

---

---

# Flexible Answer Typing with Discriminative Preference Ranking

Christopher Pinchak<sup>1</sup>, Dekang  
Lin<sup>2</sup>, and Davood Rafiei<sup>1</sup>

<sup>1</sup>University of Alberta

<sup>2</sup>Google, Inc.



---

---

# Quick Summary

- Apply discriminative preference ranking to the problem of answer typing
  - Focus on appropriate rather than correct
- Rank candidate answers because:
  - Unknown number of appropriate candidates
  - Ranked list is the desired outcome
- Experiments show improvement over alternative models
  - Examine focused what/which questions that are context-rich
  - High performance from typing alone ( $> 0.5$  MRR)

---

---

# The Problem

- Given a question and set of potential answers, decide which potential answers are plausible as responses
  - Question may have many such plausible responses
  - Question may have many correct answers
- Candidate answers come from a high-quality source
  - Most are related to the question in some way

---

---

# Potential Solutions

- Use question classification as in Li & Roth (2002)
  - Requires a predefined set of types
  - Requires named-entity recognition to identify appropriate candidates
- Use the probabilistic model of Pinchak & Lin (2006)
  - No predefined set of types
  - Brittle model → difficult to add new features

---

---

# Our Solution

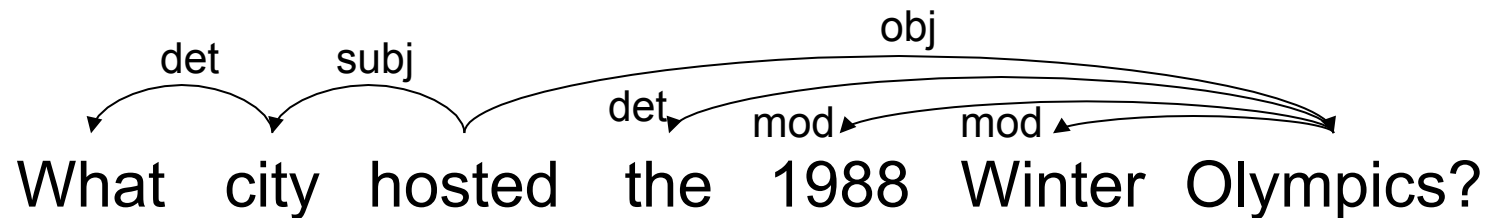
- Change the Pinchak & Lin (2006) model into a discriminative preference ranking model
- Why discriminative?
  - Add more features (specifically a candidate frequency feature)
  - Automatically weight different contexts
- Why preference ranking?
  - Unbalanced set of appropriate/inappropriate
  - Most candidates are high quality
  - Ultimately want a ranked list of candidates

---

---

# Model Resources

- Question and question contexts



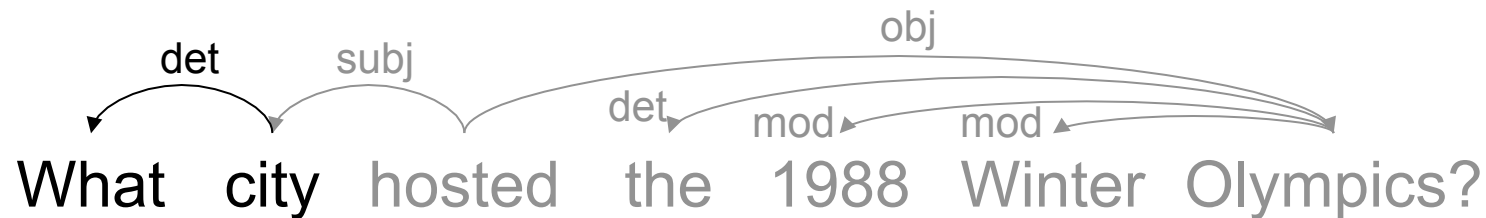
- Parsed corpus to find context fillers
- Word clusters for unseen words

---

---

# Model Resources

- Question and question contexts



X ←<sup>subj</sup> city

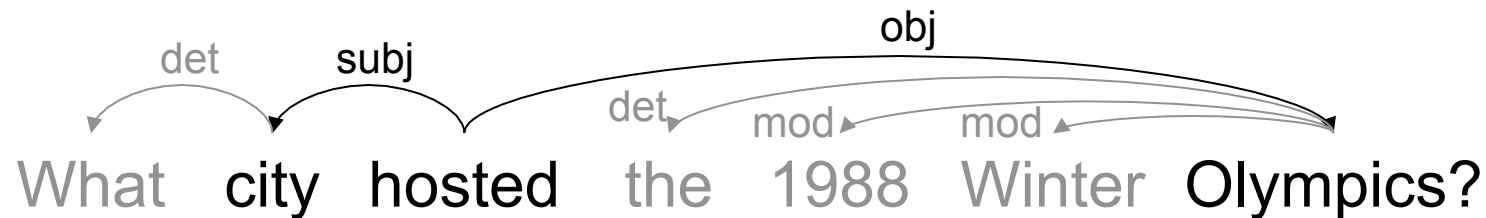
- Parsed corpus to find context fillers
- Word clusters for unseen words

---

---

# Model Resources

- Question and question contexts



X ←<sup>subj</sup> city

X ←<sup>subj</sup> hosted <sup>obj</sup>→ Olympics?

