# Beyond myopic inference in Big Data pipelines

Karthik Raman, Adith Swaminathan, Thorsten Joachims, Johannes Gehrke

Cornell University, Ithaca, NY USA

## Introduction

- **Setting:** Big Data pipelines constructed using modular components
- **Problem:** Error by a component cascades through the pipeline causing catastrophic failure in the eventual output
- **Key idea:** Establish correspondence between pipelines and ***Probabilistic Graphical Models*** that explains pipeline operation theoretically
- **Result:** More robust inference procedures while still using existing components

## An illustrative example: A NLP pipeline

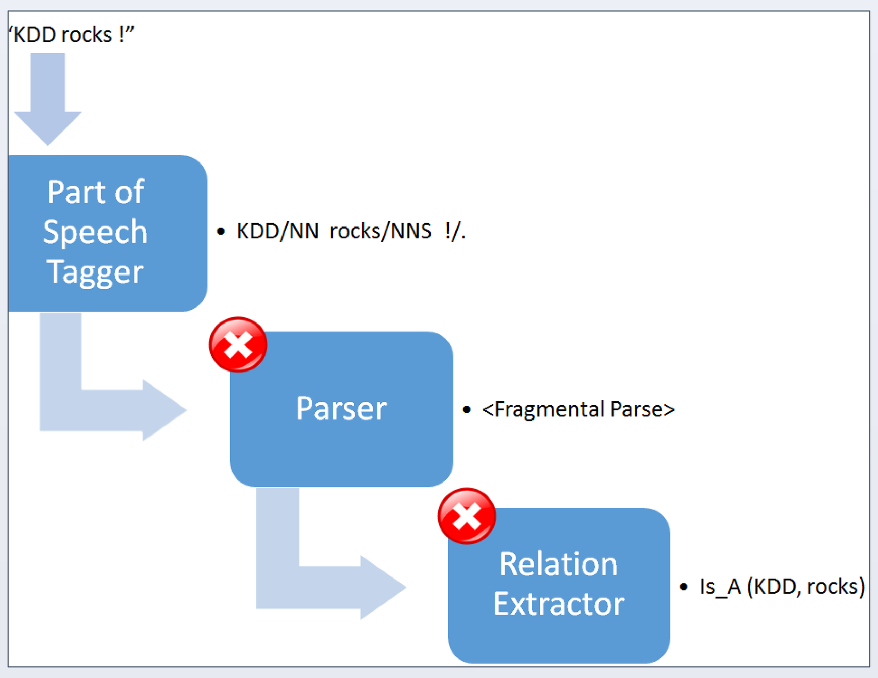



**Figure 1.** Tagger tags "rocks" incorrectly, causing an unrecoverable failure

- Using locally optimal component output is myopic
- **Want:** Globally better outputs

> - Error detection needs a notion of confidence scores for predictions.
> - Error recovery needs a mechanism for alternative predictions

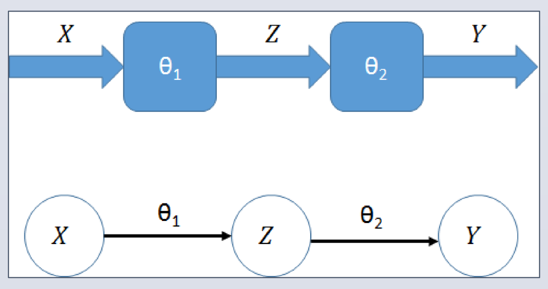
## Approach

View components as probabilistic models - regardless of their actual implementation.

- Component models $\Pr(y|x, \theta)$. For input $x$, it returns $y^* = argmax_y \Pr(y|x, \theta)$
- Confidence score = $\Pr(y^*|x, \theta)$
- When using dynamic programming to maximize, maintain and return list of $k$ top scoring outputs $[y^1, \ldots, y^k]$
- Composition of probabilistic components → a directed graphical model

**Figure 2.** Inputs/outputs of components become nodes

- Components are edges in graphical model



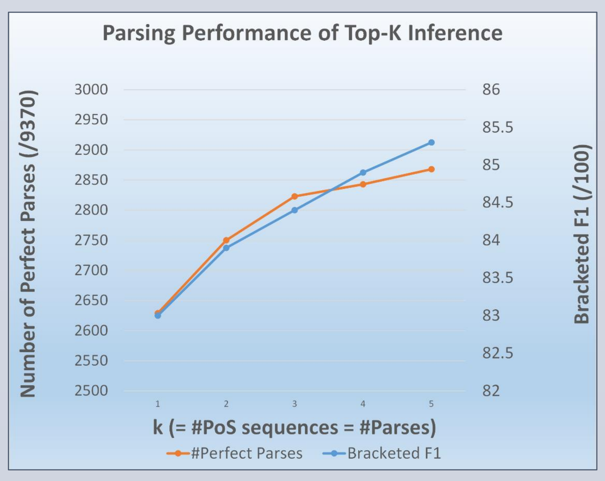

- Ideal inference in a graphical model with observed variable $X$:

$$y^* = argmax_y \sum_z \Pr(y|z, \theta_2) . \Pr(z|x, \theta_1)$$

- *Canonical inference* computes
$z^* = argmax_z \Pr(z|x, \theta_1)$ ; $y^* = argmax_y \Pr(y|z^*, \theta_2) . \Pr(z^*|x, \theta_1)$
- … a greedy approximation!
- With a list of $k$ top intermediates $\{z\} = [z^1, \ldots, z^k]$ a better approximation is *Top-K Inference* :

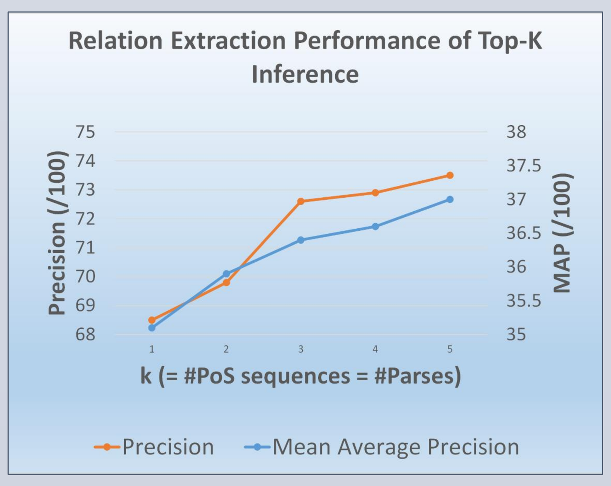
$$y^* = argmax_y \sum_{z \in \{z\}} \Pr(y|z, \theta_2) . \Pr(z|x, \theta_1)$$

## Does *Top-K* actually help?




← **Figure 3.** Parsing

**Figure 4.** → Relation extraction

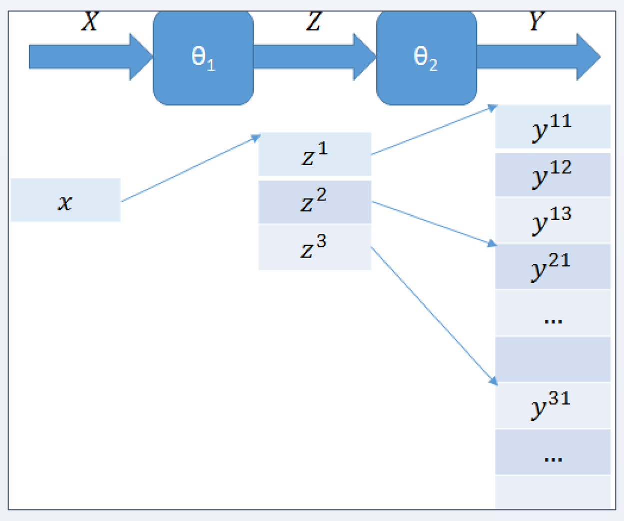



- Using more outputs better than canonical inference
- **Parsing:** Two stage pipeline, evaluated on WSJ benchmark
- **Relation extraction:** Three stage non-linear pipeline, evaluated on *difficult* subset of ACE-04 newswire benchmark

## Efficient inference : Beam and Adaptive inference

**Figure 5.** Top-$k$ inference causes multiplicative blowup of inference cost

- **Observation:** Diminishing returns from more values
- **Idea:** Use beam search to limit list lengths
- Given budget $m * k$, retain top $m$ after each stage



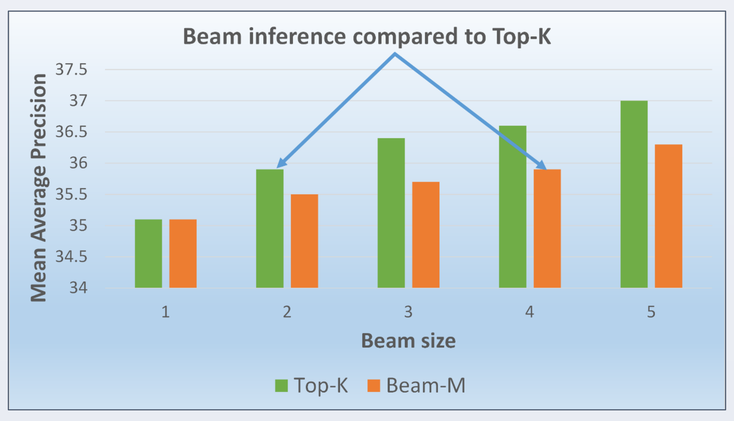




**Figure 6.** Smooth performance improvement like top-$k$ inference

- With linear increase in inference cost (in beam size)

