Learning to Rank Entities for Set Expansion from Unstructured Data

Puxuan Yu, Razieh Rahimi, Zhiqi Huang, and James Allan

Center for Intelligent Information Retrieval

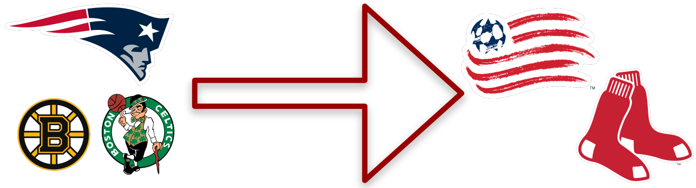
University of Massachusetts Amherst

Email: pxyu@cs.umass.edu

Slides: cs.umass.edu/~pxyu/public/pdf/nese-ictir20.pdf

Task Definition

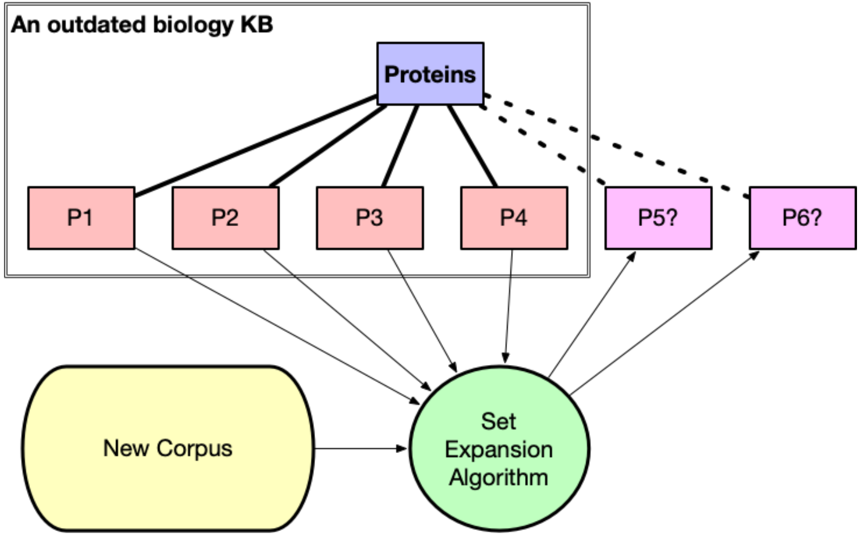
- Set expansion
  - Extract sibling entities to user-given entities
  - Dates back to "Google Sets" (user-oriented)
    - also, QA, query suggestion, …
- Restricted setting: corpus-based set expansion
  - Extract sibling entities from plain text, no access to KB at **inference time**
  - Applicable to more downstream tasks
    - relation extraction, taxonomy construction, knowledge base completion, …



"Major league sports teams in Massachusetts"

Task Definition

- Restricted setting: corpus-based set expansion
  - Example use case: knowledge base completion (KBC)



Challenges & Contributions

- We aim to train a **neural model** for set expansion
  - Training data (set labels): a series of entity sets that are also in the corpus
    - Developed a toolkit (**DBpedia-sets**) to build labelled sets from corpus via knowledge base at training time
  - Entity features from raw corpus: unigrams / skip-grams / embeddings
    - Empirical analysis of unigram vs skip-gram
    - Unigram + linearly mapped embeddings
  - Modeling and learnable parameters
    - Query-candidate interactions / query length-agnostic / generalizability

Training data: DBpedia-sets

Steps:

1. Identify entity mentions (entity linking)
2. Collect entity statistics
3. Filter entity sets

An example statistical filter: “find all entity sets containing 10 to 100 entities, where at least 90% of the entities appear at least 10 times in the corpus”

Training data: DBpedia-sets

Advantages:

- Entity sets are topically diverse
- Sets are of high quality
- Mined sets are dependent on corpus
  - In contrast, INEX Entity track and DBPedia-Entity-V2 are not quite usable
    - Hard for training and evaluation!

