Joint Inference for SUBTLE
FACTORIE & PragBot

Tiger Wu
Department of Computer Science
University of Massachusetts Amherst

Joint work with Andrew McCallum, David Smith, Jason Narad, Karl Schultz, Sameer Singh, Sebastian Riedel.
Inference in Language Interpretation

• Many sources of evidence
  – Utterance
  – Lexical resources
  – Discourse context
  – Perceptual and cognitive states (uncertain!)

• Many interacting interpretation rules
  – Syntactic, semantic, pragmatic ambiguities

• Find most likely interpretation by *joint inference* over evidence and interpretation rules
Key Scientific Challenges

**Intractable inference**
- Many interactions in large model, and between many components of interpretation
- Very large search space
- Cannot be explored exhaustively

**Building the interpretation rules & weights**
- Not practical to build by hand
- *Learn* how to score interpretations from corpus
Intractable inference

- Solve large joint problem by divide-and-conquer, coordinating components by dual decomposition [EMNLP].
- Focused inference by query-aware MCMC [NIPS].
- Scalable inference by parallel computation [ACL].

Building the interpretation rules & weights

- SampleRank learner beats Contrastive Divergence [Hinton], and has new proof of convergence [ICML].
- Lightly-supervised, active learning with Generalized Expectation Constraints lowers supervision cost [CIKM].
“Factor Graphs, Imperative, Extensible”

Implemented as a library in Scala [Martin Odersky]
- object oriented & functional
- type inference
- runs in JVM (complete interoperation with Java)
- fast, JIT compiled, but also cmd-line interpreter

Library, not new “little language”
- integrate data pre-processing & eval. w/ model spec
- leverage OO-design: modularity, encapsulation, inheritance

Scalable
- billions of variables, super-exp #factors, DB back-end
- fast parameter estimation through SampleRank [2009]
FACTORIE Progress This Year

• Design: more flexible definitions
  - *Model* now an arbitrary source of factors
  - *Factor* now no longer tied to Templates
  - *Values* representation enables parallel inference
  - *Domain* now no longer tied to class name
  - *Statistics & Values* now subclasses of *Assignment*

• New belief propagation implementation;
  including research in parallel, multi-core BP, building on Joshi’s “Bidirectional
  Incremental Construction”

• Parameter estimation flexibility, not only SampleRank, but also exact L-BFGS.

• Undirected & directed (generative) models combinable

• Many new inference procedures, e.g. collapsed variational Bayes.

• Demonstrated on 200 million variables
Pragbot I Set-up

- A maze with scattered cards;
- Two human players, each able to hold 3 cards;
- Goal: collect 6 consecutive cards in the same suit;
- Each player has limited number of moves;
- The players can chat with one another to form a strategy.
- Sections of the maze may be blocked off from players.
- Not every game is winnable, but players must find out.
Overall Objective - NLP

- Study how people use language to coordinate a cooperative task;
- How do people select useful information in a dialog (relevance)?
- What speech act do players perform (imperative, interrogative or declarative)?
- What ambiguity do the pragmatic contexts resolve?
Domain Specifics

- The pragbot world is limited
- Entities include players, cards and the maze
- Objectives are well defined
- Simple modeling of the game possible
- NLP can be integrated in a game model
- Interesting testbed for reasoning under uncertainty, ambiguous language and pragmatics
Measuring progress

- At any instance of the game, we can use the cards the players are holding to predict the final outcome as well as how close the players are to their goal.
- As a simple measure, the probability of a winning hand is the product of the probability of each card. Deficient probability, but sufficient when normalized for entropy calculation.
- Entropy can be calculated with these probabilities. The closer the game is to a solution, the lower the entropy.
Sample Entropy

Entropy (nats) vs. Actions

Actions 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Entropy (nats) 4 3 2 1 0.5 0
Sample Entropy

Player1: “I have 10S and JS”

Player1 picks up another 2S, Player1 holding 9D from before
Ignoring player1, player2 picks up 4D, and then another 8D.

Player2: “I have 4d, 8d and 9d.”
At this moment, Player1 drops 2S...

And picks up QS.
Player1: “wanna go for S? I have J Q 10”
Player2: “I will see what I find”
Sample Entropy

Player1: “let's just do S because I already got 3 consecutive.”

Player2 drops 4D and picks up KS
Sample Entropy

Player2 drops 8D and picks up 2S. Player2: “I found 2S. 10,J,Q,K,A,2?”
Player1: “OK”
Entropy indicates strategy

There?
Let's gather all the ards in our area
Let's see if we can...
Yes we have one
I'm in middle
C ya

Actions

Entropy (nats)
More examples
Entroy as an evaluation

- Since entropy are calculated from winning hand probabilities:
- Different prediction model possible by changing how individual card probabilities are calculated.
- More confident prediction of the winning hand (peaked probability distribution) lowers entropy.
- The earlier we can predict the outcome, the lower the entropy integrated over time.
- Can be used to compare different game models.
From a single player’s perspective:

- Goal and Hand influence what is talked about;
- Hand and Question under Discussion influence the guess of partner’s hand;
- Question under Discussion is revealed by actual utterance that is observed;
- Question under Discussion, Guess of Partner’s hand, and Goal all influenced by previous time steps.
Progress so far – Chain Model

- Annotated the dialogues with card disambiguation
  
  i have 5[s], 7[s], 8[s], 9[s], 10s
  I now have 5s 6s and 10s
  do you have 7[s],8[s],9[s] in your hand?
  do you have s's
  i have 5[s], 7[s], 9[s], 10[s]

- Currently investigating an inference procedure that will work well with SetVariables to be added to the Factorie library.
Chain model

Halved, simplified version of the complete model

- The Goal variable is omitted, player strategies not modeled
- Question under Discussion primitively implemented as a bag of entities, thus does not differ from the Utterance variable
- Overall, an HMM like chain model
Parallel Work: Unsupervised Adaptation for NLP

- Using neural networks (MLP) to learn a fixed-length vector representation for each lexical type [Collobert & Weston 08]
- Constructing embedding with dependency trees
- Jointly learning to parse and to embed
- Integration of neural model into Pragbot
- Dimensionality reduction for better semantics representation; alleviates out-of-domain problems
Road Map - Pragmatics

- Goal: separate Question-under-Discussion from Utterance – finer modeling of the discourse
- Annotating the data with speech-acts
- Reasoning with Gricean Maxims
- Scale up inference to full graphical model with hidden goals and QUDs