RIN: Reformulation Inference Network for Context-Aware Query Suggestion

Jyun-Yu Jiang and Wei Wang

University of California, Los Angeles (UCLA)

Oct 23, 2018 (CIKM)

Context-Aware Query Suggestion

- Context captures user's search intents.
  - submitted queries
  - click-through information

$$\underbrace{q_1 \rightarrow q_2 \rightarrow \cdots \rightarrow \cdot q_{T-1}}_{context} \rightarrow q_T$$

- Previous work statistically models query dependencies and similarity.

Query Session

- query dependencies
- query similarity
- personal history

Click-through Data

- relevant queries
- query clusters
- clicked webpages

However, a user may have some behaviors to reformulate their queries.

Context-Aware Query Suggestion

- Context captures user's search intents.
  - submitted queries
  - click-through information

$$\underbrace{q_1 \rightarrow q_2 \rightarrow \cdots \rightarrow \cdot q_{T-1}}_{context} \rightarrow q_T$$

- Previous work statistically models query dependencies and similarity.

Query Session

- query dependencies
- query similarity
- personal history

Click-through Data

- relevant queries
- query clusters
- clicked webpages

However, a user may have some behaviors to reformulate their queries.

User Reformulation Behavior

- Studied as query reformulation strategies.

Syntactic Relations – Simple to Analyze

- Syntactic and explicit changes between queries
  - Such as adding terms, removing terms, acronym expansion.
- Clear definitions of reformulation types.

Semantic Relations – Difficult to Analyze

- *specialization*: narrow the search constraints, e.g., *computer* → *mac*
- *generalization*: relax the search constraints, e.g., *lion* → *animal*

How to exploit user reformulation behavior for query suggestion?

User Reformulation Behavior

- Studied as query reformulation strategies.

Syntactic Relations – Simple to Analyze

- Syntactic and explicit changes between queries
  - Such as adding terms, removing terms, acronym expansion.
- Clear definitions of reformulation types.

Semantic Relations – Difficult to Analyze

- *specialization*: narrow the search constraints, e.g., *computer* → *mac*
- *generalization*: relax the search constraints, e.g., *lion* → *animal*

How to exploit user reformulation behavior for query suggestion?

User Reformulation Behavior

- Studied as query reformulation strategies.

Syntactic Relations – Simple to Analyze

- Syntactic and explicit changes between queries
  - Such as adding terms, removing terms, acronym expansion.
- Clear definitions of reformulation types.

Semantic Relations – Difficult to Analyze

- *specialization*: narrow the search constraints, e.g., *computer* → *mac*
- *generalization*: relax the search constraints, e.g., *lion* → *animal*

How to exploit user reformulation behavior for query suggestion?

User Reformulation Behavior

- Studied as query reformulation strategies.

Syntactic Relations – Simple to Analyze

- Syntactic and explicit changes between queries
  - Such as adding terms, removing terms, acronym expansion.
- Clear definitions of reformulation types.

Semantic Relations – Difficult to Analyze

- *specialization*: narrow the search constraints, e.g., *computer* → *mac*
- *generalization*: relax the search constraints, e.g., *lion* → *animal*

How to exploit user reformulation behavior for query suggestion?

Approximated Semantic Relations from Syntactic Relations

Assumption of Semantic Relations [SIGIR'14]

- Specialization
  - Narrow the search constraints
  - More terms are required to describe the intents (constraints).
- Generalization:
  - Relax the search constraints
  - Terms (constraints) can be removed.

- The syntactic analysis is supposed help us learn semantic relations.

However, existing approaches based on the above may be fragile.

Approximated Semantic Relations from Syntactic Relations

Assumption of Semantic Relations [SIGIR'14]

- Specialization
  - Narrow the search constraints
  - More terms are required to describe the intents (constraints).
- Generalization:
  - Relax the search constraints
  - Terms (constraints) can be removed.

- The syntactic analysis is supposed help us learn semantic relations.

However, existing approaches based on the above may be fragile.

Syntactic features can be discrete and hard to be defined

Table 2: Defined reformulation behavior and the formulas for calculating reformulation features.

