Confidence Modeling for Neural Semantic Parsing

July 16, 2018

Li Dong*, Chris Quirk^, Mirella Lapata*

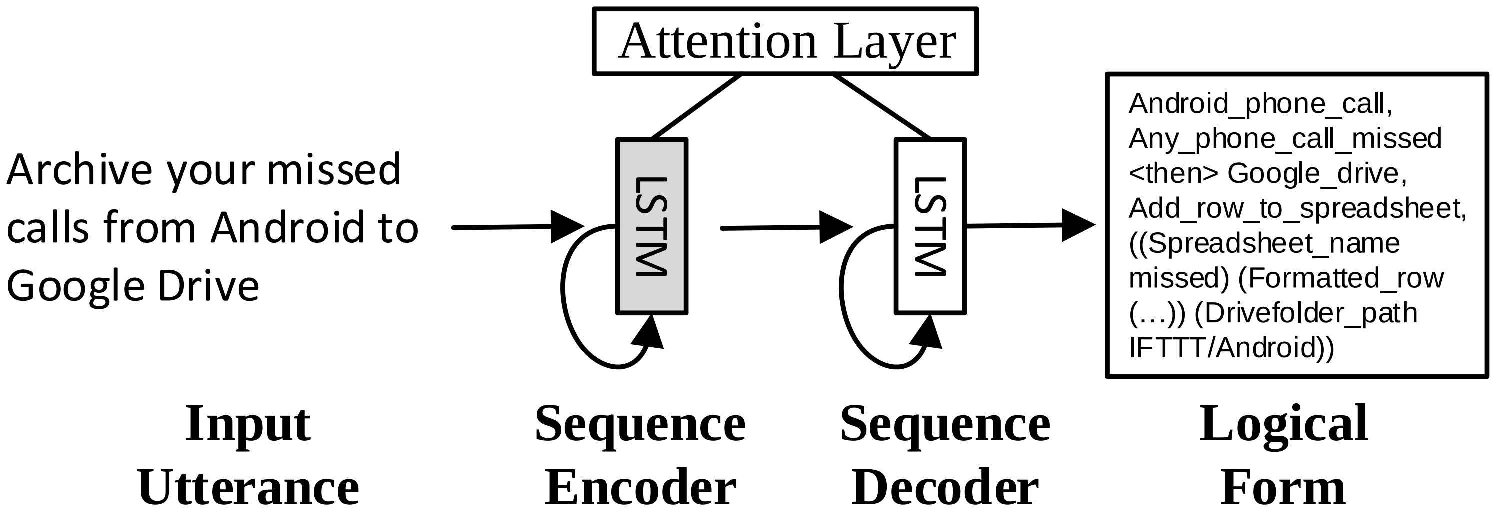
*: University of Edinburgh

^: Microsoft Research, Redmond

Neural Semantic Parsing (NSP)

Model used in this work

(Dong and Lapata, 2016; Jia and Liang, 2016)



Confidence Modeling is Important

Most models always tend to guess some outputs

We also want to know how confident they are

Alexa, buy me something from Whole Foods

Buying Whole Foods

http://time.com/4821460/amazon-buying-whole-foods-twitter/

Motivation

- From the perspective of applications
  - More reliable decisions
  - Generate clarification questions to verify the results
- Nonlinearity of neural networks
  - For linear models, $p(y|x) \propto \sum score_{evidence}$
  - Unclear for neural models (Johansen and Socher, 2017)
- Lack of explicit lexicons or templates

