Learning Dense Models of Query Similarity from User Click Logs

Fabio De Bona, Stefan Riezler*, Keith Hall,
Massi Ciaramita, Amac Herdagdelen, Maria Holmqvist

Google Research, Zürich
*Dept. of Computational Linguistics, University of Heidelberg

Query Rewriting in Large Scale Web Search

- Problem:
  - Web search: Term mismatch between user queries and web docs.

    Users describe their information need by a few keywords, which are likely to be different from the index terms of the web documents.
  - Sponsored search / Ads: Additional difficulty of matching queries against very few, very short documents.
- Task: Conjunctive term matching needs to be relaxed by
  - rewriting query terms into new terms with similar statistical properties (generative models for query expansion),
  - ranking candidate rewrites w.r.t. criteria such as click-through-rate or semantic similarity (discriminative models for rewrite ranking).

Discriminative Models for Rewrite Ranking

- Rewrite candidates from different sources need to be filtered according to criteria such as click-through-rate or semantic similarity

= Learning-to-Rank problem: Learn ranking of query rewrites from data that are ranked according to measures of interest.

- Task(s):
  - Create training data (by sampling from user logs) and test data (by manual labeling a subsample).
  - Feature engineering, incorporating complex models of string similarity as dense features.
  - Find most robust learner, i.e., learner that performs best under various evaluation metrics on clean test data when trained on noisy training set.

Extracting Weak Labels from Co-click Data

- Training data extraction:
  - Assume two queries to be related if they lead to certain amount of user clicks on the same retrieval results (cf. Fitzpatrick & Dent (1997)'s model of query similarity based on the intersection of retrieval results).
  - Threshold of >= 10 co-clicks suffices to find query-pairs that are considered similar by human judges.
  - Data set of > 1 billion query-rewrite pairs extracted for experiments.
- Test data labeling:
  - 100 queries with 30 rewrites, sampled in descending order of co-clicks.
  - Labeling in two steps: Rank rewrites using GUI, then (re)assign rank labels and binary relevance score (see Rubenstein & Goodenough (1965)).

Data Statistics

	Train	Dev	Test
Number of queries	250,000	2,500	100
Average number of rewrites per query	4,500	4,500	30
Percentage positive rewrites per query	0.2	0.2	43

- Train, Dev, and Test sets are sampled from same user logs data.
- Different percentage of relevant documents per query.
- Co-click threshold of 10 just sufficient for significant correlation between human relevance judgments and automatic labeling.

Features

- Features are composed of following building blocks:
  - Levenshtein distance, based on following edit operations:
    - insertion
    - deletion
    - substitution
    - all
  - Cost functions for Levenshtein edit operations:
    - unit cost for all operations
    - character-based edit-distance as cost function for substitution operation
    - probabilistic cost functions for substitution = generalized edit distance

Features, continued

- Probabilistic term substitution models based on Pointwise Mutual Information:

$$PMI = \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)}$$

  - Introduced by Church & Hanks (1990) as word association ratio.
  - Negative PMI values happen in rare events where strings co-occur less frequently than random:

$$p(w_i, w_j) < p(w_i)p(w_j)$$

  - Negative PMI values set to zero in our case.

Features, continued

- Normalizations of PMI:
  - Positive PMI values bounded to range between 0 and 1 by linear rescaling.

- Joint normalization:

$$PMI_J = \frac{PMI(w_i, w_j)}{-\log(p(w_i, w_j))}$$

  - Measures the amount of shared information between two strings relative to sum of the information of the individual strings.

Features, continued

- Specialization normalization:

$$PMI_S = \frac{PMI(w_i, w_j)}{-\log(p(w_i))}$$

  – $\mathrm{PMI_S}$ favors pairs where w_j is a specialization of w_i
  – $\mathrm{PMI_S}$ is at maximum when $p(w_i,w_j) = p(w_j)$, i.e. when $p(w_i|w_j) = 1$

- Generalization normalization:
  – w_j generalizes w_i:

$$PMI_G = \frac{PMI(w_i, w_j)}{-\log(p(w_j))}$$

Features, continued

- Examples:
  - PMI_G(apple, mac os) = .2917
  - PMI_S(apple, mac os) = .3686
  - Evidence for specialization.

  - PMI_G(ferrari models, ferrari) = 1
  - PMI_S(ferrari models, ferrari) = .5558
  - Perfect generalization.

  - PMI values computed from Web counts.

