

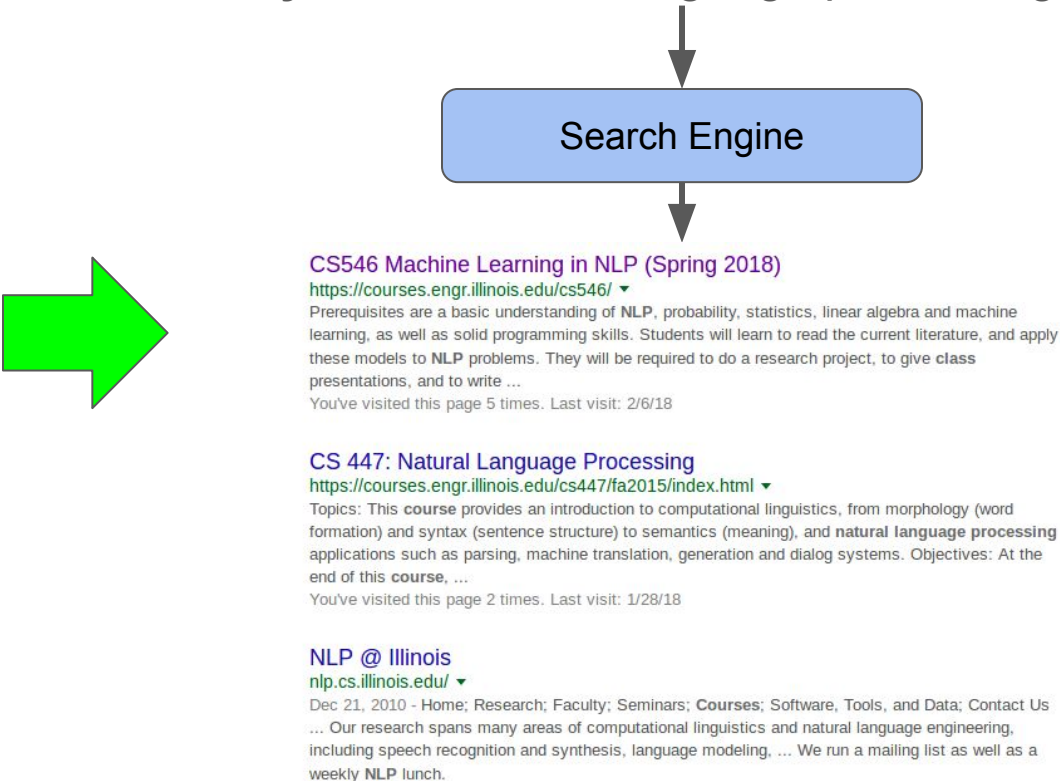
# Task-Oriented Query Reformulation with Reinforcement Learning

Authors: Rodrigo Nogueira and Kyunghyun Cho

Slides: Chris Benson

# Motivation

Query: “uiuc natural language processing class”



Search Engine

## CS546 Machine Learning in NLP (Spring 2018)

<https://courses.engr.illinois.edu/cs546/> ▼

Prerequisites are a basic understanding of NLP, probability, statistics, linear algebra and machine learning, as well as solid programming skills. Students will learn to read the current literature, and apply these models to NLP problems. They will be required to do a research project, to give class presentations, and to write ...

You've visited this page 5 times. Last visit: 2/6/18

## CS 447: Natural Language Processing

<https://courses.engr.illinois.edu/cs447/fa2015/index.html> ▼

Topics: This course provides an introduction to computational linguistics, from morphology (word formation) and syntax (sentence structure) to semantics (meaning), and natural language processing applications such as parsing, machine translation, generation and dialog systems. Objectives: At the end of this course, ...

You've visited this page 2 times. Last visit: 1/28/18

## NLP @ Illinois

[nlp.cs.illinois.edu/](http://nlp.cs.illinois.edu/) ▼

Dec 21, 2010 - Home; Research; Faculty; Seminars; Courses; Software, Tools, and Data; Contact Us ... Our research spans many areas of computational linguistics and natural language engineering, including speech recognition and synthesis, language modeling, ... We run a mailing list as well as a weekly NLP lunch.