  Difficult to trace errors and inconsistencies

Research Goal

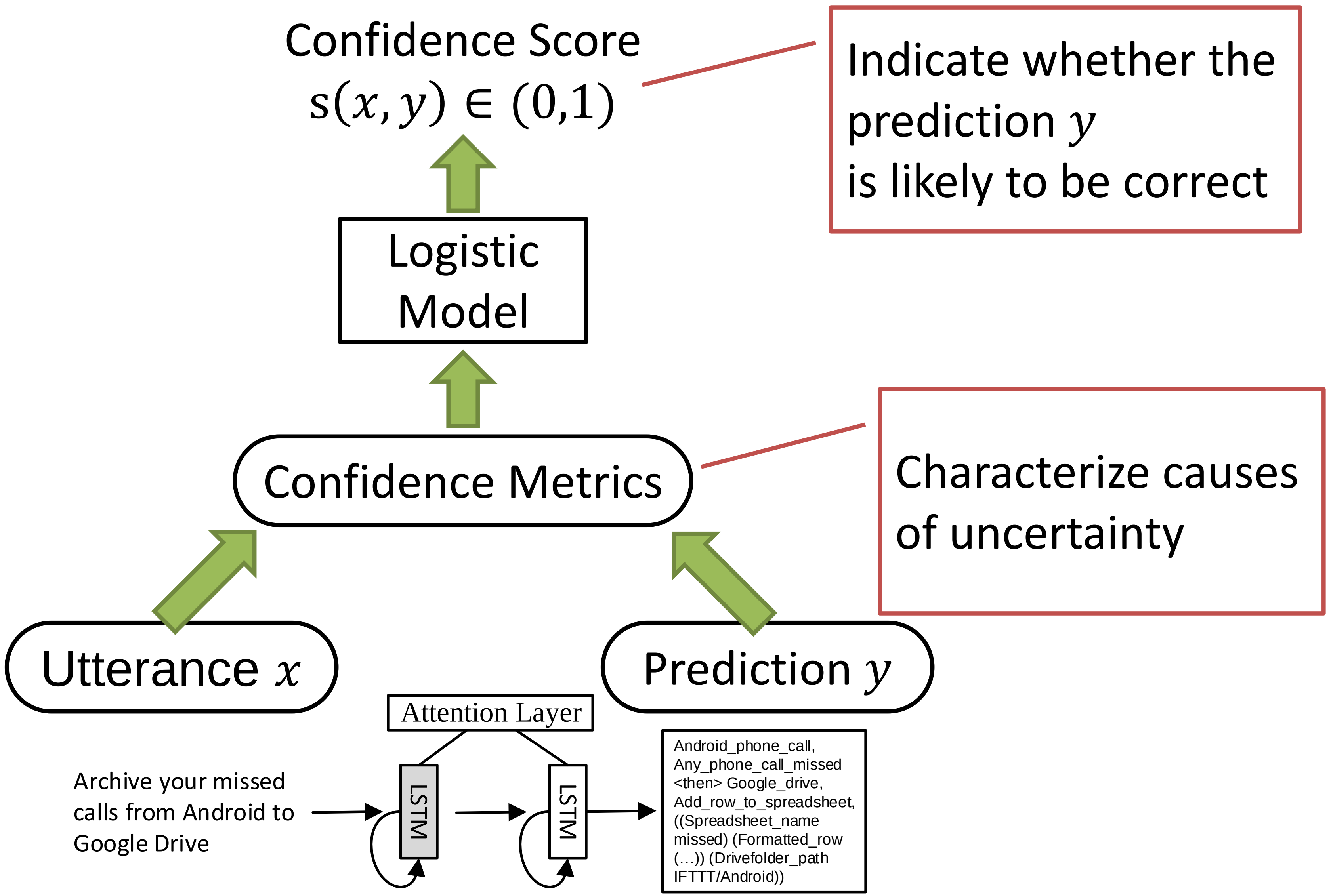
- Estimate confidence scores for NSP

  Higher score -> the prediction is more likely correct

- Provide uncertainty interpretations

  Which parts of input contribute to uncertain predictions

Confidence Estimation - Overview



Confidence Metrics

- Model is unconfident about $p(y|x)$
  - Model uncertainty

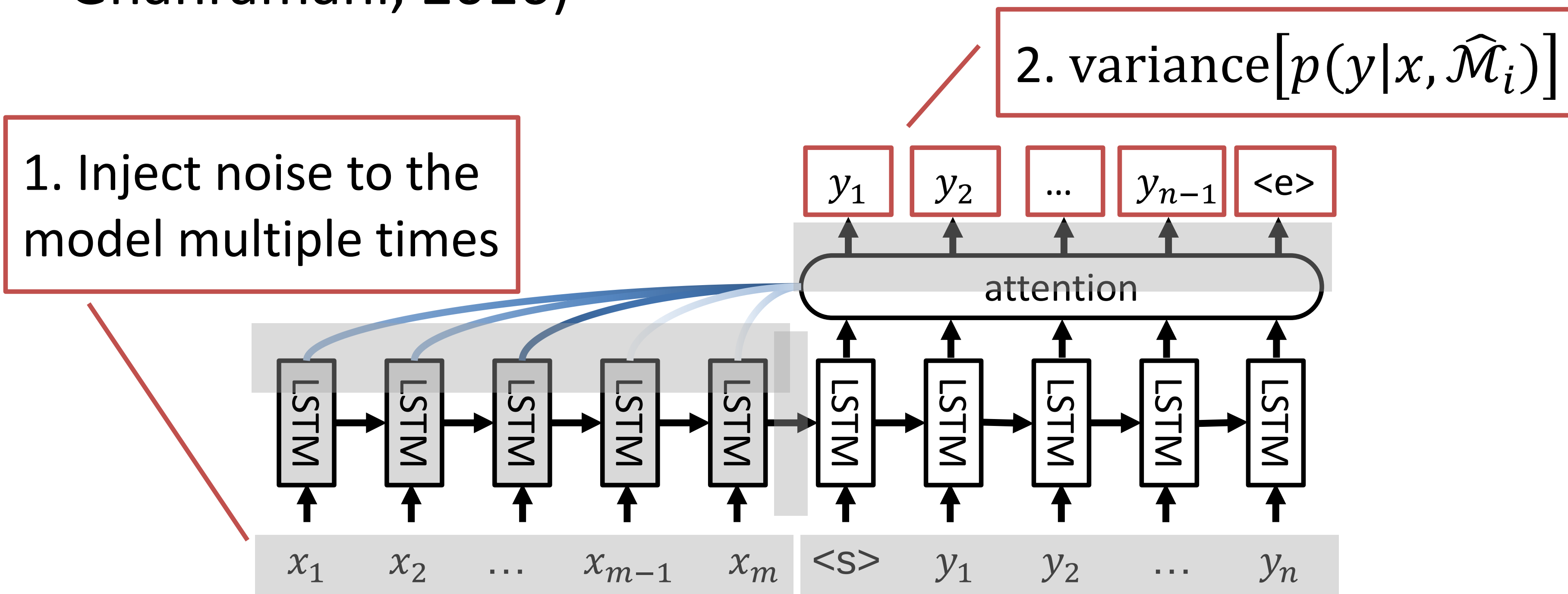
    Unsure about model parameters or structure
  - Data uncertainty

    Out-of-distribution/-domain examples
- Estimate $p(y|x)$ reliably, but the entropy is large
  - Input uncertainty

    Input itself is unspecific/ambiguous, which would lead to several different correct outputs

Model Uncertainty

- Posterior probability
  - Sequence-level: $\log p(y|x)$
  - Token-level: $\text{avg}\{\log p(y_t|x, y_{<t})\}$, $\min\{p(y_t|x, y_{<t})\}$
- Dropout as a Bayesian approximation (Yarin Gal, Zoubin Ghahramani, 2016)



Data Uncertainty

Out-of-distribution/-domain examples

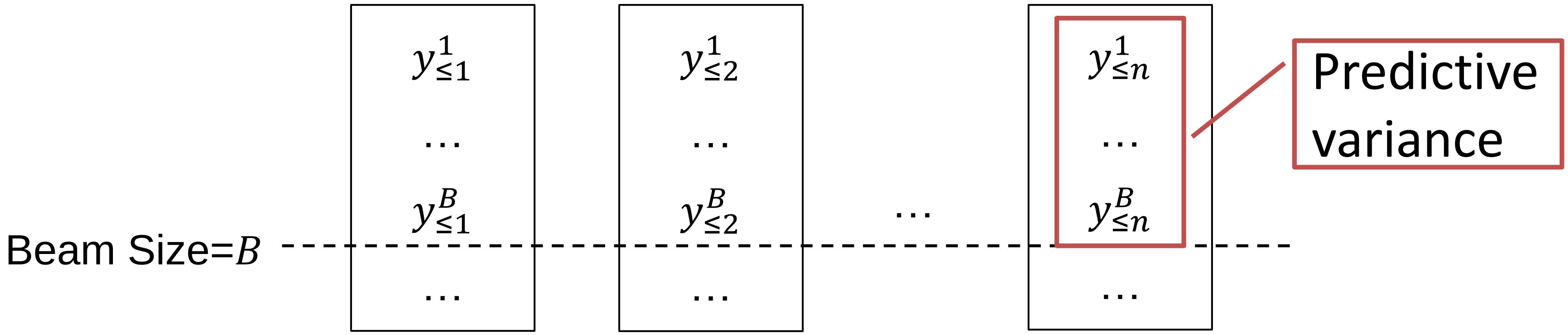
- $p(x|\mathcal{D})$: probability of input