Features, continued

- Multiword queries:
  - Original order of query terms
  - Or: alphabetically sorted bag-of-words
- Estimation of cost matrix:
  - Relative frequency of session transitions in query log of 1.3 billion English queries
  - Smoothed transition probability from clustering model trained on user logs
- Resulting feature set of about 60 dense features

Learning to Rank Query Rewrites

- Various loss functions optimized in Stochastic Gradient Descent framework:
  - Training data $S = \{x_q^{(i)}, y_q^{(i)}\}_{i=1}^{n}$ where $x_q=\{x_{q1}, \ldots, x_{q,n(q)}\}$ is a set of rewrites for query q, and $y_q = (y_{q1}, \ldots, y_{q,n(q)})$ is a ranking on rewrites.
  - Minimize regularized objective for training set

$$\min_{w} \sum_{x_q, y_q} l(w) + \Omega(w)$$

by stochastic updating $$w_{t+1} = w_t - \eta_t g_t$$

where $$g_t = \nabla(l(w) + \Omega(w))$$

Learning to Rank Rewrites, cont.

- Conditional log-linear model on set(!) of relevant queries (Riezler et al. ACL'02) for binary relevance scores (expressed as rank *1* for relevant, and rank *2* for non-relevant rewrites):

$$\mathrm{l}_{llm}(w) = -\log \frac{\sum_{x_{qi} \in x_q | y_{qi}=1} e^{\langle w, \phi(x_{qi}) \rangle}}{\sum_{x_{qi} \in x_q} e^{\langle w, \phi(x_{qi}) \rangle}}$$

- Gradient: $$\frac{\partial}{\partial w_k} \mathrm{l}_{llm}(w) = -p_w\left[\phi_k \mid x_q; y_{qi} = 1\right] + p_w\left[\phi_k \mid x_q\right]$$

Learning to Rank Rewrites, cont.

- Listwise hinge loss for prediction loss $L(y_q,\pi_q)$ = MAP (Mean Average Precision) (= SVM-MAP of Yue et al. SIGIR'07):

$$\mathrm{l}_{lh}(w) = (L(y_q, \pi_q^*) - \langle w, \phi(x_q, y_q) - \phi(x_q, \pi_q^*) \rangle)_+$$

where $\pi_q^* = \arg\max_{\pi_q \in \Pi_q \setminus y_q} L(y_q, \pi_q) + \langle w, \phi(x_q, \pi_q) \rangle$

$(z)_+ = max\{0,z\}$, and $\phi(x_q,y_q)$ is a partial order feature map (see Yue et al.'07).

- Gradient:

$$\frac{\partial}{\partial w_k} \mathrm{l}_{lh}(w) = \begin{cases} 0 \quad if \quad \langle w, \phi(x_q, y_q) - \phi(x_q, \pi_q^*) \rangle > L(y_q, \pi_q^*) \\ -(\phi(x_q, y_q) - \phi(x_q, \pi_q^*)) \quad else \end{cases}$$

Learning to Rank Rewrites, cont.

- (Margin-rescaled) pairwise hinge loss (Joachims'02; Cortes et al. ICML'07; Agarwal & Niyogi JMLR'09;):

$$\ell_{ph}(w) = \sum_{(i,j)\in P_q} \left(\left(\left| \frac{1}{y_{qi}} - \frac{1}{y_{qj}} \right| \right) - \left\langle w, \phi(x_{qi}) - \phi(x_{qj}) \right\rangle \operatorname{sgn}\left(\frac{1}{y_{qi}} - \frac{1}{y_{qj}} \right) \right)_+$$

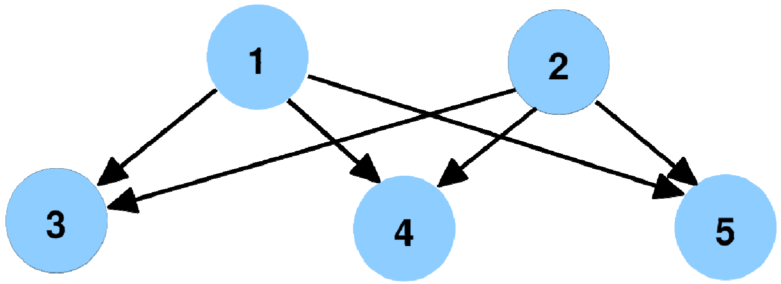
where P_q is the set of pairs of rewrites for query q that need to be compared.