Category	Feature Class	Description	Formulas
Term	Term Combination (16 features)	number of terms	$\lvert \cup_{i=1}^{T} S(q_i) \rvert$, $\lvert S(q_{T-1}) \cup S(q_T) \rvert$
		term keeping	$\lvert \cap_{i=1}^{T} S(q_i) \rvert$, $\lvert S(q_{T-1}) \cap S(q_T) \rvert$, $\mathrm{sgn}(\lvert S(q_{T-1}) \cap S(q_T) \rvert)$
		term adding	$\lvert S(q_T) - S(q_{T-1}) \rvert$, $\mathrm{sgn}(\lvert S(q_T) - S(q_{T-1}) \rvert)$
		term removing	$\lvert S(q_{T-1}) - S(q_T) \rvert$, $\mathrm{sgn}(\lvert S(q_{T-1}) - S(q_T) \rvert)$
		number of used terms	$\lvert S_{\mathrm{used}}(q_T) \rvert$, $\lvert S(q_T) - S_{\mathrm{used}}(q_T) \rvert$
		ratio of used terms	$\lvert S_{\mathrm{used}}(q_T) \rvert / \lvert S(q_T) \rvert$, $1 - \lvert S_{\mathrm{used}}(q_T) \rvert / \lvert S(q_T) \rvert$
		number of repeat times	$\mathrm{Rep}(q_T)$, $\mathrm{Rep}(q_T)/T$, $\mathrm{Rep}(q_T)/\lvert S(q_T) \rvert$
Query	Query Similarity (10 features)	cosine similarity	$\mathrm{sim}_{\mathrm{cos}}(q_{T-1}, q_T)$
		average cosine similarity	$\frac{1}{T-1} \sum_{i=1}^{T-1} \mathrm{sim}_{\mathrm{cos}}(q_i, q_{i+1}))$, $\frac{1}{T-1} \sum_{i=1}^{T-1} \mathrm{sim}_{\mathrm{cos}}(q_i, q_T))$
		trends of cosine similarity	$\mathrm{sim}_{\mathrm{cos}}(q_{T-1}, q_T) / \frac{1}{T-2} \sum_{i=1}^{T-2} \mathrm{sim}_{\mathrm{cos}}(q_i, q_{i+1}))$
			$\mathrm{sim}_{\mathrm{cos}}(q_{T-1}, q_T) / \frac{1}{T-2} \sum_{i=1}^{T-2} \mathrm{sim}_{\mathrm{cos}}(q_i, q_T))$
		Lev. similarity	$\mathrm{sim}_{\mathrm{Lev}}(q_{T-1}, q_T)$
		average Lev. similarity	$\frac{1}{T-1} \sum_{i=1}^{T-1} \mathrm{sim}_{\mathrm{Lev}}(q_i, q_{i+1}))$, $\frac{1}{T-1} \sum_{i=1}^{T-1} \mathrm{sim}_{\mathrm{Lev}}(q_i, q_T))$
		trends of Lev. similarity	$\mathrm{sim}_{\mathrm{Lev}}(q_{T-1}, q_T) / \frac{1}{T-2} \sum_{i=1}^{T-2} \mathrm{sim}_{\mathrm{Lev}}(q_i, q_{i+1}))$
			$\mathrm{sim}_{\mathrm{Lev}}(q_{T-1}, q_T) / \frac{1}{T-2} \sum_{i=1}^{T-2} \mathrm{sim}_{\mathrm{Lev}}(q_i, q_T))$
	Query Length (6 features)	number of terms	$\lvert S(q_T) \rvert$
		average number of terms	$\frac{1}{T-1} \sum_{i=1}^{T-1} \lvert S(q_i) \rvert$, $\frac{1}{T} \sum_{i=1}^{T} \lvert S(q_i) \rvert$, $\lvert S(q_{T-1}) \rvert + \lvert S(q_T) \rvert$
		trends of term number	$\lvert S(q_T) \rvert / \frac{1}{T-1} \sum_{i=1}^{T-1} \lvert S(q_i) \rvert$, $\lvert S(q_{T-1}) \rvert - \lvert S(q_T) \rvert$
	Query Frequency (2 features)	pairwise frequency	$P((q_{T-1}, q_T) \mid q_T)$, $P((q_{T-1}, q_T) \mid q_{T-1})$
Session	Click-through Data (6 features)	previous clicks	c_{T-1} , $\mathrm{sgn}(c_{T-1})$
		number of effective terms	$\lvert C_{\mathrm{eff}}(q_T) \rvert$
		ratio of effective terms	$\lvert C_{\mathrm{eff}}(q_T) \rvert / T$, $\lvert C_{\mathrm{eff}}(q_T) \rvert / \lvert S(q_T) \rvert$, $\lvert C_{\mathrm{eff}}(q_T) \rvert / \lvert S_{\mathrm{used}}(q_T) \rvert$
	Time Duration (2 features)	average time duration	$\frac{1}{T-1} \sum_{i=1}^{T-1} (t_{i+1} - t_i)$
		trends of time duration	$(t_T - t_{T-1}) / \frac{1}{T-2} \sum_{i=1}^{T-2} (t_{i+1} - t_i)$
	Position Number (1 feature)	position in the session	(T)

