End-to-End Learning of Parsing Models for Information Retrieval

Microsoft

Jennifer Gillenwater Xiaodong He Jianfeng Gao Li Deng

ICASSP 2013

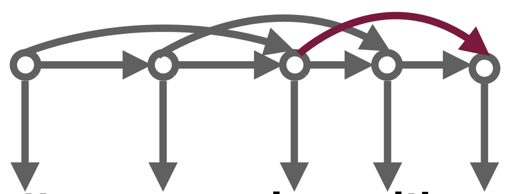
jengi@cis.upenn.edu, {xiaohe,jfgao,deng}@microsoft.com

EXAMPLE TASK: WEB SEARCH

Focus: Long queries (5+ words)

PRIOR WORK

Non-parsing approaches (e.g. [1]):
Fail to exploit *long-range* dependencies.



Intended meaning: "rock" = "tool used by predator"

Inferred meaning: "rock" = "pearl", because of its close proximity to "clam"

Syntactic parsing approaches (e.g. [2]):
Fail to exploit *meaningful* dependencies.

Issue #1: Queries lacking verbs, prepositions, punctuation, etc. are incorrectly parsed.

Ex: "electrical" should have parent "fire"

query: electrical fire causes precautions safety

Issue #2: Even correct parses fail to link terms whose interaction is *meaningful* for query disambiguation.

Ex: "rock" needs a more direct link with "otter"

query: otter opens clam with rock

N-gram-based parsing approaches (e.g. [3]):
Parses linking frequently co-occurring words are better, but don't exploit the available direct *supervision*.

***Supervision*: Human-annotated relevance scores (between 0 and 4) for many document-query pairs.**

query: otter opens clam with rock

scores	titles of (possibly) relevant documents
4	Sea otter breaks open mollusk against a rock
3	Wild otters and their use of rocks as tools
2	Facts about the giant otter of the Amazon river
1	Clams camouflaged on a rocky river bottom
0	You otter investigate this really great website

Our main contribution:
Principled method for learning a parser based on information retrieval (IR) supervision.

MEASURING SUCCESS IN IR

The higher a relevant document appears on a list of search results for a given query, the larger the NDCG.

$$\text{NDCG}@L = \frac{1}{Z}\sum_{i=1}^{L}\frac{2^{v_i}-1}{\log_2(1+i)}$$

v_i = relevance of the i^{th} document to the query

Z = normalization constant s.t. NDCG@L = 1 for a perfect ranking of the top L documents

PARSING MODEL

$T_{\mathbf{w}}$

otter opens clam with rock

w_1 w_2 w_3 w_4 w_5

$$p_\theta(T_{\mathbf{w}}) = \prod_{w_i \to w_j \in T_{\mathbf{w}}} \theta_{w_j|w_i}$$

$$\theta = \text{parser parameters}$$

Goal: Use IR supervision to learn θ that maximize NDCG.

TREE EDIT DISTANCE RANKER

There are many ways to use the dependencies of a query parse to rank documents. In this work, we use tree edit distance (TED).

$$f(\mathbf{q},\mathbf{d}) = \text{substitution} + \text{deletion} + \text{insertion}$$

q: otter opens clam with rock

d: Sea otter breaks open mollusk against a rock

Example costs:

sub: $\frac{\theta_{\text{clam|otter}}+\theta_{\text{mollusk|otter}}}{\text{sim(clam,mollusk)}}$

del: $\theta_{\text{with|rock}}$

ins: 0 (free)

SMOOTH NDCG-BASED OBJECTIVE

NDCG is non-smooth, so we follow recent work [4] in defining a related but smooth objective to optimize.

The logistic loss for query k on documents h and s, where h is more relevant than s, is:

$$C_{k,h,s} = \log\left(1+\exp\left[f(\mathbf{q^{(k)}},\mathbf{d^{(h)}}) - f(\mathbf{q^{(k)}},\mathbf{d^{(s)}})\right]\right)$$

Full objective:

$$\min_\theta = \sum_{k=1}^{|Q|}\sum_{h=1}^{|D^{(k)}|}\sum_{s=h+1}^{|D^{(k)}|} C_{k,h,s}$$

$$\text{s.t. the } \theta \text{ are in the probability simplex}$$

Optimization: Gradient descent on the Lagrangian dual.