- For robust inference, ideal #outputs required from each component will vary for different inputs
- Unlike Top-$k$ and *Beam*, *Adaptive inference* exploits this
- Effect of an output on overall prediction is estimated first
- Propagate iff it has a large effect

> Create scored list $[z^1, \ldots, z^k]$. If $Score(z^i) > \tau.Score(z^{i+1})$, return $[z^1, \ldots, z^i]$.

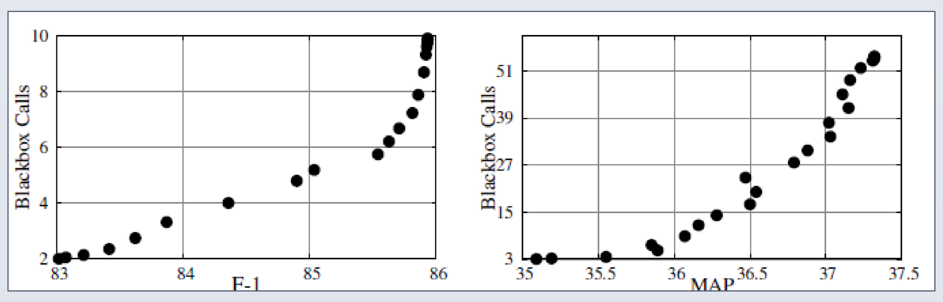



**Figure 7.** Increasing threshold τ smoothly increases overall accuracy and cost

## Discussion

- *Top-K*, *Beam* and *Adaptive Inference* are generic algorithms
  - No assumptions about components' error models, or the pipeline structure.

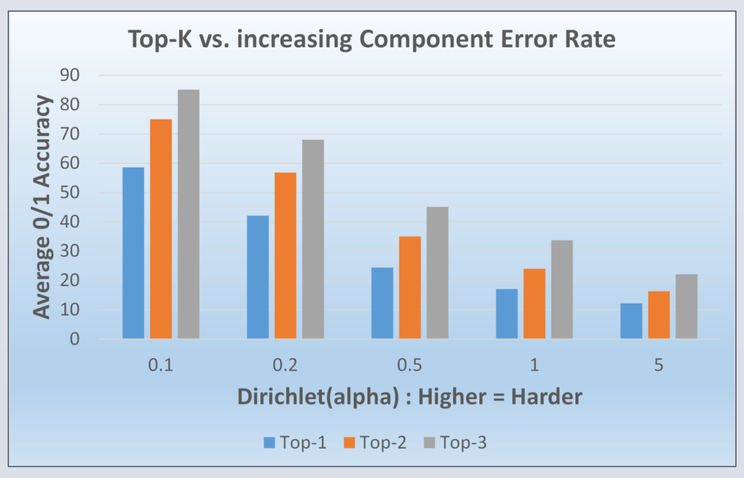



**Figure 8.** Synthetic pipeline with 3 components.

- Components model $\Pr(y|x, \theta)$ with a $Dirichlet(\alpha)$ distribution
- As task becomes harder ($\alpha$ increases), Top-$k$ remains robust

- Graphical model view of pipelines viable even with components that aren't probabilistic models
  - Calibrated optimization criterion → surrogate for $\Pr(y|x, \theta)$
  - Redundant components can be used to get "top-$k$" outputs
- Components make two kinds of errors:
  - "Near miss": When the correct output is in the top-$k$ list for small $k$
  - Catastrophic: Cannot recover cheaply even using *Top-K Inference*
- This work suggests a novel objective to train components by minimizing the number of catastrophic errors they make.

## Conclusion and Future Work

- Canonical inference with myopic components cause unrecoverable pipeline errors
- Viewing pipelines as graphical models allows reasoning about overall inference
- Proposed different inference procedures to approximate ideal inference problem
- Experiments demonstrate robust pipelines constructed using existing components
- Handling pipelines with feedback
- Incorporating uncertainty of predictions into training

## Contact

The full paper is available for personal use at
http://www.cs.cornell.edu/~adith/Papers/PipelineInference.pdf

For more information, please e-mail: adith@cs.cornell.edu