Figure: The table of features in the SIGIR'14 paper.

Contradictory Semantics Relations

- Contradictions in the analysis of previous studies.
- 15.8% to 17.5% of consecutive queries contradict the assumption.

Relation	% in Log	Avg. Pos.	Med. Pos.	Change of Term Number	% in Relation	Example
Specialization	27.7%	2.9951	2	Increase	84.2%	camera → digital camera
				Decrease	**3.7%**	perennial plants → stonecrop
				Equal	**12.1%**	guest book for party → anniversary party guest book
Generalization	12.2%	3.3122	3	Increase	**4.0%**	airport parking newark → airport parking new york
				Decrease	82.5%	great lakes auto → great lakes
				Equal	**13.5%**	honda blue book → car blue book

We need robust representations for modeling semantic reformulations.

Contradictory Semantics Relations

- Contradictions in the analysis of previous studies.
- 15.8% to 17.5% of consecutive queries contradict the assumption.

Relation	% in Log	Avg. Pos.	Med. Pos.	Change of Term Number	% in Relation	Example
Specialization	27.7%	2.9951	2	Increase	84.2%	camera → digital camera
				Decrease	**3.7%**	perennial plants → stonecrop
				Equal	**12.1%**	guest book for party → anniversary party guest book
Generalization	12.2%	3.3122	3	Increase	**4.0%**	airport parking newark → airport parking new york
				Decrease	82.5%	great lakes auto → great lakes
				Equal	**13.5%**	honda blue book → car blue book

We need robust representations for modeling semantic reformulations.

Problem Statement

- Suppose the user intends to submit a query q_{L+1} after the search context $\langle q_1, q_2, \cdots, q_L \rangle$, we have two goals of query suggestion.

Discriminative Query Suggestion

- Given a set of candidate queries Q_{can}.
- Rank candidates $q_{can} \in Q_{can}$ so that q_{L+1} ranks as high as possible.

Generative Query Suggestion

- Generate a query q'_{L+1} expected to be as similar as possible to q_{L+1}.

Problem Statement

- Suppose the user intends to submit a query q_{L+1} after the search context $\langle q_1, q_2, \cdots, q_L \rangle$, we have two goals of query suggestion.

Discriminative Query Suggestion

- Given a set of candidate queries Q_{can}.
- Rank candidates $q_{can} \in Q_{can}$ so that q_{L+1} ranks as high as possible.

Generative Query Suggestion

- Generate a query q'_{L+1} expected to be as similar as possible to q_{L+1}.

Problem Statement

- Suppose the user intends to submit a query q_{L+1} after the search context $\langle q_1, q_2, \cdots, q_L \rangle$, we have two goals of query suggestion.

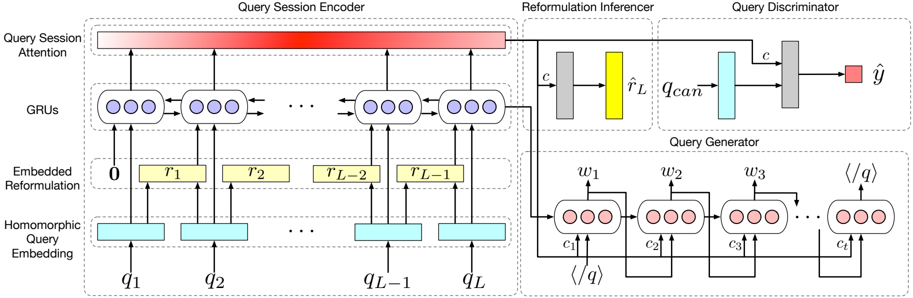
Discriminative Query Suggestion

- Given a set of candidate queries Q_{can}.
- Rank candidates $q_{can} \in Q_{can}$ so that q_{L+1} ranks as high as possible.