Entity features

- Combination of lexical features and distributed representations (embeddings) [1]
- Unigram PPMI:
  - Each dimension corresponds to a word in corpus

$$S_{ij} = \max(\mathrm{PMI}(e_i, u_j), 0),$$
$$\mathrm{PMI}(e, u) = \log \frac{P(e,u)}{P(e)P(u)} = \log \frac{\mathrm{freq}(e,u)|\mathrm{corpus}|}{\mathrm{freq}(e)\mathrm{freq}(u)}$$

- Entity embeddings: No graph embeddings! (no access to KB at inference time)
  - Treat entity as word, and get its word2vec/GloVe embeddings [1]
  - Or use contextualized representations (e.g., BERT)

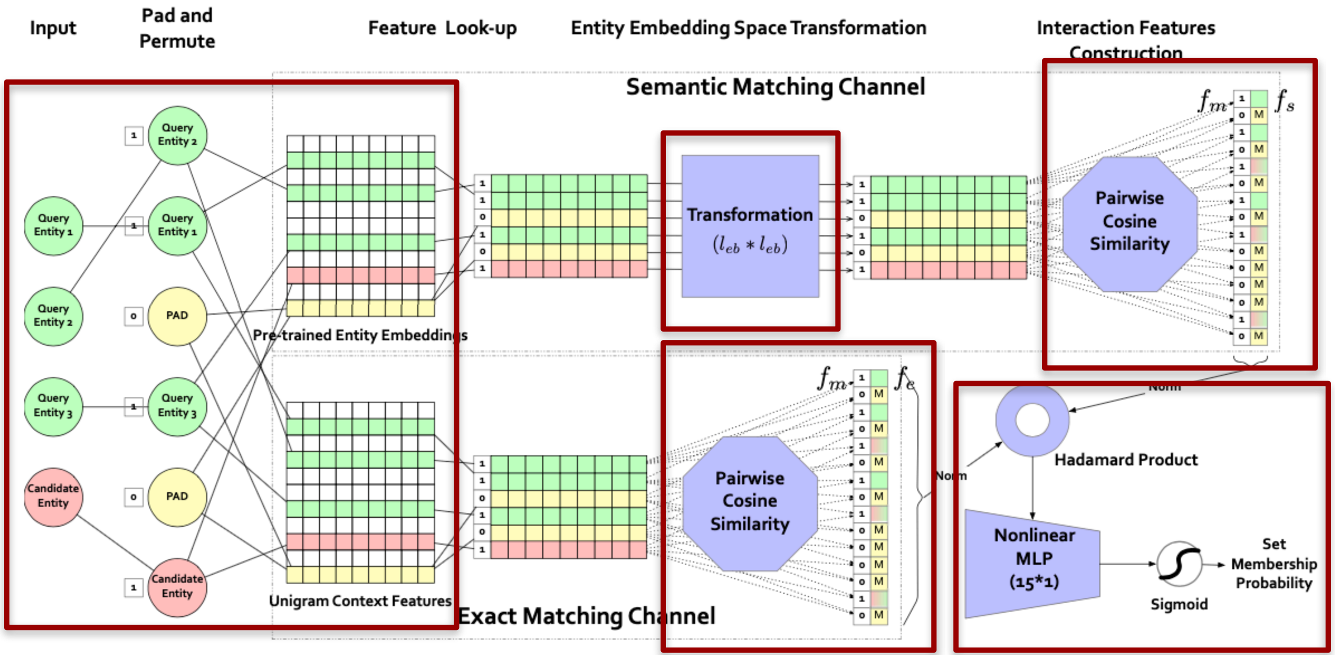
[1] Yu et al. "Corpus-based Set Expansion with Lexical Features and Distributed Representations." SIGIR 2019.

Entity features

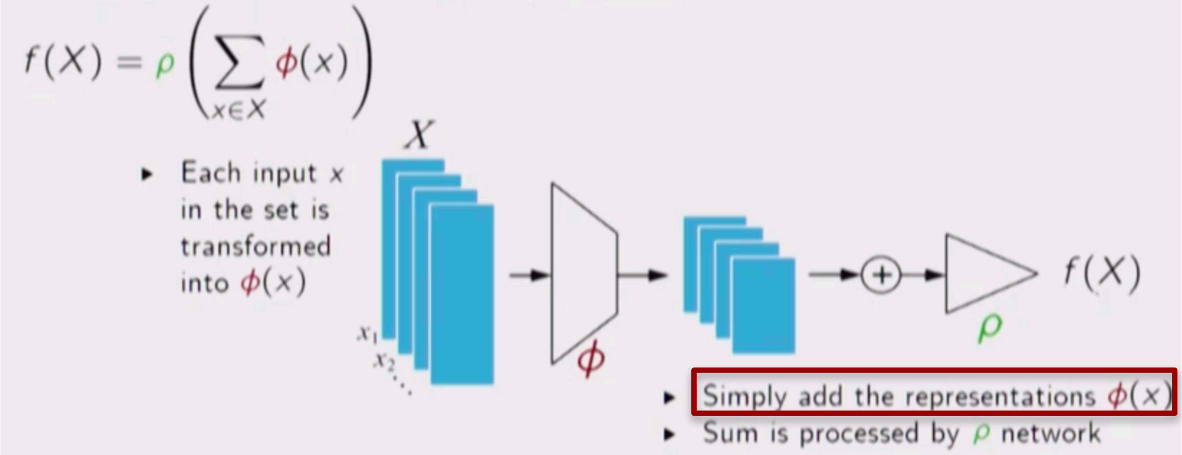
- BERT for set expansion [2]
    - One-instance: average of embeddings of entity tokens in a sentence
    - Corpus-wise: average of all one-instance embeddings in the corpus
- No finetuning performed. Acquire embeddings from BERT directly.
- Linear mapping of entity embeddings is a “hacky” way of finetuning BERT! (Later)