# Motivation

Query: “uiuc class ai language words computer science”

Search Engine

Course Descriptions | CS @ ILLINOIS - Computer Science at the ...

<https://cs.illinois.edu/courses/profile/CS421> ▼

May 28, 2013 - CS 421 - Progrmg Languages & Compilers ... Progrmg Languages & Compilers, B3, 31375, LCD, 3, 1700 - 1815, T R , 1320 Digital Computer Lab, A Mattox Beckman, Jr. Progrmg ...

Program translation: be able to write a syntax-directed translator from abstract syntax to intermediate representations (a), (j)



Programming Languages, Formal Methods, and ... - CS @ Illinois

<https://cs.illinois.edu/.../programming-languages-formal-methods-and-software-engine...> ▼

word cloud with programming-related words The growing complexity and scale of software poses formidable challenges for reliability, security, performance, and productivity. Our faculty tackle these problems by developing innovative techniques in programming language design and semantics; techniques and tools for ...

Artificial Intelligence | CS @ ILLINOIS - Computer Science at the ...

<https://cs.illinois.edu/research/artificial-intelligence> ▼

Natural language processing systems understand written and spoken language; possibilities include automatic translation of text from one language to another ... Crain's Chicago Business -- Based on the deep computer science talent pool produced by the University of Illinois and other area schools, ServiceNow plans hire ...

# Motivation

Using **inexact** or **long** queries in search engines tend to result in poor document retrieval

- Vocabulary Mismatch Problem
- Iterative Searching

# Idea: Automatic Query Reformulation

Query: “uiuc class ai language words computer science”

Search Engine

Course Descriptions | CS @ ILLINOIS - Computer Science at the ...

<https://cs.illinois.edu/courses/profile/CS421> ▼

May 28, 2013 - CS 421 - Progrmg Languages & Compilers ... Progrmg Languages & Compilers, B3, 31375, LCD, 3, 1700 - 1815, T R , 1320 Digital Computer Lab, A Mattox Beckman, Jr. Progrmg ...  
Program translation: be able to write a syntax-directed translator from abstract syntax to intermediate representations (a), (j)

Programming Languages, Formal Methods, and ... - CS @ Illinois

<https://cs.illinois.edu/.../programming-languages-formal-methods-and-software-engine...> ▼

word cloud with programming-related words The growing complexity and scale of software poses formidable challenges for reliability, security, performance, and productivity. Our faculty tackle these problems by developing innovative techniques in programming language design and semantics; techniques and tools for ...

Artificial Intelligence | CS @ ILLINOIS - Computer Science at the ...

<https://cs.illinois.edu/research/artificial-intelligence> ▼

Natural language processing systems understand written and spoken language; possibilities include automatic translation of text from one language to another ... Crain's Chicago Business -- Based on the deep computer science talent pool produced by the University of Illinois and other area schools, ServiceNow plans hire ...

Reformulator

Query: “uiuc natural language processing class”

CS546 Machine Learning in NLP (Spring 2018)

<https://courses.engr.illinois.edu/cs546/> ▼

Prerequisites are a basic understanding of NLP, probability, statistics, linear algebra and machine learning, as well as solid programming skills. Students will learn to read the current literature, and apply these models to NLP problems. They will be required to do a research project, to give class presentations, and to write ...

You've visited this page 5 times. Last visit: 2/6/18

CS 447: Natural Language Processing

<https://courses.engr.illinois.edu/cs447/fa2015/index.html> ▼

Topics: This course provides an introduction to computational linguistics, from morphology (word formation) and syntax (sentence structure) to semantics (meaning), and natural language processing applications such as parsing, machine translation, generation and dialog systems. Objectives: At the end of this course, ...

You've visited this page 2 times. Last visit: 1/28/18

NLP @ Illinois

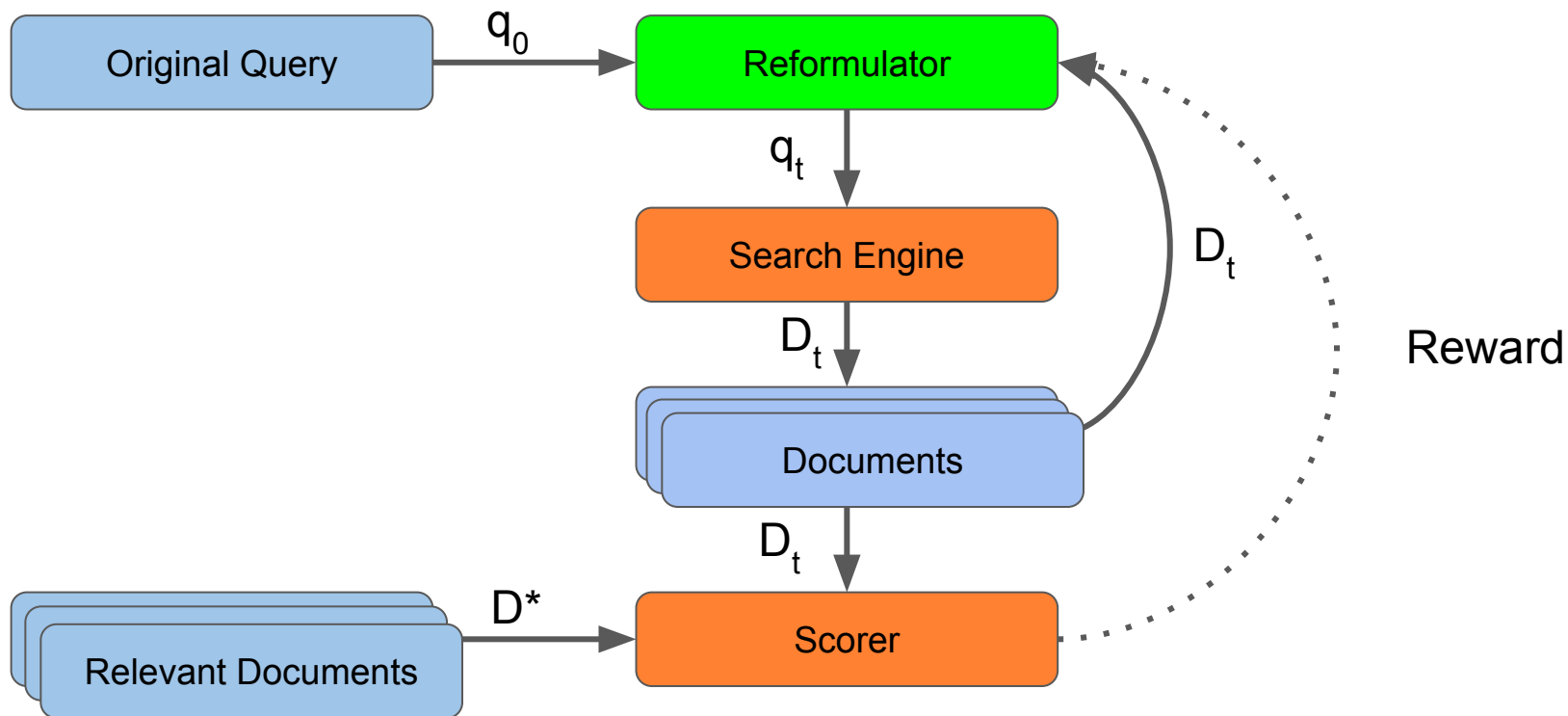
[nlp.cs.illinois.edu/](http://nlp.cs.illinois.edu/) ▼

Dec 21, 2010 - Home; Research; Faculty; Seminars; Courses; Software, Tools, and Data; Contact Us ... Our research spans many areas of computational linguistics and natural language engineering, including speech recognition and synthesis, language modeling, ... We run a mailing list as well as a weekly NLP lunch.