  KenLM (Heafield et al., 2013) estimated on the training set

- Number of unknown words of input

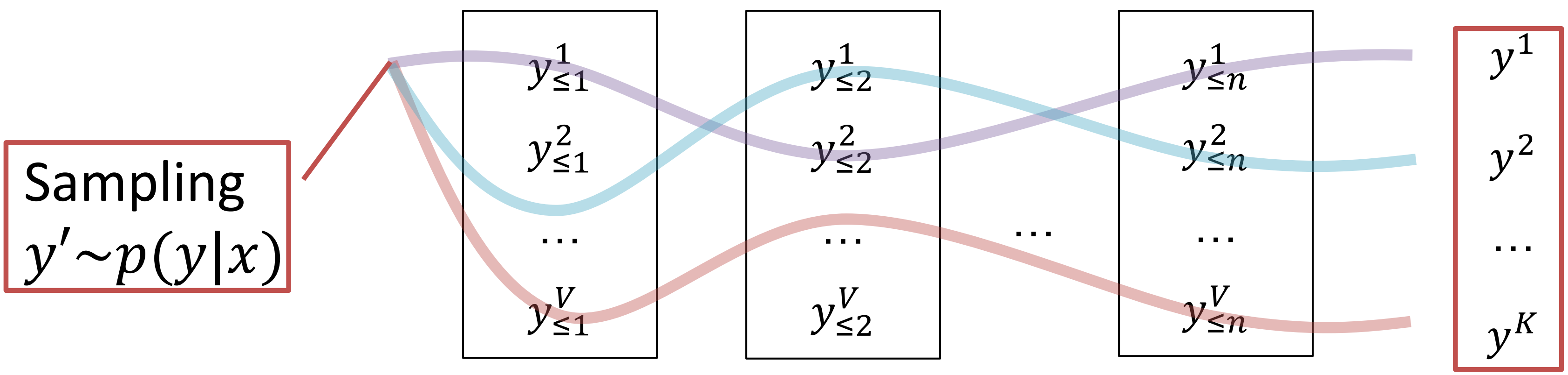
Input Uncertainty

- Variance of top candidates $\mathrm{var}\left[p\left(y^i \middle| x\right)\right]$



- Entropy of decoding $H[y|x] = -\sum_{y'} p(y'|x) \log p(y'|x)$

  Approximated by Monte Carlo sampling



Confidence Scoring

Use logistic regression to fit F1 scores of outputs

Logistic loss: $\mathcal{L} = \sum_i [y_i \ln(1 + e^{-\hat{y}_i}) + (1 - y_i) \ln(1 + e^{\hat{y}_i})]$

Confidence Score $\in (0,1)$

↑

Tree Boosting Model

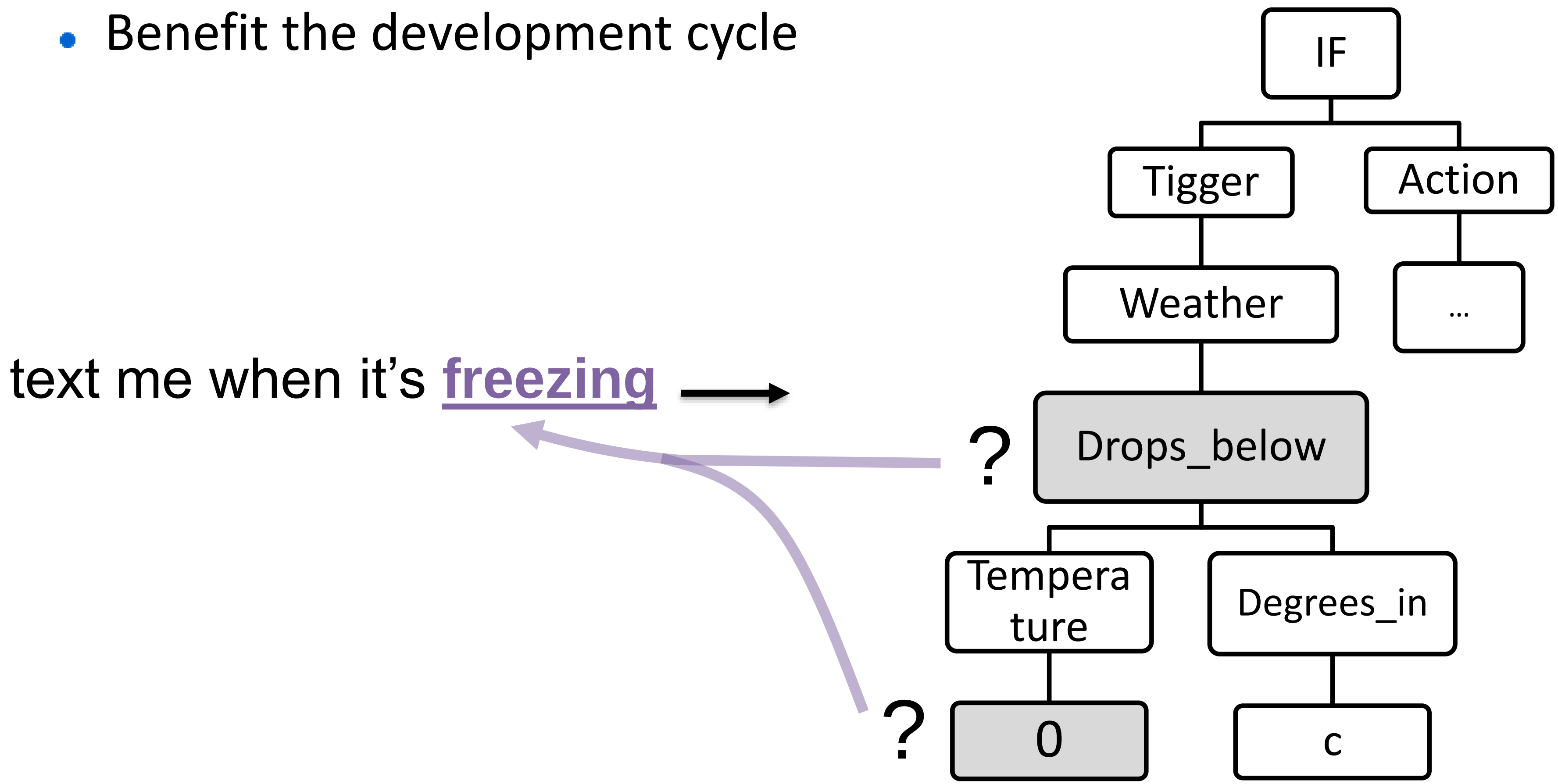
Confidence Metrics

	Model Uncertainty	**Data Uncertainty**	**Input Uncertainty**
Token-level	• Dropout perturbation • Gaussian noise • Posterior probability	• Probability of input • Number of unknown tokens	• Variance of top candidates • Entropy of decoding
Sequence-level			

