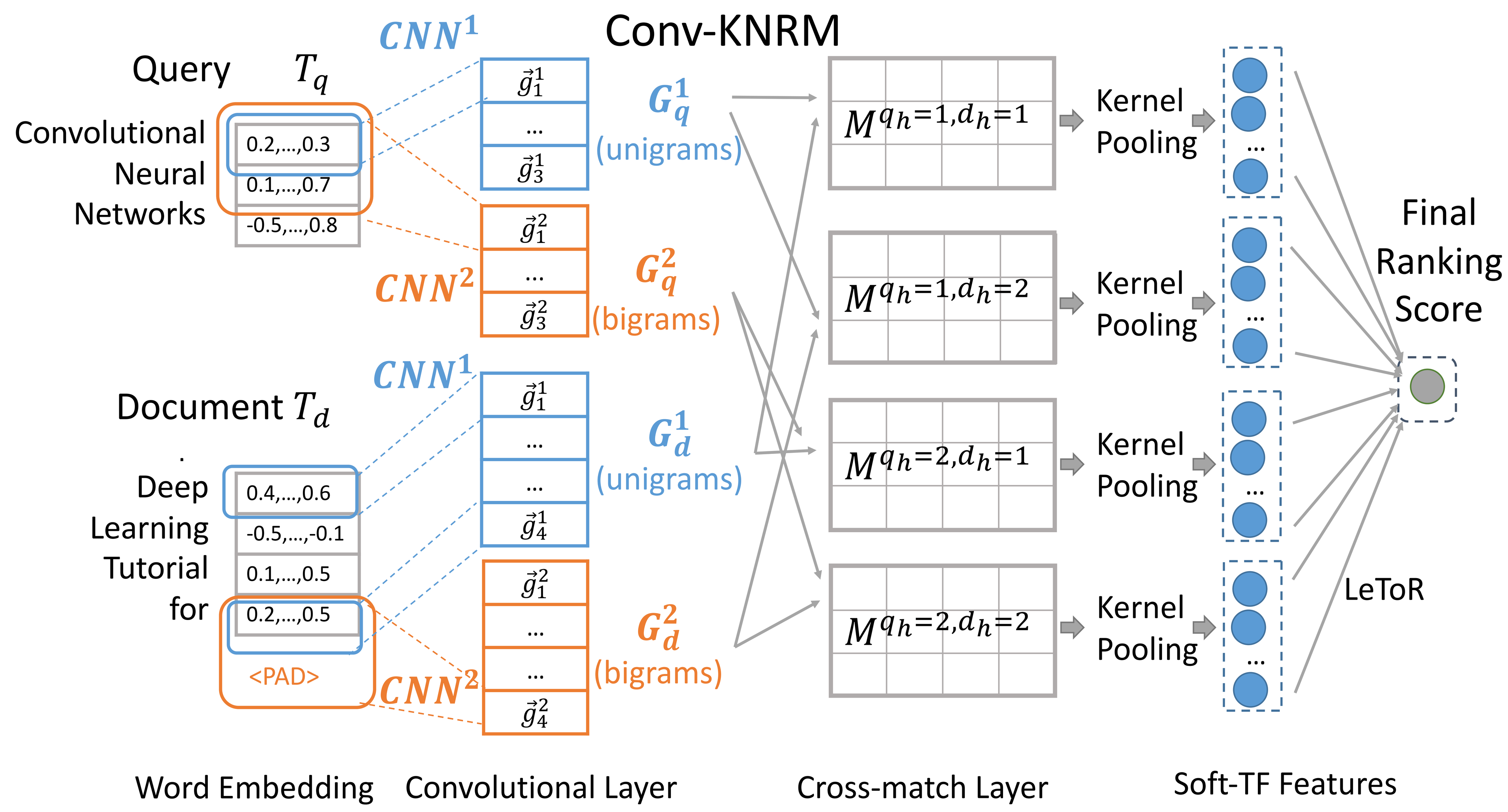
Convolutional Neural Networks for Soft-Matching N-grams in Ad-hoc Search

Zhuyun Dai, Chenyan Xiong, Jamie Callan — Carnegie Mellon University
Zhiyuan Liu — Tsinghua University

Motivation and Background:

- **Queries and documents often match at n-gram level**
  - Query: "*Convolutional Neural Networks*"
  - Document: "*Deep Learning* Tutorial for beginners…"

- **Traditional IR approach: exact match n-grams**
  - Lexical Mismatch Problem
- **Interaction-based Neural IR models**
  - Capture soft match using word embeddings
- **K-NRM: kernel-based neural ranking model**
  - Learns embedding tailored for relevance ranking b end-to-end training from user feedback
  - Soft-match at word level



Conv-KNRM: a neural ranker for soft-matching n-grams

Convolutional Layer
Compose n-gram embeddings from adjacent words' embeddings. Different kernel sizes lead to n-grams of various lengths

Cross-Match Layer
Build similarity matrices between n-grams of different lengths.
E.g. query unigrams to document bigrams

Kernel-Pooling
`Count' soft-match pairs at multiple similarity levels using Gaussian Kernels
E.g. exact match, strong match, weak match…

Pairwise Learning
Learn to differentiate relevant documents from irrelevant ones.
$Loss = max(0, 1 - (f(q, d^+) - f(q, d^-)))$

Domain Adaptation

- Source Domain: large number of training data and labels
- Target Domain: limited amount of training data
- Learn soft-match on source domain; Tune LeToR weights on target domain

Source Domain(End-to-End)		**Target Domain**(Adapt)
• Learn Word Embedding	→	• Apply Word Embedding
• Learn CNN	→	• Apply CNN
• Learn LeToR weights		• Learn LeToR weights

Experiments

End-to-End Learning

- **Sogou-Log**: Chines **Bing-Log:** English
- Both have ~100K training queries, 1K testing queries
- Train on relevance labels estimated by a click model (DCTR)
- Test on 1) click model labels and 2) raw user clicks

Domain Adaptation

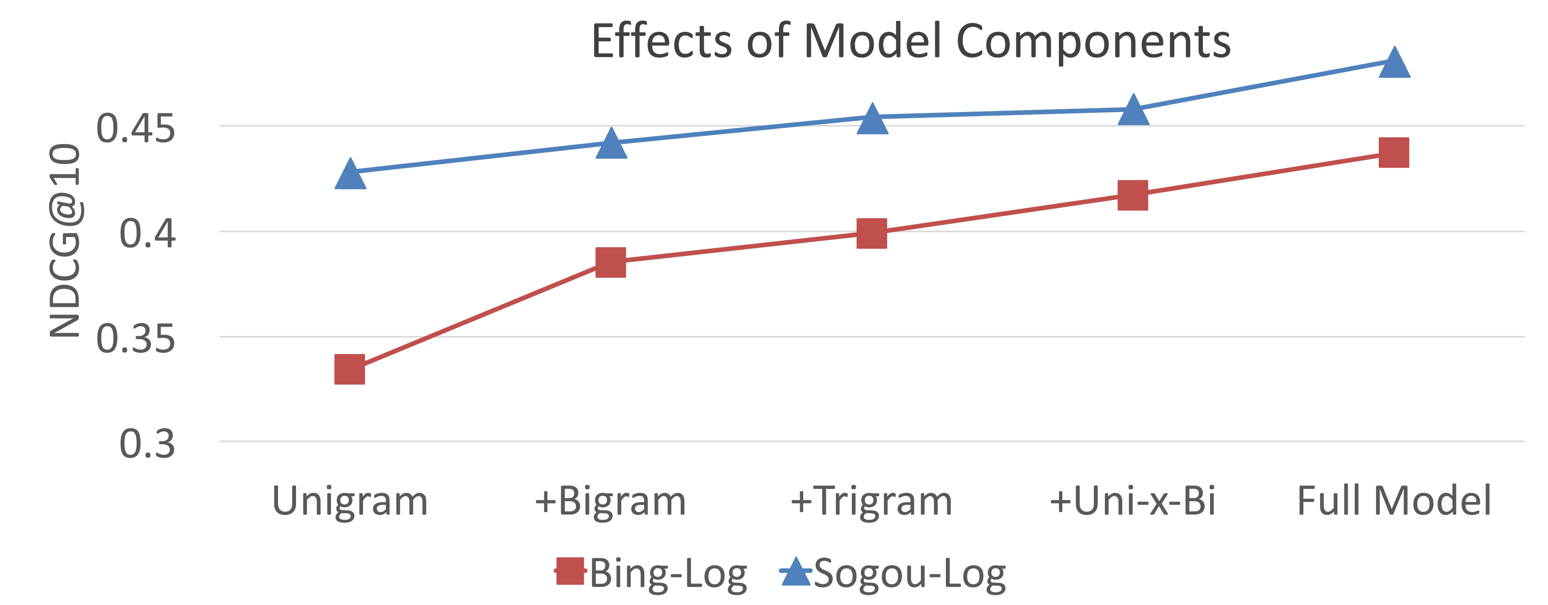
- **Bing-Log** => **Clueweb09B** (TREC 2009-2012, 200 queries)
- Learn LeToR weights on Clueweb09B using 10-fold CV

Experimental Results

End-to-End Learning

	Sogou-Log				Bing-Log			
Method	NDCG@1		NDCG@10		NDCG@1		NDCG@10	
BM25	0.142	-45%	0.287	-34%	0.043	-79%	0.123	-63%
RankSVM	0.146	-44%	0.309	-29%	0.128	-39%	0.266	-20%
Coor-Ascent	0.169	-34%	0.355	-16%	0.142	-32%	0.268	-20%
DRMM	0.137	-51%	0.315	-27%	0.137	-34%	0.247	-26%
CDSSM	0.144	-44%	0.333	-23%	0.156	-25%	0.273	-18%
MP	0.218	-15%	0.379	-12%	0.182	-12%	0.301	-10%
K-NRM	0.264	--	0.428	--	0.208	--	0.334	--
Conv-KNRM	**0.336**	**+30%**	**0.481**	**+11%**	**0.300**	**+44%**	**0.437**	**+31%**

Performance on Testing-SAME (DCTR click model for train & test)



- Conv-KNRM outperformed previous state-of-the-arts
- N-gram has higher gain on English search log. Chinese unigrams have phrase-like characteristics
- Cross-matching is the key

Domain Adaptation

	Clueweb09B	
Method	NDCG@1	NDCG@20
Galago-SDM	0.219	0.250
RankSVM	0.236	0.263
Coor-Ascent	0.255	0.268
DRMM+SDM	0.215	0.269
K-NRM-adapt	0.235	0.270
Conv-KNRM-adapt	**0.294**	**0.287**

Performance on Clueweb09B

- The n-gram soft-match features learned from Bing-Log are also effective on Clueweb09B

Query	Matched N-gram
sewing instructions	*quilting 101*
fickle creek *farm*	*eat & drink*
atypical squamous cells	*cervical cancer*
wedding budget calculator	*perfect planner tools*

Case study: example of matched n-grams

- The matchings make sense in various contexts than just in one dataset
- Soft-match overcomes lexical mismatch
- Matches n-grams whose individual word may not match

Conclusions

- Conv-KNRM: uses CNNs to compose n-gram embeddings from word embeddings, and cross-matches n-grams of various lengths
- IR-customized n-gram soft match: Learns n-gram soft match patterns tailored for relevance matching with kernel pooling
- Cross-matching: cross-matching is important because related concepts do not necessarily have the same number of words
- Generalizable: model trained on one domain is generalizable to a related search domain. Beat strong feature-based LeToR baseline on TREC