# Model as a Reinforcement Learning Problem

- Hard to create annotated data for queries
  - What is the “correct” query?
  - Successful queries are not unique
- Learn directly from reward based on relevant document retrieval
- Train to use search engine as a black box

# Automatic Query Reformulation



# Reinforcement Learning: Policy Algorithms

- Directly learn policy of how to act
- Policy ( $\pi$ ) gives probabilities of taking an action ( $a$ ) in a given state ( $s$ ) using parameters  $\theta$

$$\pi_{\theta}(a,s) = P(a \mid s, \theta)$$

- Find policy that maximizes reward by finding the best parameters  $\theta$
- Learn policy instead of a value function
  - Q-learning learns a value function



# Policy Gradient Algorithms

- $J(\theta)$  = Expected reward for policy  $\pi_\theta$  with parameters  $\theta$
- Goal: Maximize  $J(\theta)$
- Update policy parameters  $\theta$  using gradient ascent
  - Follow gradient with respect to  $\theta$  ( $\nabla_\theta$ ):

$$\theta := \theta + \alpha \nabla_\theta J(\theta)$$

- REINFORCE
  - Monte Carlo Policy Gradient

$$\theta_{t+1} = \theta_t + \alpha r_t \nabla_\theta \log(\pi_\theta(a_t, s_t))$$

# Policy Gradient Algorithms

- $J(\theta)$  = Expected reward for policy  $\pi_\theta$  with parameters  $\theta$
- Goal: Maximize  $J(\theta)$
- Update policy parameters  $\theta$  using gradient ascent
  - Follow gradient with respect to  $\theta$  ( $\nabla_\theta$ ):

$$\theta := \theta + \alpha \nabla_\theta J(\theta)$$

- REINFORCE
  - Monte Carlo Policy Gradient

$$\theta_{t+1} = \theta_t + \alpha r_t \nabla_\theta \log(\pi_\theta(a_t, s_t))$$

Reward at step t

# REINFORCE with Baseline

- Monte Carlo Policy gradient algorithms suffer from high variance
  - Problem: If  $r_t$  is always positive, probabilities of actions just keep going up
- Rather than update when a reward is **positive** or **negative**, update when a reward is **better** or **worse** than expected
- Baseline:
  - Value to subtract from the reward to reduce variance
  - Estimate the reward  $v_t$  for state  $s_t$  using a **value function**

$$\theta_t = \theta_t + \alpha(r_t - v_t) \nabla_{\theta} \log(\pi_{\theta}(a_t | s_t))$$

# REINFORCE with Baseline

- Monte Carlo Policy gradient algorithms suffer from high variance
  - Problem: If  $r_t$  is always positive, probabilities of actions just keep going up
- Rather than update when a reward is **positive** or **negative**, update when a reward is **better** or **worse** than expected
- Baseline:
  - Value to subtract from the reward to reduce variance
  - Estimate the reward  $v_t$  for state  $s_t$  using a **value function**

$$\theta_t = \theta_t + \alpha (r_t - v_t) \nabla_{\theta} \log(\pi_{\theta}(a_t | s_t))$$

(Reward - Baseline)

# Reformulator: Inputs and Outputs

- Inputs:

- Original query:  $q_o = (w_1, \dots, w_n)$
- Documents from  $q_o$ :  $D_o$
- Candidate term:  $t_i$
- Context terms:  $(t_{i-k}, \dots, t_{i+k})$ 
  - Terms around candidate term to give information on how word is used

- Outputs:

- Probability of using candidate term in new query (**Policy**):  $P(t_i | q_o)$
- Estimated Reward Value (**Baseline**):  $\hat{R}$

# REINFORCE

- Stochastic Objective Function for Policy

$$C_a = (R - \bar{R}) \sum_{t \in T} -\log P(t|q_0),$$

- Value Network Trained to Minimize:

$$C_b = \alpha ||R - \bar{R}||^2,$$

- Minimize using stochastic gradient descent

# Reward

**R = Recall@K**

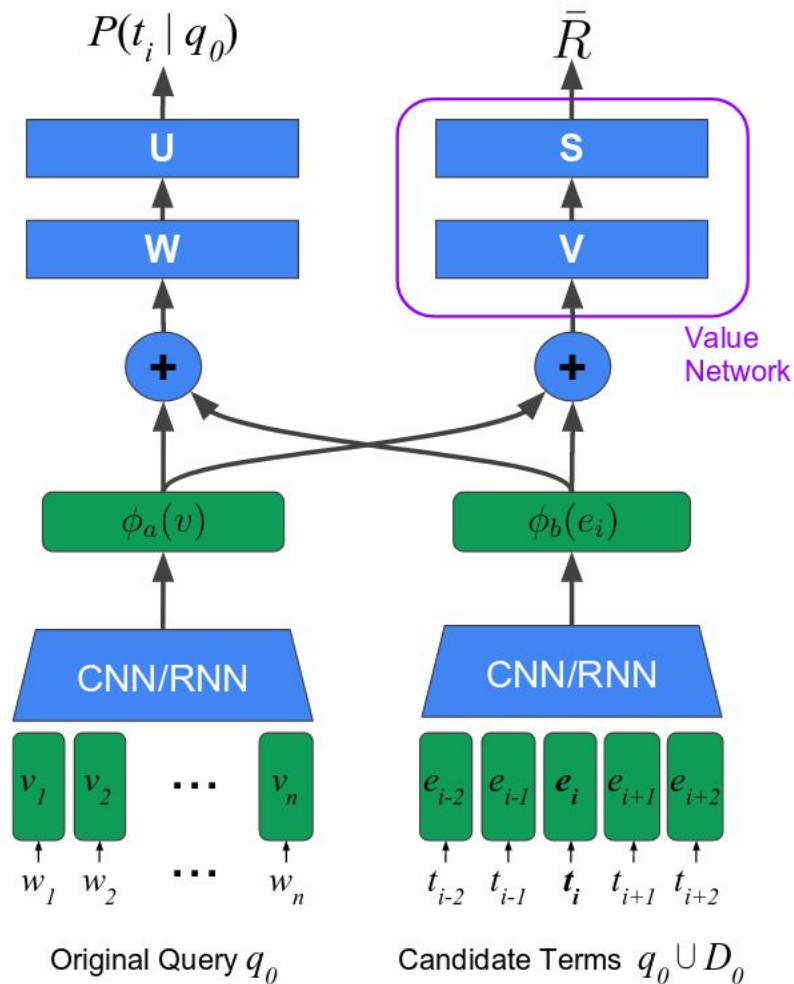
$$\mathbf{R@K} = \frac{|D_K \cap D^*|}{|D^*|}$$

Where  $D_K$  are the top-K retrieved documents and  $D^*$  are the relevant documents

**R@40** used for training reinforcement learning models

# Reformulator: Model

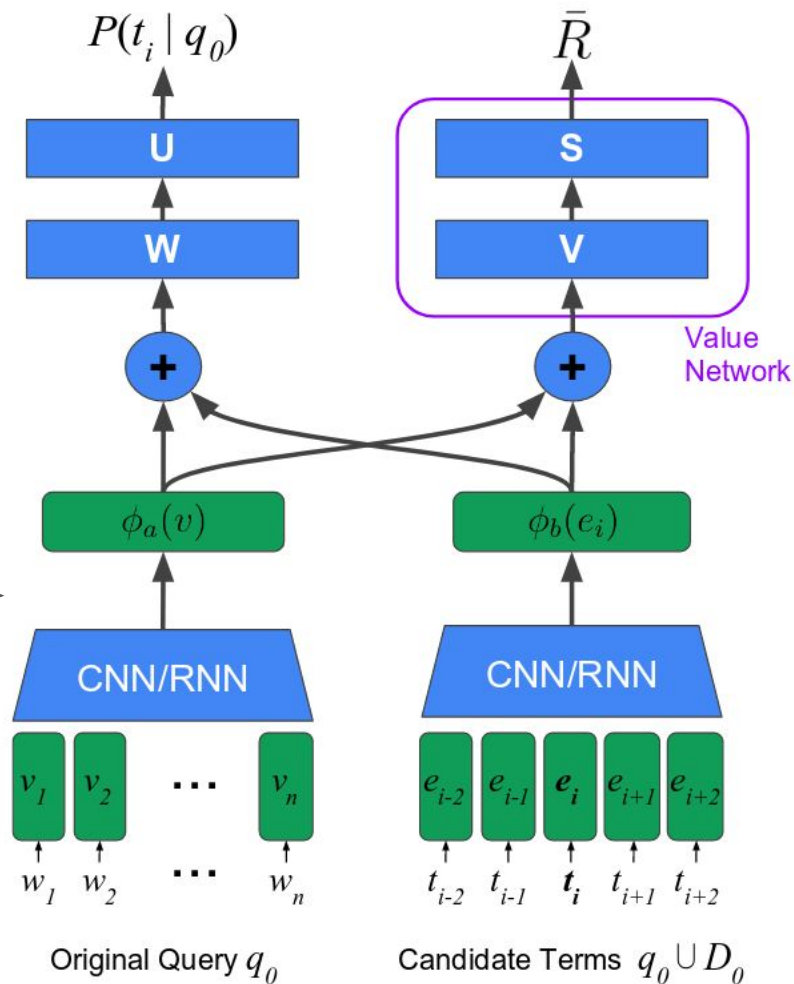
Use Word2vec to convert  
Inputs terms to vector  
representations





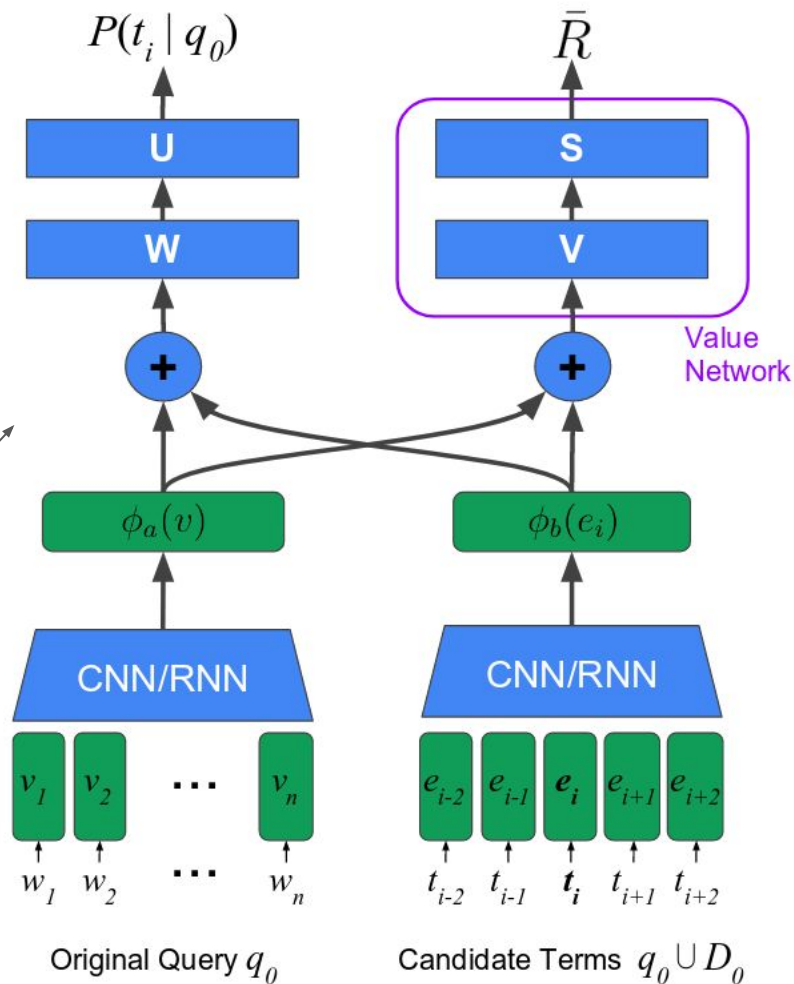
# Reformulator: Model

Use CNN/RNN to create  
fixed length vector outputs



# Reformulator: Model

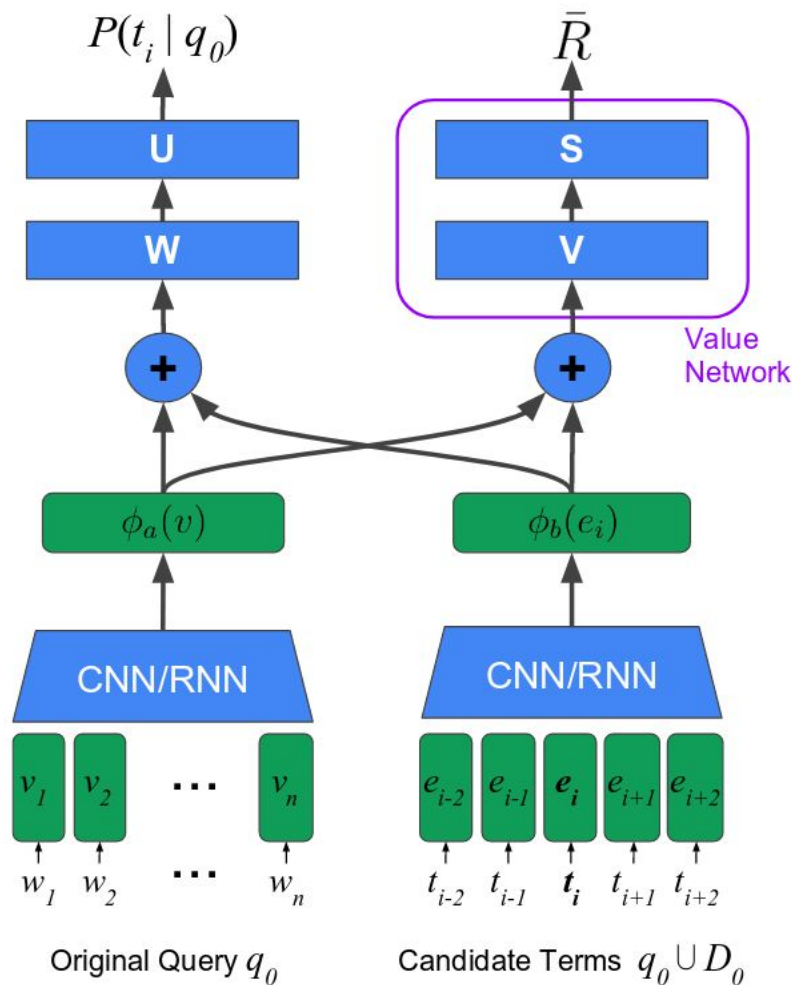
Concatenate outputs from  
original query and  
candidate terms



# Reformulator: Model

$$P(t_i | q_0) = \sigma(U^T \tanh(W(\phi_a(v) \| \phi_b(e_i)) + b))$$

$$\bar{R} = \sigma(S^T \tanh(V(\phi_a(v) \| \bar{e}) + b)),$$



# Reinforcement Learning Extensions

- Sequential model of term addition
  - Produces shorter queries
- Oracle to estimate upper bound on performance for RL methods
  - Split **validation or test** data into N smaller subsets
  - Train an RL agent on each subset until it overfits the subset
  - Average the rewards achieved by each agent on their given subset

# Baseline Method: Supervised Learning

- Assume terms independently affect query results
- Train binary classifier to predict if adding a term to a given query will increase recall
- Add terms that are predicted to increase performance above a threshold

# Experiments: Datasets

- **TREC - Complex Answer Retrieval (TREC-CAR)**
  - Query: wikipedia title and subsection title
  - Relevant Documents: Paragraphs in subsection
- **Jeopardy**
  - Query: A Jeopardy question
  - Relevant Documents: Wikipedia article with title of the answer
- **Microsoft Academic (MSA)**
  - Query: Paper Title
  - Relevant Documents: Papers cited in the original paper

Dataset	Corpus	Docs	Queries			Relevant Docs/Query		Words/Doc	
			Train	Valid	Test	Avg.	Std.	Avg.	Std.
TREC-CAR	Wikipedia Paragraphs	3.5M	585k	195k	195k	3.6	5.7	84	68
Jeopardy	Wikipedia Articles	5.9M	118K	10k	10k	1.0	0.0	462	990
MSA	Academic Papers	480k	270k	20k	20k	17.9	21.5	165	158

# Results

(Original) <i>Daikon</i> <i>Cultivation</i>	<i>“...many types of pickles are made with daikon, includ...”</i> <i>“Certain varieties of daikon can be grown as a winter...”</i> <i>“In Chinese cuisine, turnip cake and chai tow kway...”</i>
(Reformulated) <i>Daikon</i> <i>Cultivation root seed</i> <i>grow fast-growing</i> <i>Chinese leaves</i>	<i>“...many types of pickles are made with daikon, includ...”</i> <i>“Certain varieties of daikon can be grown as a winter...”</i> <i>“The Chinese and Indian varieties tolerate higher....”</i>

# Results

Method	TREC-CAR			Jeopardy			MSA		
	R@40	P@10	MAP@40	R@40	P@10	MAP@40	R@40	P@10	MAP@40
Raw-Lucene	43.6	7.24	19.6	23.4	1.47	7.40	12.9	7.24	3.36
Raw-Google	-	-	-	30.1	1.92	7.71	-	-	-
PRF-TFIDF	44.3	7.31	19.9	29.9	1.91	7.65	13.2	7.27	3.50
PRF-RM	45.1	7.35	19.5	30.5	1.96	7.64	12.3	7.22	3.38
PRF-Emb	44.5	7.32	19.0	30.1	1.92	7.74	12.2	7.22	3.20
Vocab-Emb	44.2	7.30	19.1	29.4	1.87	7.80	12.0	7.21	3.21
SL-FF	44.1	7.29	19.7	30.8	1.95	7.70	13.2	7.28	3.88
SL-CNN	45.3	7.35	19.8	31.1	1.98	7.79	14.0	7.42	3.99
SL-Oracle	50.8	8.25	21.0	38.8	2.50	9.92	17.3	10.12	4.89
RL-FF	44.1	7.29	20.0	31.0	1.98	7.81	13.9	7.33	3.81
RL-CNN	47.3	7.45	20.3	33.4	<b>2.14</b>	8.02	14.9	7.63	4.30
RL-RNN	<b>47.9</b>	<b>7.52</b>	<b>20.6</b>	<b>33.7</b>	2.12	<b>8.07</b>	<b>15.1</b>	<b>7.68</b>	<b>4.35</b>
RL-RNN-SEQ	47.4	7.48	20.3	33.4	2.13	8.01	14.8	7.63	4.27
RL-Oracle	55.9	9.06	23.0	42.4	2.74	10.3	24.6	12.83	6.33

Table 2: Results on Test sets. We use R@40 as a reward to the RL-based models.



# Conclusions

- **RL** methods work the best overall
  - RL-RNN achieves highest scores
  - RL-RNN-SEQ produces shorter queries and is faster
- There is a large gap between best **RL** method and **RL-Oracle**.
  - Shows there is significant room for improvement using RL methods

Questions?

# References

- Rodrigo Nogueira and Kyunghyun Cho. Task-oriented query reformulation with reinforcement learning. In Proceedings of EMNLP, 2017.
- Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine learning
- Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. Cambridge, MA: MIT Press.
- Query Reformulator Github: <https://github.com/nyu-dl/QueryReformulator>
- Slides on paper by authors: <https://github.com/nyu-dl/QueryReformulator/blob/master/Slides.pdf>