Uncertainty Interpretation

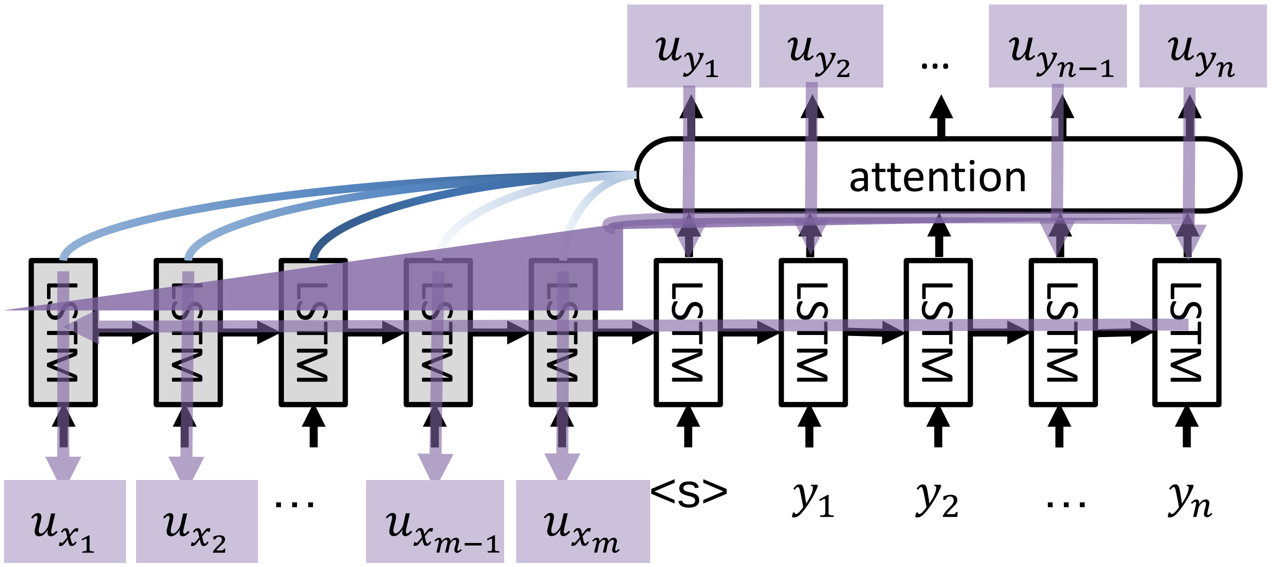
Trace prediction uncertainty back to input words

- Users can verify or refine the input quickly
- Benefit the development cycle



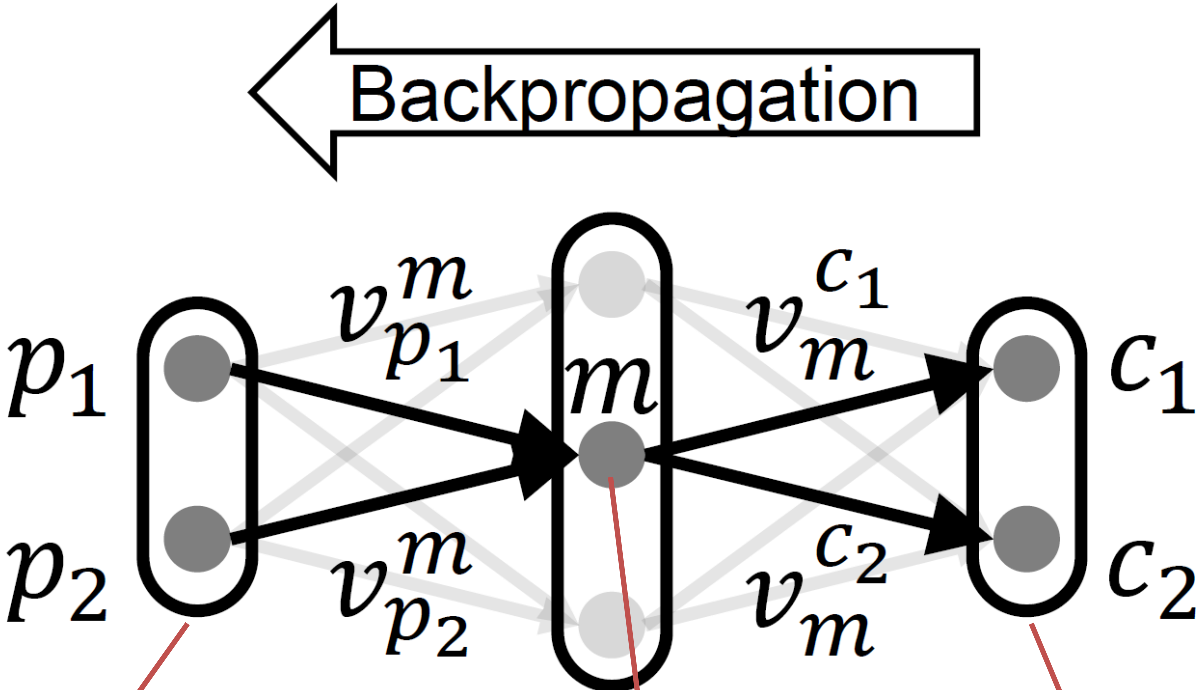
Uncertainty Backpropagation

1) Initialize decoder's output neuron with uncertainty scores
2) Backpropagate scores layer-wisely
3) Obtain scores u_{x_i} for input words



(Bach et al., 2015; Zhang et al., 2016)

Uncertainty Backpropagation



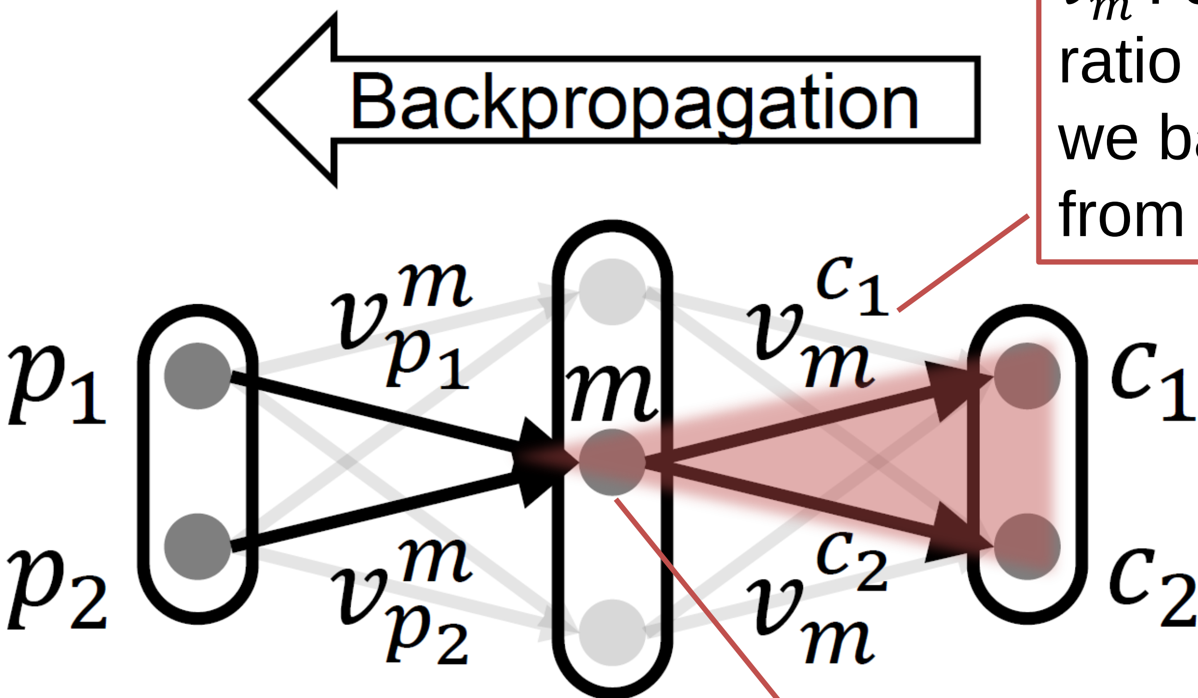
$\text{Parent}(m) = \{p_1, p_2\}$

u_m: uncertainty score

$\text{Child}(m) = \{c_1, c_2\}$

Uncertainty Backpropagation

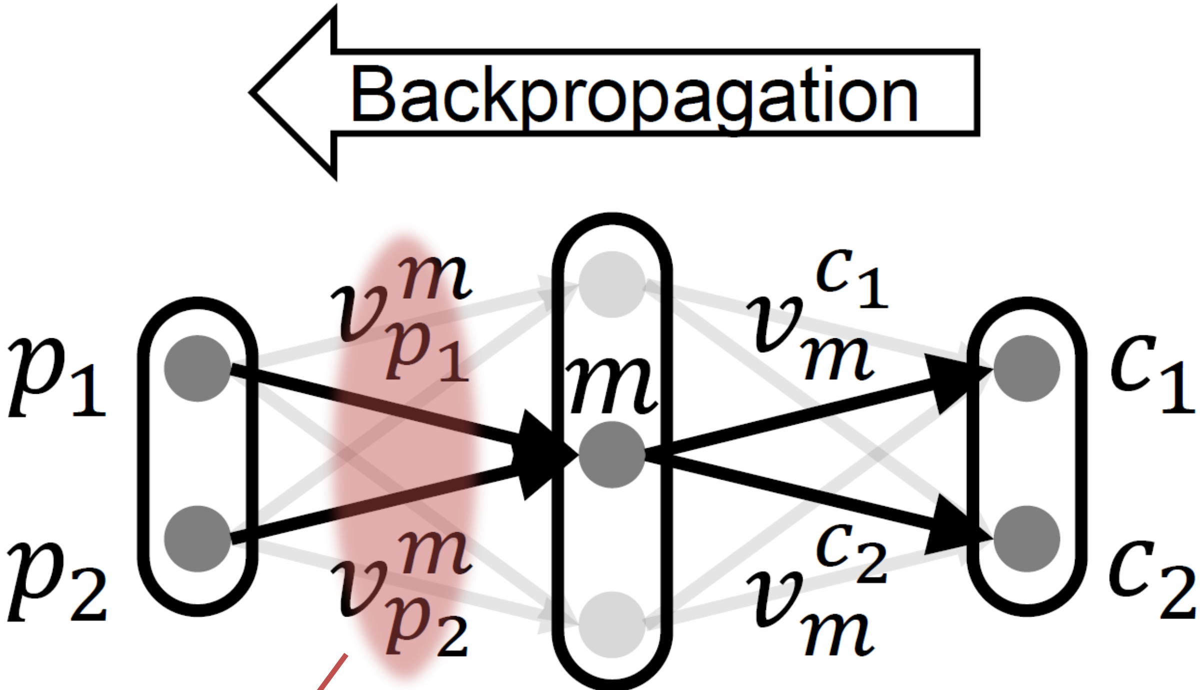
$v_m^{c_1}$: contribution ratio (how much we backprop from c_1 to m)



Scores are backpropagated from child neurons

$$u_m = \sum_{c \in \mathrm{Child}(m)} v_m^c u_c$$

Uncertainty Backpropagation



$$\sum_{p \in \mathrm{Parent}(m)} v_p^m = 1$$

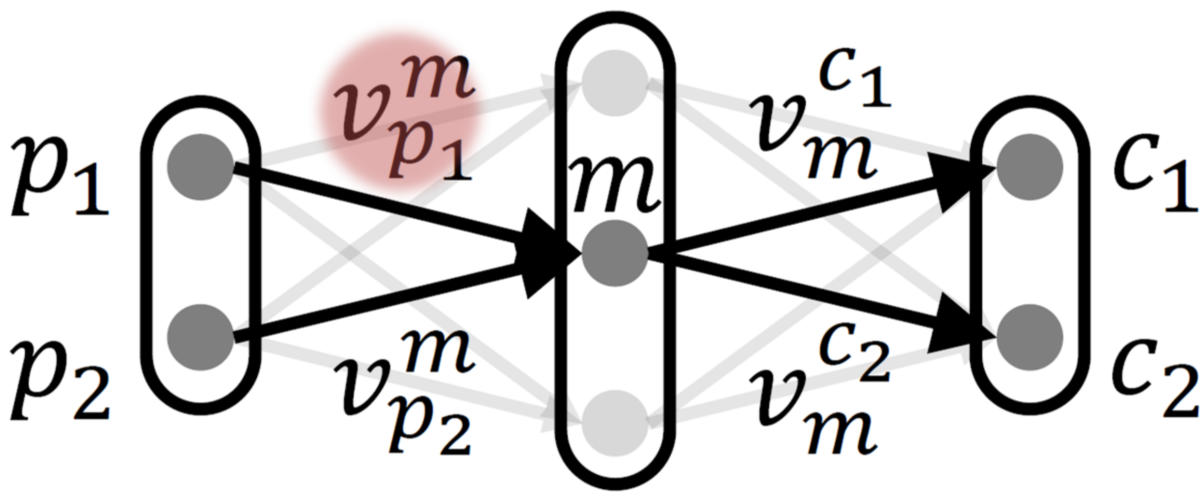
Contribution ratios from m to its parent neurons are normalized to 1

Backpropagation Rules

Fully-connected layers

If p_1 contributes more to m's value, ratio $v_{p_1}^{m}$ should be larger (i.e., backprop more from m to p_1)

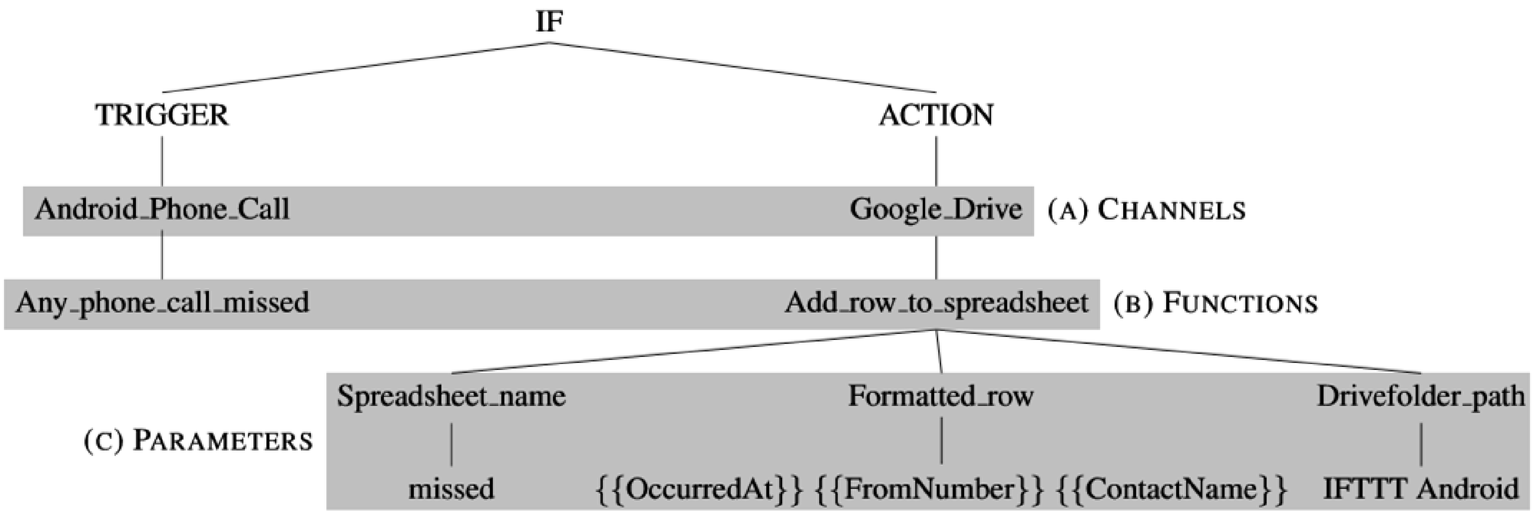
$$m = \sigma(W_1 p_1 + W_2 p_2) \quad \Rightarrow \quad v_{p_1}^{m} = \frac{|W_1 p_1|}{|W_1 p_1| + |W_1 p_1|}$$



Experiments

- IFTTT-style semantic parsing (Quirk et al., 2015)

  "Archive your missed calls from Android to Google Drive"



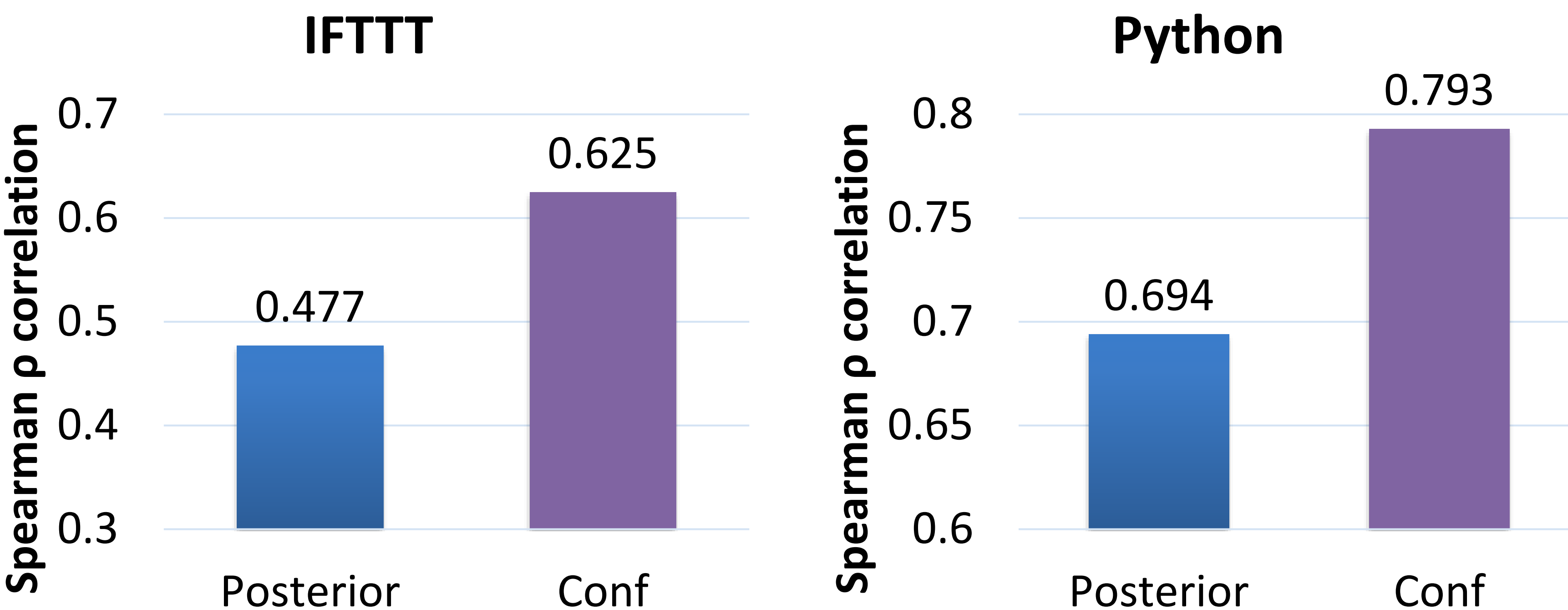
- Python code generation (Yin et al., 2017)

for every key in sorted list of user_settings

```
for key in sorted(user_settings):
```