- Gradient for SGD on pair-level:

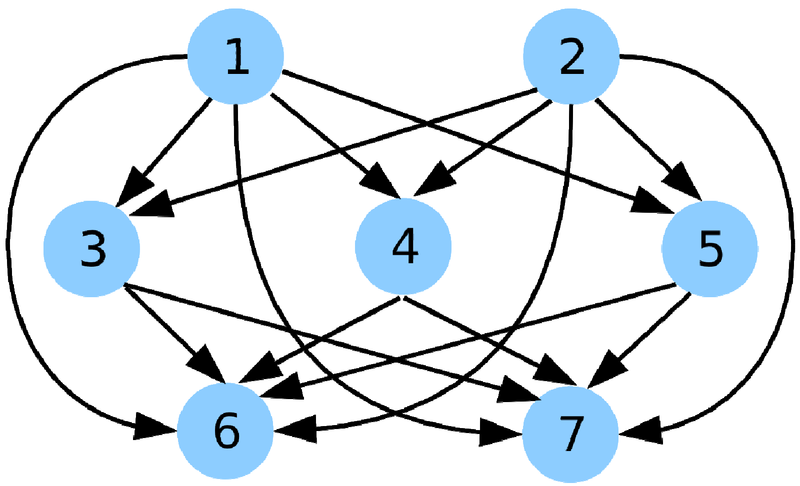
$$\frac{\partial}{\partial w_k} \ell_{ph}(w) = \begin{cases} 0 & \mathit{if} \quad \left\langle w, \phi(x_{qi}) - \phi(x_{qj}) \right\rangle \operatorname{sgn}\left(\frac{1}{y_{qi}} - \frac{1}{y_{qj}} \right) > \left| \frac{1}{y_{qi}} - \frac{1}{y_{qj}} \right| \\ -(\phi(x_{qi}) - \phi(x_{qj})) \operatorname{sgn}\left(\frac{1}{y_{qi}} - \frac{1}{y_{qj}} \right) & \mathit{else} \end{cases}$$

Learning to Rank Rewrites, cont.

- Bipartite pairwise ranking for binary relevances, e.g, co-clicks >= 10 vs. < 10:



- Multipartite pairwise ranking for relevance levels, e.g., number of co-clicks:



Experimental Evaluation

- Baselines:
  - Random shuffling of relevant/non-relevant rewrites.
  - Single dense feature that performed best on development set (clustering model log-probability used for cost-matrix estimation).
- SGD training:
  - Constant learning rates $\eta \in \{1, 0.5, 0.1, 0.01, 0.001\}$
  - Each metaparameter evaluated on development set after every fifth out of 100 passes over the training set.
- Evaluation:
  - Evaluated on manually labeled test set of 100 queries with 30 rewrites each.
  - Evaluation metrics Mean Average Precision (MAP), Normalized Discounted Cumulative Gain (NDCG), Area-under-the-ROC-curve (AUC), Precision@n.

Experimental Results

	MAP	NDCG@10	AUC	P@1	P@3	P@5
Random	51.8	48.7	50.4	45.6	45.6	46.6
Best Feature	71.9	70.2	74.5	70.2	68.1	68.7
Log-linear	74.7	75.1	75.7	75.3	72.2	71.3
SVM-MAP	74.3	75.2	75.3	76.3	71.8	72.0
SVM-bipartite	73.7	73.7	74.7	79.4	70.1	70.1
SVM-multipart.	76.5	77.3	77.2	83.5	74.2	73.6
SVM-multipart. -margin	75.7	76.0	76.6	82.5	72.9	73.0

Statistical Significance

- Statistical significance of result differences for pairwise system comparisons:
  - Approximate Randomization test with stratified shuffling applied to results on the query level (Noreen 1989)

	Best-feat.	**SVM-bipart.**	**SVM-MAP**	**Log-linear**	**SVM-multi.-marg.**	**SVM-multi.**
Best-feature	-	<0.005	<0.005	<0.005	<0.005	<0.005
SVM-bipart.	-	-	**0.324**	<0.005	<0.005	<0.005
SVM-MAP	-	-	-	**0.374**	<0.005	<0.005
Log-linear	-	-	-	-	**0.053**	<0.005
SVM-multi.-marg.	-	-	-	-	-	<0.005
SVM-multi.	-	-	-	-	-	-

Experimental Results

- Evaluation results:
  - SVM-multipartite outperforms all other ranking systems under all evaluation metrics at a significance level >= 0.995.
  - Result differences for systems ranked next to each other are not statistically significant.
  - All systems outperform random and best-feature baselines.
- Discussion:
  - SVM-multipartite ranker is most robust across all eval metrics.
  - Position-sensitive margin rescaling does not help.
  - SVM-MAP overtrains on dev set, thus does not win on MAP evaluation.

Conclusion

- Research questions:
  - Is number of co-clicks useful implicit feedback to create multipartite rankings for training rankers?
  - Are machine learning techniques robust enough to learn from noisy data and achieve good performance w.r.t. human quality standards?
- Results:
  - Co-click information could be shown to correlate well with human judgments on rewrite quality
  - Large-scale experiment finds robust learner in multipartiee-ranking SVM
- TODO:
  - More support needed from extrinsic evaluation / live search experiment!