Generative Query Suggestion

- Generate a query q'_{L+1} expected to be as similar as possible to q_{L+1}.

Framework: Reformulation Inference Network



Homomorphic Query Embedding

- Suppose every term t has a representative embedding $\boldsymbol{v}_t$.
- The homomorphic embedding of a query q is defined as

$$\boldsymbol{v}_q = \sum_{t \in T(q)} \boldsymbol{v}_t$$

- The reformulation $\boldsymbol{r_i}$ from q_i to q_{i+1} can be represented as

$$\boldsymbol{v}_{q_{i+1}} - \boldsymbol{v}_{q_i}$$

Homomorphic query embedding is beneficial.

- The syntactic relations are homomorphically preserved.
  - e.g., $\boldsymbol{v}_{\texttt{Tokyo hotel}} - \boldsymbol{v}_{\texttt{Japan hotel}} = +\boldsymbol{v}_{\texttt{Tokyo}} - \boldsymbol{v}_{\texttt{Japan}}$
- The latent space of embeddings implicitly captures query semantics.
  - Queries with similar semantics are also close in the space.
- The embeddings have linear substructures.
  - Helpful to understand the semantic relations between reformulations.
  - High interpretability for reformulation embeddings.

Homomorphic query embedding is beneficial.

- The syntactic relations are homomorphically preserved.
  - e.g., $\boldsymbol{v}_{\texttt{Tokyo hotel}} - \boldsymbol{v}_{\texttt{Japan hotel}} = +\boldsymbol{v}_{\texttt{Tokyo}} - \boldsymbol{v}_{\texttt{Japan}}$
- The latent space of embeddings implicitly captures query semantics.
  - Queries with similar semantics are also close in the space.
- The embeddings have linear substructures.
  - Helpful to understand the semantic relations between reformulations.
  - High interpretability for reformulation embeddings.

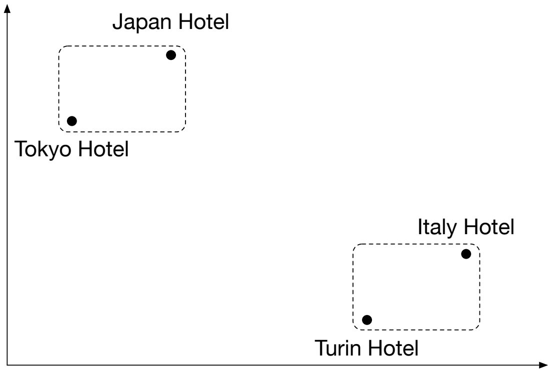
Homomorphic query embedding is beneficial.

- The syntactic relations are homomorphically preserved.
  - e.g., $\boldsymbol{v}_{\texttt{Tokyo hotel}} - \boldsymbol{v}_{\texttt{Japan hotel}} = +\boldsymbol{v}_{\texttt{Tokyo}} - \boldsymbol{v}_{\texttt{Japan}}$
- The latent space of embeddings implicitly captures query semantics.
  - Queries with similar semantics are also close in the space.
- The embeddings have linear substructures.
  - Helpful to understand the semantic relations between reformulations.
  - High interpretability for reformulation embeddings.

Homomorphic query embedding is beneficial.

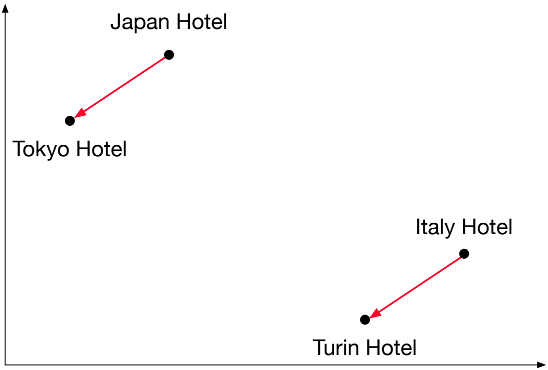
- The syntactic relations are homomorphically preserved.
  - e.g., $\boldsymbol{v}_{\texttt{Tokyo hotel}} - \boldsymbol{v}_{\texttt{Japan hotel}} = +\boldsymbol{v}_{\texttt{Tokyo}} - \boldsymbol{v}_{\texttt{Japan}}$
- The latent space of embeddings implicitly captures query semantics.
  - Queries with similar semantics are also close in the space.
- The embeddings have linear substructures.
  - Helpful to understand the semantic relations between reformulations.
  - High interpretability for reformulation embeddings.

Semantic Homomorphic Embeddings



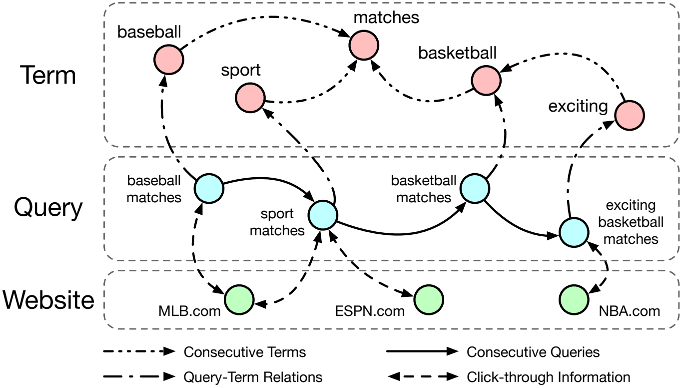
Queries with similar semantics are close in the space.

Semantic Homomorphic Embeddings



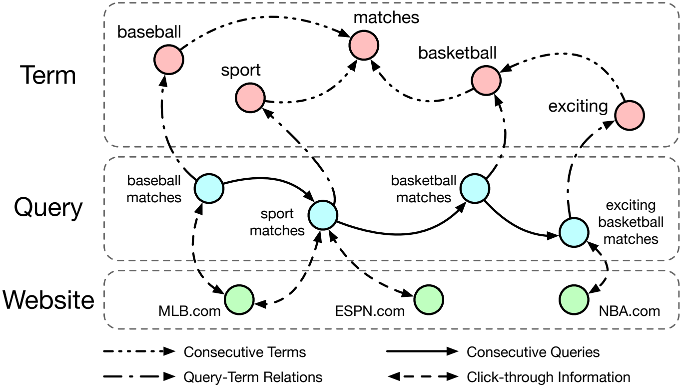
The linear substructures reveal the semantics of reformulations.

Learning Term Embeddings with Heterogeneous Networks



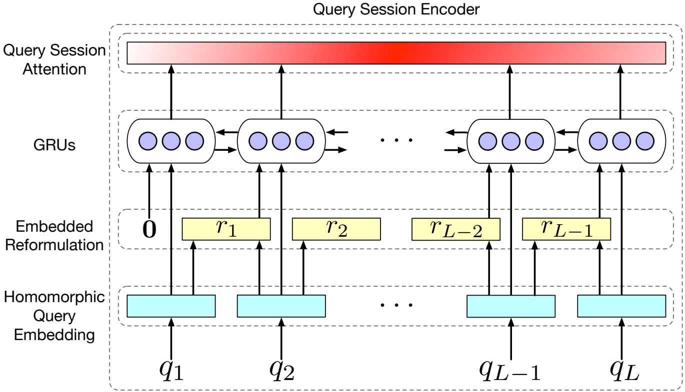
Here node2vec is applied to derive term embeddings.

Learning Term Embeddings with Heterogeneous Networks

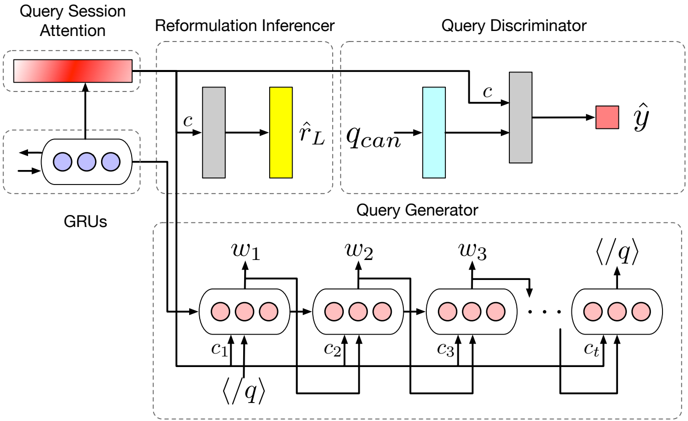


Here node2vec is applied to derive term embeddings.

Attention-based Query Session Encoder with a RNN



Three Tasks in Reformulation Inference Network



Multi-task Learning

- Reformulation Inferencer
  - Minimize distances as a regression problem.
$$\text{loss}_R = \frac{1}{2}||\boldsymbol{r_L} - \hat{\boldsymbol{r}}_{\boldsymbol{L}}||_F^2$$
- Query Discriminator
  - Discriminate queries as a classification problem.
$$\text{loss}_D = -\left(y \log(\hat{y}) + (1-y)\log(1-\hat{y})\right)$$
- Query Generator
  - Generate term sequences as a generation problem.
$$\text{loss}_G = -\sum_{w_t} \log P(w_t \mid S_t)$$
- Final Objectives: $\text{loss} = \text{loss}_R + \text{loss}_{\text{task}}$
  - $\text{loss}_{\text{task}}$ could be loss_D, loss_G, or $\text{loss}_D + \text{loss}_G$

Multi-task Learning

- Reformulation Inferencer
  - Minimize distances as a regression problem.

$$\text{loss}_R = \frac{1}{2}||\boldsymbol{r_L} - \boldsymbol{\hat{r}_L}||_F^2$$

- Query Discriminator
  - Discriminate queries as a classification problem.

$$\text{loss}_D = -\left(y \log(\hat{y}) + (1-y)\log(1-\hat{y})\right)$$

- Query Generator
  - Generate term sequences as a generation problem.

$$\text{loss}_G = -\sum_{w_t} \log P(w_t \mid S_t)$$

- Final Objectives: $\text{loss} = \text{loss}_R + \text{loss}_{\text{task}}$
  - $\text{loss}_{\text{task}}$ could be loss_D, loss_G, or $\text{loss}_D + \text{loss}_G$

Multi-task Learning

- Reformulation Inferencer
  - Minimize distances as a regression problem.

$$\mathrm{loss}_R = \frac{1}{2}||\boldsymbol{r_L} - \hat{\boldsymbol{r}}_{\boldsymbol{L}}||_F^2$$

- Query Discriminator
  - Discriminate queries as a classification problem.

$$\mathrm{loss}_D = -\left(y\log(\hat{y}) + (1-y)\log(1-\hat{y})\right)$$

- Query Generator
  - Generate term sequences as a generation problem.

$$\mathrm{loss}_G = -\sum_{w_t} \log P(w_t \mid S_t)$$

- Final Objectives: $\mathrm{loss} = \mathrm{loss}_R + \mathrm{loss}_{\mathrm{task}}$
  - $\mathrm{loss}_{\mathrm{task}}$ could be loss_D, loss_G, or $\mathrm{loss}_D + \mathrm{loss}_G$

Multi-task Learning

- Reformulation Inferencer
  - Minimize distances as a regression problem.

$$\text{loss}_R = \frac{1}{2}||\boldsymbol{r_L} - \boldsymbol{\hat{r}_L}||_F^2$$

- Query Discriminator
  - Discriminate queries as a classification problem.

$$\text{loss}_D = -\left(y \log(\hat{y}) + (1-y)\log(1-\hat{y})\right)$$

- Query Generator
  - Generate term sequences as a generation problem.

$$\text{loss}_G = -\sum_{w_t} \log P(w_t \mid S_t)$$

- Final Objectives: $\text{loss} = \text{loss}_R + \text{loss}_{\text{task}}$
  - $\text{loss}_{\text{task}}$ could be loss_D, loss_G, or $\text{loss}_D + \text{loss}_G$

Multi-task Learning

- Reformulation Inferencer
  - Minimize distances as a regression problem.

$$\text{loss}_R = \frac{1}{2}||\boldsymbol{r_L} - \boldsymbol{\hat{r}_L}||_F^2$$

- Query Discriminator
  - Discriminate queries as a classification problem.

$$\text{loss}_D = -\left(y \log(\hat{y}) + (1 - y)\log(1 - \hat{y})\right)$$

- Query Generator
  - Generate term sequences as a generation problem.