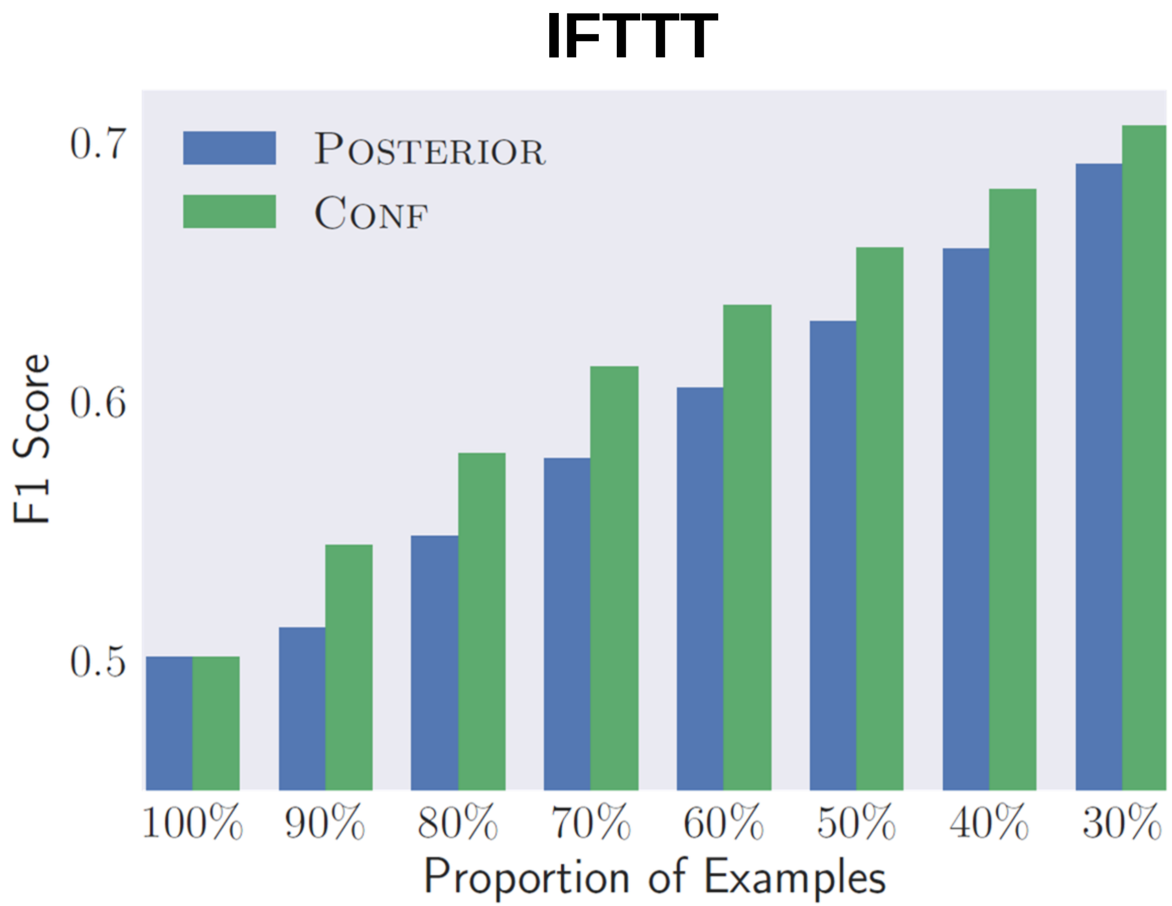
Confidence Estimation

Spearman ρ correlation ($\in [-1,1]$) between confidence score and F1 score

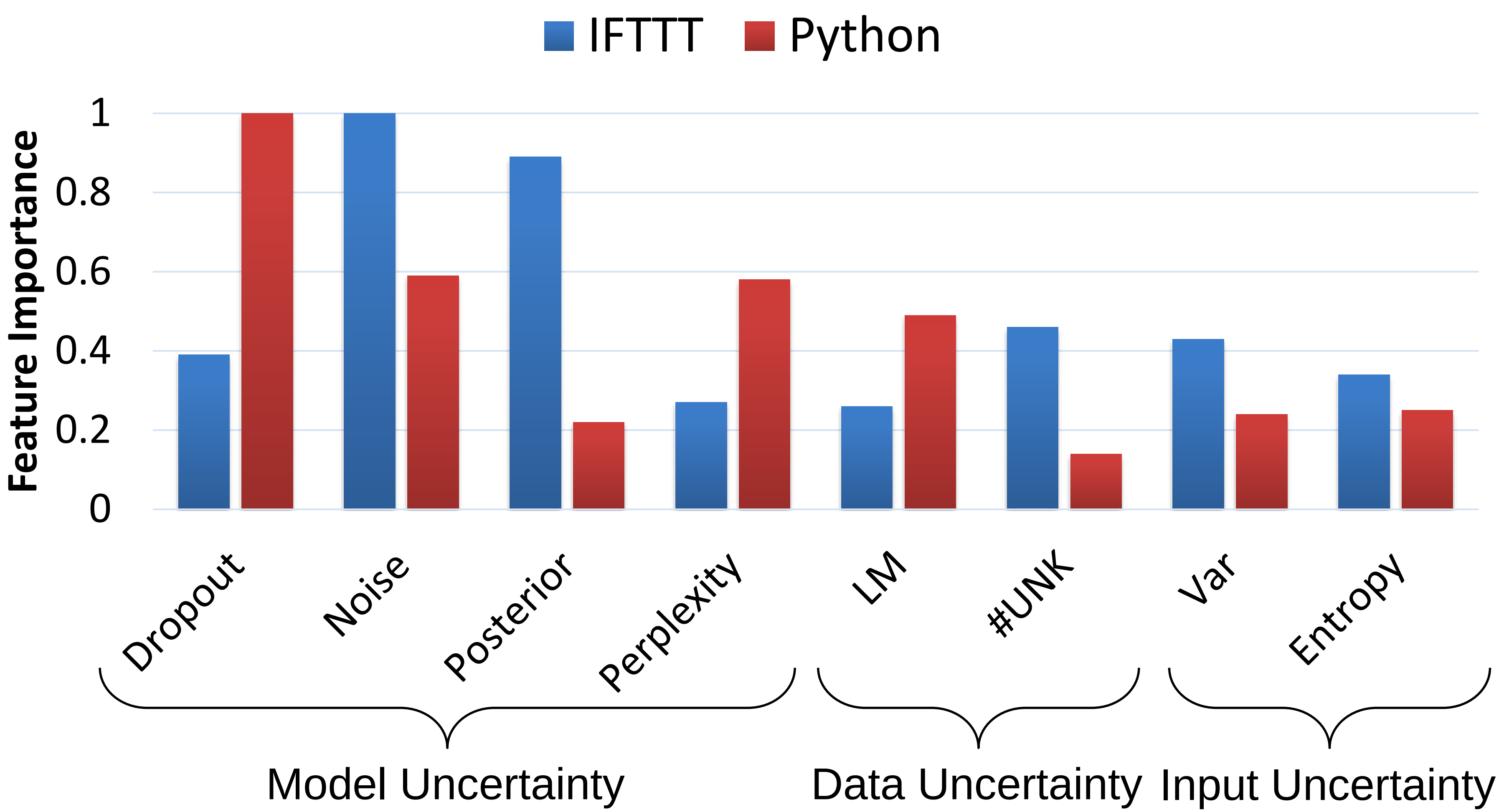


Confidence Estimation

Confidence scores are used as threshold to filter out uncertain examples



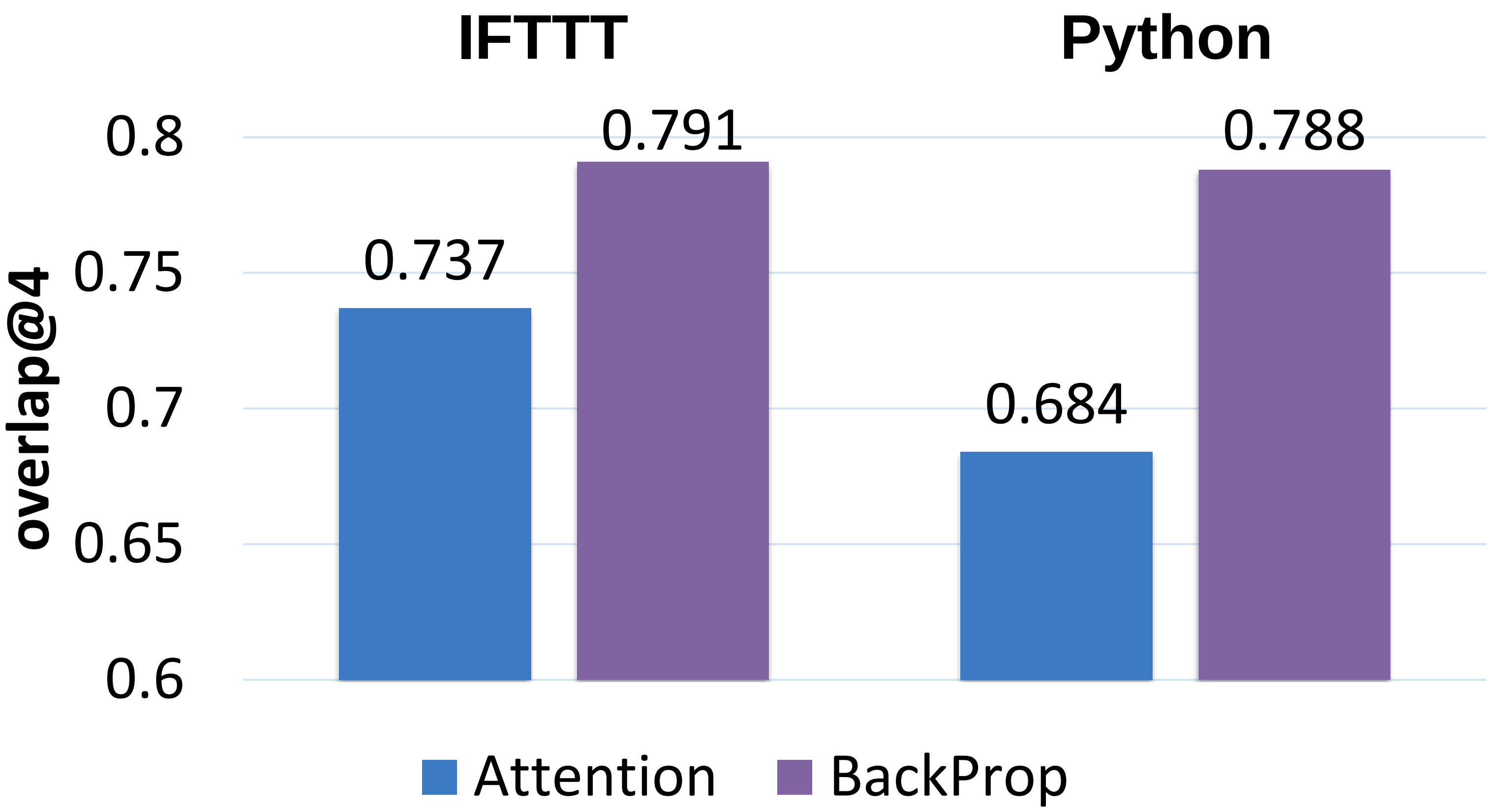
Importance of Confidence Metrics



Uncertainty Interpretation

Agreement of top-4 uncertain input words

Between model prediction and gold standard



Examples - IFTTT

ATT: attention; BP: uncertainty backpropagation

`google_calendar—any_event_starts THEN facebook —create_a_status_message—(status_message ({description}))`	
ATT	post calendar event to facebook
BP	post calendar event to facebook
`feed—new_feed_item—(feed_url( _url_sports.espn.go.com)) THEN ...`	
ATT	espn mlb headline to readability
BP	espn mlb headline to readability
`weather—tomorrow's_low_drops_below—(( temperature(0)) (degrees_in(c))) THEN ...`	
ATT	warn me when it's going to be freezing tomorrow
BP	warn me when it's going to be freezing tomorrow

Thanks!

Q&A

Code Available:
http://homepages.inf.ed.ac.uk/s1478528