- Parsed corpus to find context fillers
- Word clusters for unseen words

---

---

# Feature Templates

- $E(t,c)$ : expected count
  - $E(\text{Calgary}, X \leftarrow \text{subj} \leftarrow \text{hosted} \rightarrow \text{obj} \rightarrow \text{Olympics}) = 7.2$
  - Computed as:  $E(t,c) = \sum_{\chi} \Pr(\chi | t) C(\chi,c)$
- $C(t,c)$ : actual count
  - $C(\text{Calgary}, X \leftarrow \text{subj} \leftarrow \text{hosted} \rightarrow \text{obj} \rightarrow \text{Olympics}) = 1$
- $\sum_{t'} C(t',c)$ : context count
  - $C(*, X \leftarrow \text{subj} \leftarrow \text{hosted} \rightarrow \text{obj} \rightarrow \text{Olympics}) = 50$
- $\sum_{c'} C(t,c')$ : candidate corpus count
  - $C(\text{Calgary}, *) = 125$
- $S(t)$ : candidate list count
  - $S(\text{Calgary}) = 10$

---

---

# The Role of Preference Ranking

- For ranking  $c_i <_r c_k$  we want  $\mathbf{w} \cdot \Phi(c_i) > \mathbf{w} \cdot \Phi(c_k)$ 
  - Use SVM<sup>light</sup> such that  $\mathbf{w} \cdot (\Phi(c_i) - \Phi(c_k)) \geq 1 - \xi_{i,k}$
  - Finds a minimum  $\mathbf{w}$  just like for a SVM classifier
  - Known as a rank constraint
- Rank constraints are only created for  $i,k$  pairs under  $r$ 
  - Can assign equivalent ranks
- For all questions  $q$ , with appropriate candidates  $a_{qi}$  and inappropriate candidates  $b_{qk}$ 
  - $r = \{a_{qi} <_r b_{qk}\}$

---

---

# Experimental Framework

- Training data:
  - 9-fold cross validation on 385 focused what/which questions from TREC 2002-2006
  - Train on correct vs incorrect answers
- Testing data
  - Two annotators labeled chunks from top 20 snippets
  - ~140 candidates per question, 80-90% inappropriate
  - Rank noun chunks in top 20 Google snippets
  - Look at both correct and labeled appropriate chunks
    - MRR on correct
    - MRR and precision/recall on appropriate