TRAINING ALGORITHM

Note that for additional correlation with NDCG, as in [5], gradients are scaled by the NDCG gain of swapping documents:

$$\gamma = \frac{2^{v_{r_{k,h}}}-2^{v_{r_{k,s}}}}{Z}\left(\frac{1}{\log(1+r_{k,h})}-\frac{1}{\log(1+r_{k,s})}\right)$$

```
1 Initialize θ randomly
2 while objective gradient is significant do
3     Parse each w ∈ Q ∪ D: argmax_{T_w} p_θ(T_w)
4     foreach q ∈ Q, d ∈ D^q do
5         Compute tree edit distance f(q, d)
6     end
7     Update θ according to γ-scaled gradients
8 end
```

TRAINING IN PRACTICE

Despite the non-convexity introduced by line 3 in the above algorithm, in practice optimization quickly converges.

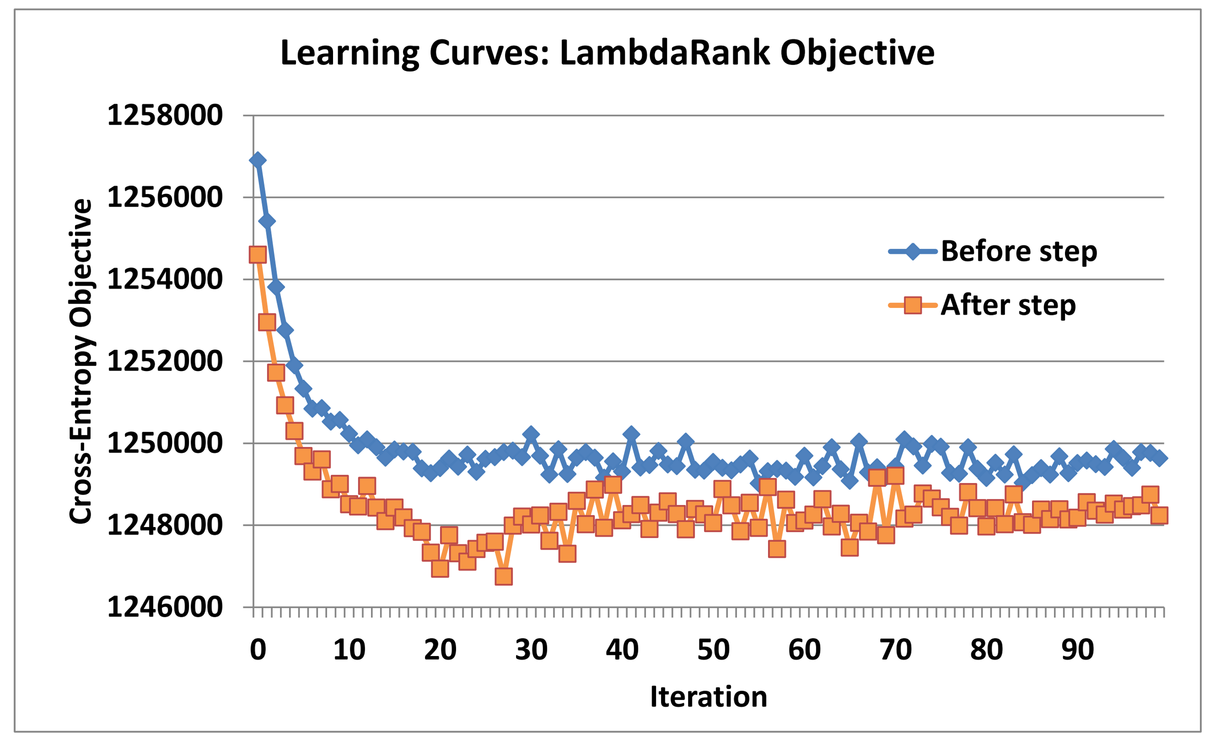


Figure: Objective value just before updating the parameters (before line 7 in the above algorithm) and after updating.

RESULTS FOR NDCG@10

Baseline (ML): instead of directly optimizing NDCG, the baseline uses the Viterbi Expectation-Maximization algorithm to maximize the likelihood of the parse trees.

Query length	# of queries	ML trained	Our method	Absolute improvement
5	211	32.16	32.27	0.11
6	92	30.05	30.33	0.28
7	51	27.69	28.20	0.51†
≥ 8	56	24.52	25.18	0.66†

Superscript † indicates statistical significance ($p < 0.05$).

(1) D. Metzler and B. Croft. "Latent concept expansion using Markov random fields." SIGIR, 2007.
(2) V. Punuakanok et al. "Natural language inference via dependency tree mapping: An application to question answering." Computational Linguistics, 2004.
(3) R. Nallapati and J. Allan. "Capturing term dependencies using a language model based on sentence trees." CIKM, 2002.
(4) C. Burges et al. "Learning to rank using gradient descent." ICML, 2005.
(5) C. Burges et al. "Learning to rank with non-smooth cost functions." NIPS, 2006.