$$\text{loss}_G = -\sum_{w_t} \log P(w_t \mid S_t)$$

- Final Objectives: $\text{loss} = \text{loss}_R + \text{loss}_{\text{task}}$
  - $\text{loss}_{\text{task}}$ could be loss_D, loss_G, or $\text{loss}_D + \text{loss}_G$

Experimental Settings

- 3-month AOL search engine logs
  - 2 months for training, and 1 month for testing.
  - Randomly sample 10% of training data for validation.
- Evaluate Metrics
  - Discriminative Task: mean reciprocal rank (MRR).
  - Generative Task: position independent word error rate (PER).

Dataset	Context Length		
	Short (1 query)	Medium (2-3 queries)	Long (4+ queries)
Training	852,350	386,970	118,180
Testing	403,772	184,843	58,944

Seven Competitive Baselines

- Dependency-based Baseline Methods
  - Most Popular Suggestion (MPS)
  - Query-based Variable Markov Model (QVMM) [ICDE'09]
- Similarity-based Baseline Method
  - Hybrid Completion (HYB) [WWW'11]
- Feature-based Baseline Methods
  - Personalized Completion (PC) [SIGIR'13]
  - Reformulation-based Syntactic Features (RC) [SIGIR'14]
- Deep Learning Baseline Methods
  - Hierarchical Recurrent Encoder-Decoder (HRED) [CIKM'15]
  - Seq2Seq with Copiers (ACG) [CIKM'17]

Discriminative Query Suggestion (MRR)

Dataset	MPS	Hybrid	PC	QVMM
Overall Context	0.5471	0.5823	0.5150	0.5671
Short Context	0.5680	0.5822	0.5343	0.5862
Medium Context	0.5167	0.5841	0.4865	0.5338
Long Context	0.4826	0.5768	0.4575	0.5026
Dataset	**RC**	**HRED**	**ACG**	**RIN**
Overall Context	0.6202	0.6207	0.6559	**0.8254**
Short Context	0.5960	0.6100	0.6471	**0.8361**
Medium Context	0.6689	0.6489	0.6542	**0.8190**
Long Context	0.6704	0.6122	0.6669	**0.7611**

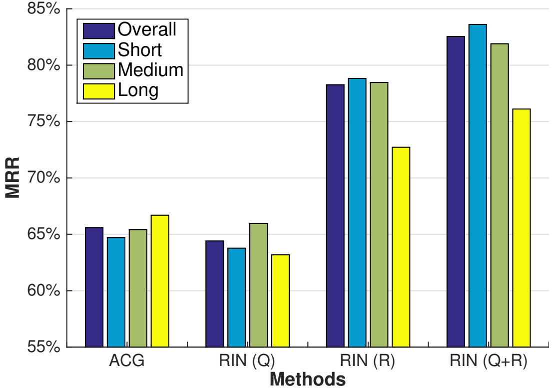
Generative Query Suggestion (PER)

Only two baseline methods are capable for generative query suggestion.

Dataset	HRED	ACG	RIN
Overall	0.8069	0.6925	**0.6612**
Short	0.8179	0.7015	**0.6851**
Medium	0.8338	0.6733	**0.6197**
Long	0.6753	0.6673	**0.6115**

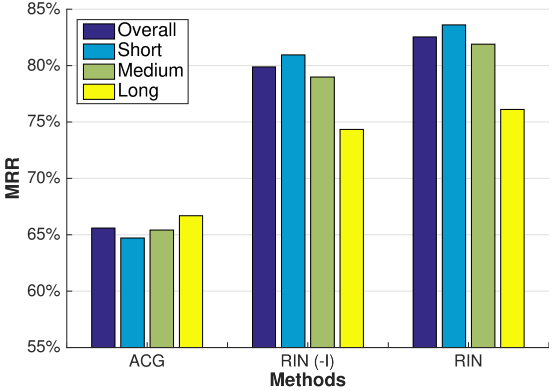
Effectiveness of Reformulation Embeddings

Q and R represent query and reformulation embeddings used in RIN.



Effectiveness of Reformulation Inferencer

(-I) means the removal of the reformulation inferencer.