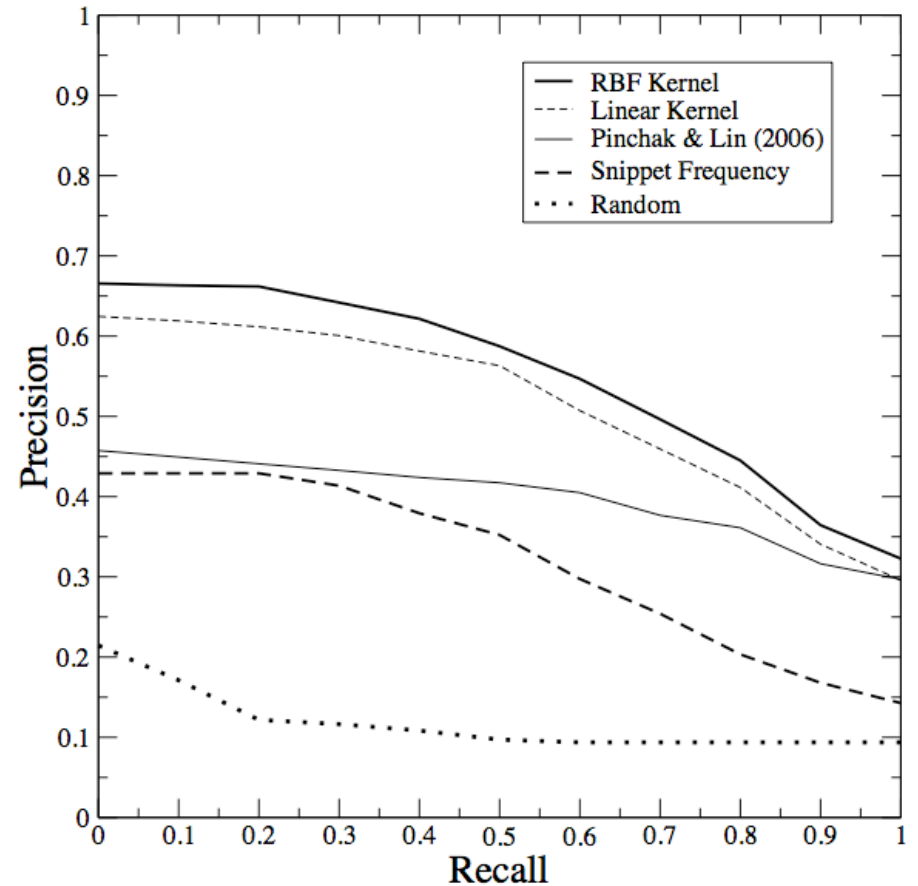
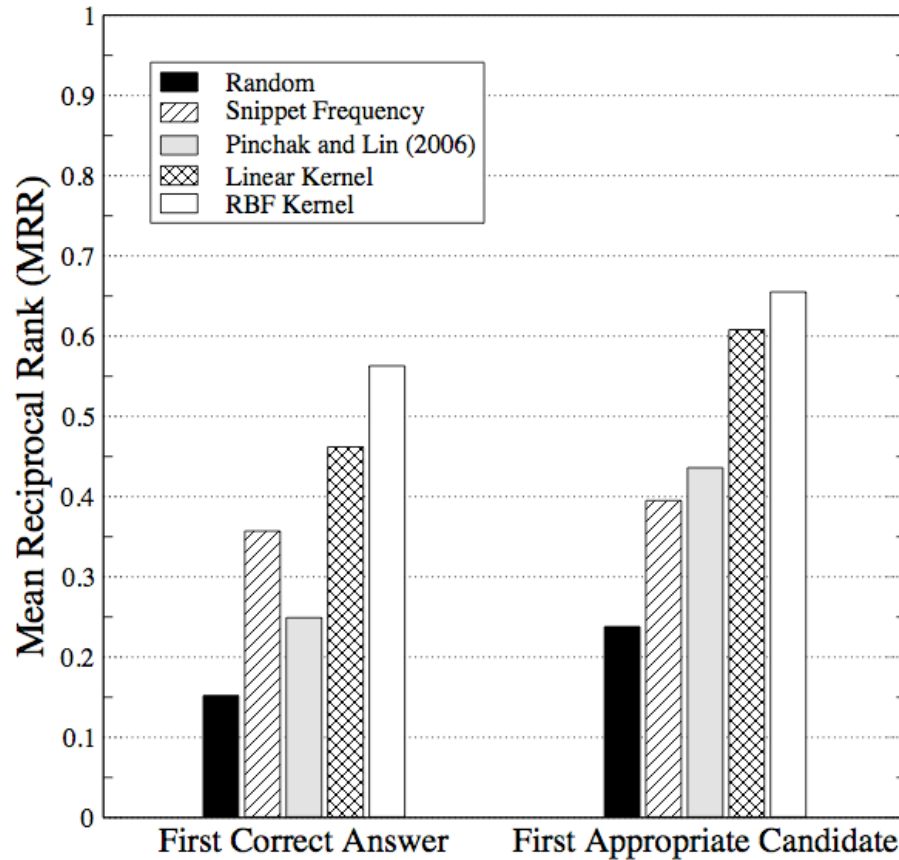
---

---

# Experiment: Comparing with Alternative Methods

- Five systems to compare
  - Random: randomly sort the list 100 times and take the average
  - Snippet Frequency: order by how often we see the candidate in the list
  - Pinchak & Lin (2006): precursor to this work
  - Linear Kernel: model built with a linear kernel
  - RBF Kernel: model built with a non-linear RBF kernel

# Experiment Results



---

---

# Key Points

- Regardless of kernel, discriminative preference ranking performs better
  - Pinchak & Lin (2006) and Snippet Frequency models also perform well
- Model achieves over 0.5 MRR
  - No additional answer extraction is used
- No need for labeled data
  - Correct answers from TREC results

---

---

# Review

- Discriminative preference ranking model
  - No need to deal with balancing the set of examples
  - More natural fit for the problem
  - Not typical fixed-set typing (i.e., question classification)
- Experiments support improvements in typing accuracy
  - High performance on its own
  - Trained on correct and applied to appropriate

---

---

# Future work

- Applied the same idea to how-adjective questions (not in the paper)
  - Extended the simple model of Pinchak & Bergsma (2007)
  - Allows the use of many features
- Working on applying a similar model to information retrieval results
  - Hopefully an upcoming paper

- 
- 
- Thank you for attending!
  - Questions?
  
  - Thanks to:

