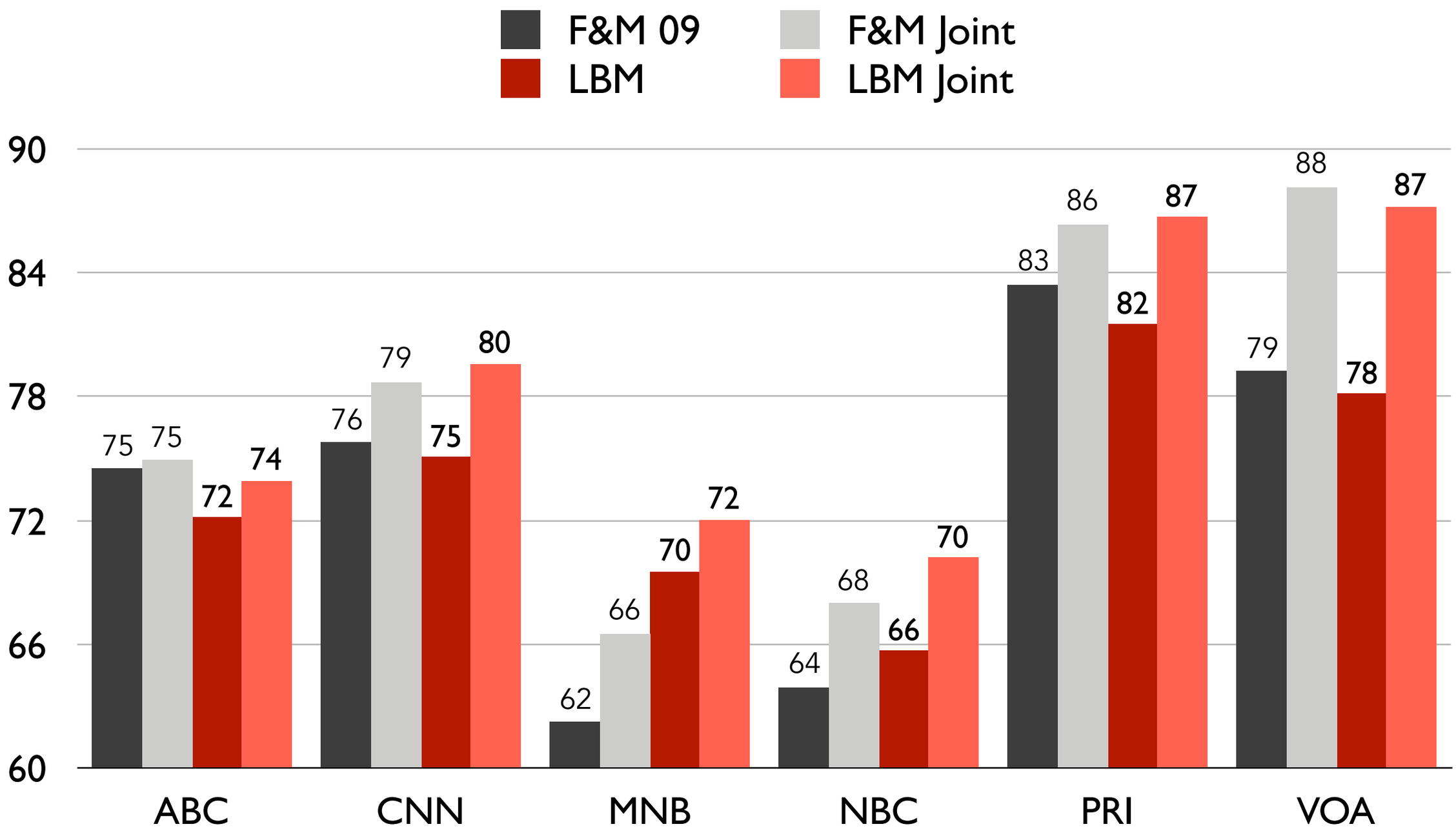
Coordination Factors

- Special-purpose logical factors used for connecting components of different models
- Implement a soft NAND function
  - fire when all connected variables are true
  - Learn a set of coordinating feature weights

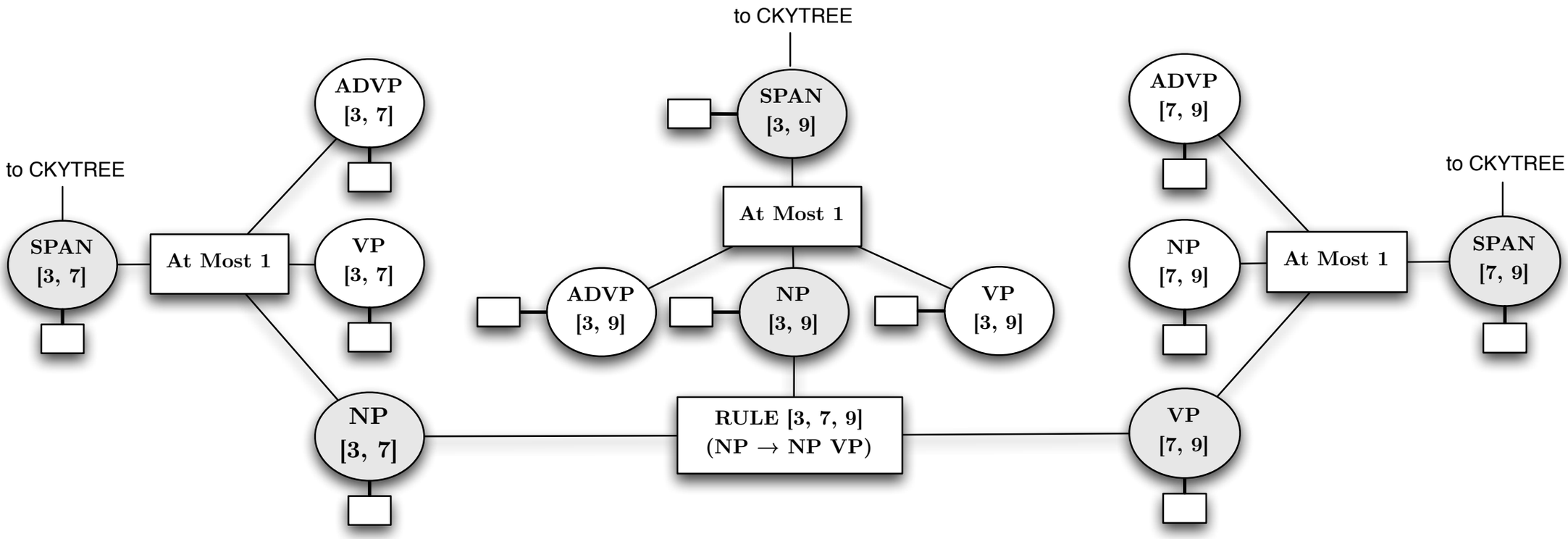
NER Results (F1)



NER Results - Joint Model (F1)

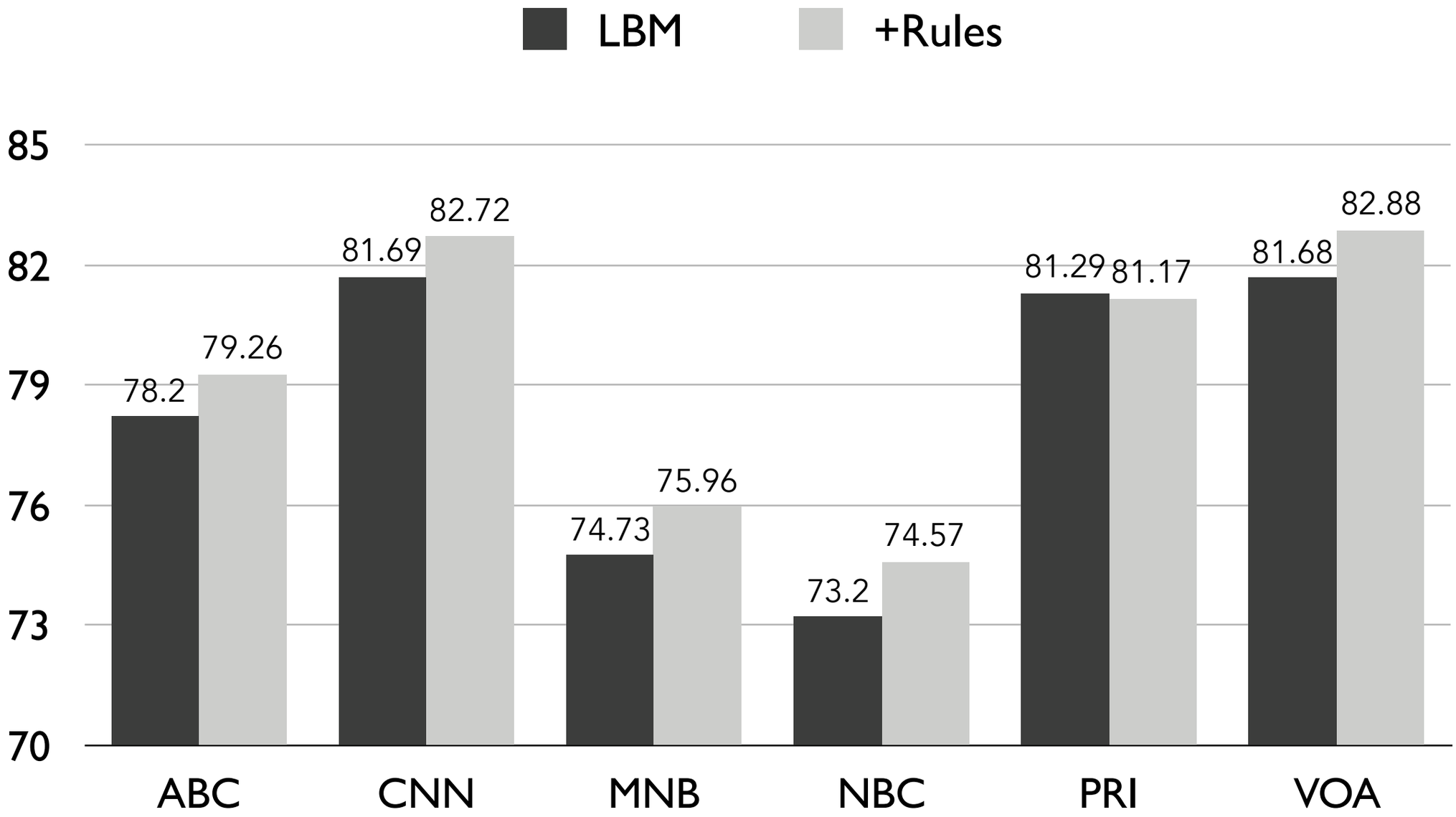


Addendum: Parsing with Rules



- Rule complexity is additive: $O(ln^3 + |G|)$
- Rules are *sparse*: only correct predictions of grammarless model, do not specify a full derivation
- Grammar constructed via perceptron updates on top of the labeled prediction model

Rule Parsing Results



Conclusions

- Joint inference again improves performance on syntax and NER, yielding best-reported results on OntoNotes data set
- A New Parser Decomposition for Factor Graphs
  - Constrain variables [strong local information] to form valid parses with combinatorial factors [strong global information]
  - More efficient than grammar-based parsing
  - Relies on logical factors to connect components of different models in flexible ways
  - Extensible for joint inference to other factor graph models and other NLP tasks