Conclusions

- Proposed RIN to model reformulation behaviors for query suggestion.
- Homomorphic query embedding provides flexible reformulation embeddings.
- An attention-based RNN encodes sessions with homomorphic embeddings.
- Jointly optimized tasks and reformulation inferencer for better suggestions.
- Outperformed seven competitive baselines in extensive experiments.
- See our paper for more detailed parameter sensitivity experiments.
- Thank the SIGIR travel grant for the great support!

Thanks for your attention! Questions?

Conclusions

- Proposed RIN to model reformulation behaviors for query suggestion.
- Homomorphic query embedding provides flexible reformulation embeddings.
- An attention-based RNN encodes sessions with homomorphic embeddings.
- Jointly optimized tasks and reformulation inferencer for better suggestions.
- Outperformed seven competitive baselines in extensive experiments.
- See our paper for more detailed parameter sensitivity experiments.
- Thank the SIGIR travel grant for the great support!

Thanks for your attention! Questions?

Conclusions

- Proposed RIN to model reformulation behaviors for query suggestion.
- Homomorphic query embedding provides flexible reformulation embeddings.
- An attention-based RNN encodes sessions with homomorphic embeddings.
- Jointly optimized tasks and reformulation inferencer for better suggestions.
- Outperformed seven competitive baselines in extensive experiments.
- See our paper for more detailed parameter sensitivity experiments.
- Thank the SIGIR travel grant for the great support!

Thanks for your attention! Questions?

Conclusions

- Proposed RIN to model reformulation behaviors for query suggestion.
- Homomorphic query embedding provides flexible reformulation embeddings.
- An attention-based RNN encodes sessions with homomorphic embeddings.
- Jointly optimized tasks and reformulation inferencer for better suggestions.
- Outperformed seven competitive baselines in extensive experiments.
- See our paper for more detailed parameter sensitivity experiments.
- Thank the SIGIR travel grant for the great support!

Thanks for your attention! Questions?

Conclusions

- Proposed RIN to model reformulation behaviors for query suggestion.
- Homomorphic query embedding provides flexible reformulation embeddings.
- An attention-based RNN encodes sessions with homomorphic embeddings.
- Jointly optimized tasks and reformulation inferencer for better suggestions.
- Outperformed seven competitive baselines in extensive experiments.
- See our paper for more detailed parameter sensitivity experiments.
- Thank the SIGIR travel grant for the great support!

Thanks for your attention! Questions?

Conclusions

- Proposed RIN to model reformulation behaviors for query suggestion.
- Homomorphic query embedding provides flexible reformulation embeddings.
- An attention-based RNN encodes sessions with homomorphic embeddings.
- Jointly optimized tasks and reformulation inferencer for better suggestions.
- Outperformed seven competitive baselines in extensive experiments.
- See our paper for more detailed parameter sensitivity experiments.
- Thank the SIGIR travel grant for the great support!

Thanks for your attention! Questions?

Conclusions

- Proposed RIN to model reformulation behaviors for query suggestion.
- Homomorphic query embedding provides flexible reformulation embeddings.
- An attention-based RNN encodes sessions with homomorphic embeddings.
- Jointly optimized tasks and reformulation inferencer for better suggestions.
- Outperformed seven competitive baselines in extensive experiments.
- See our paper for more detailed parameter sensitivity experiments.
- Thank the SIGIR travel grant for the great support!

Thanks for your attention! Questions?

Conclusions

- Proposed RIN to model reformulation behaviors for query suggestion.
- Homomorphic query embedding provides flexible reformulation embeddings.
- An attention-based RNN encodes sessions with homomorphic embeddings.
- Jointly optimized tasks and reformulation inferencer for better suggestions.
- Outperformed seven competitive baselines in extensive experiments.
- See our paper for more detailed parameter sensitivity experiments.
- Thank the SIGIR travel grant for the great support!

Thanks for your attention! Questions?

Conclusions

- Proposed RIN to model reformulation behaviors for query suggestion.
- Homomorphic query embedding provides flexible reformulation embeddings.
- An attention-based RNN encodes sessions with homomorphic embeddings.
- Jointly optimized tasks and reformulation inferencer for better suggestions.
- Outperformed seven competitive baselines in extensive experiments.
- See our paper for more detailed parameter sensitivity experiments.
- Thank the SIGIR travel grant for the great support!

Thanks for your attention! Questions?