[2] Huang et al. "Guiding Corpus-based Set Expansion by Auxiliary Sets Generation and Co-Expansion." WWW 2020.

Neural model



Neural model



Structure of Deep Sets [3], figure from the NIPS presentation

[3] Zaheer et al. "Deep sets." NIPS 2017.

Experiments

- Setup:
  - Re-ranking top-100 candidates
  - List-wise loss: Listnet
- Candidate generation: best unsupervised approach (recall) on each corpus
- A non-neural supervised approach: AdaRank
  - Cannot do padding, have to train a model for each query length
  - No linear mapping of entity embeddings
- Metrics: MAP and P@20

Experiments

- uni: using only unigram features
- emb: using only embedding features
- cmb: combing features from uni and emb
- nt: without entity embedding linear transformation

Dataset	**AP89** (34.8M tokens)						**WaPo** (395M tokens)						**Wiki** (928M tokens)					
Metrics	MAP@100			P@20			MAP@100			P@20			MAP@100			P@20		
Query length	3	4	5	3	4	5	3	4	5	3	4	5	3	4	5	3	4	5
GloVe	.110	.123	.128	.101	.108	.104	.167	.179	.183	.161	.160	.157	.231	.250	.259	.214	.216	.210
BERT	.262	.270	.267	.227	.212	.208	.235	.234	.239	.211	.203	.196	.180	.186	.187	.174	.169	.161
SetExpan	.154	.153	.153	.120	.121	.119	.171	.172	.165	.172	.168	.162	.220	.217	.217	.201	. 195	.188
CaSE-skip	.174	.183	.183	.148	.147	.143	.206	.196	.196	.205	.188	.184	.249	.248	.248	.227	.216	.205
CaSE-uni	.168	.181	.179	.152	.153	.146	.204	.195	.195	.200	.185	.180	.254	.253	.254	.231	.219	.208
AdaRank-uni	.223	.245	.256	.217	.220†	.217†	.238	.240	.247†	.226†	.223†	.218†	.213	.264†	.267†	.206	.237†	.230†
AdaRank-emb	.227	.245	.259	.217	.210	.203	.235	.232	.238	.211	.202	.197	.260	.273†	.282†	.239†	.237†	.232†
AdaRank-cmb	.227	.246	.256	.218	.220†	.215	.238	.242†	.247†	.226†	.225†	.219†	.259	.270†	.280†	.239†	.236†	.230†
NESE-uni	.241	.252	.256	.230	.227†	.220†	.242	.246†	.248†	.232†	.228†	.218†	.249	.264†	.268†	.240†	.237†	.231†
NESE-emb-nt	.236	.250	.261	.207	.200	.201	.225	.230	.236	.202	.197	.193	.261	.273†	.281†	.239†	.238†	.230†
NESE-emb	.206	.206	.212	.192	.182	.178	.217	.217	.222	.201	.196	.192	.217	.228	.235	.213	.210	.203
NESE-nt	.244	.253	.277†	.231	.226	.224†	.246†	.248†	.266†	.234†	.229†	.224†	.260	.270†	.281†	.239†	.240†	.232†
NESE	**.273†**	**.283†**	**.291†**	**.240†**	**.237†**	**.231†**	**.264†**	**.268†**	**.282†**	**.253†**	**.247†**	**.240†**	**.272†**	**.288†**	**.293†**	**.252†**	**.246†**	**.239†**
Δ	+4.2%	+4.8%	+9.0%	+5.7%	+11.8%	+11.1%	+12.3%	+14.5%	+18.0%	+19.9%	+21.7%	+22.4%	+7.1%	+12.8%	+15.4%	+9.1%	+ 12.3%	+14.9%

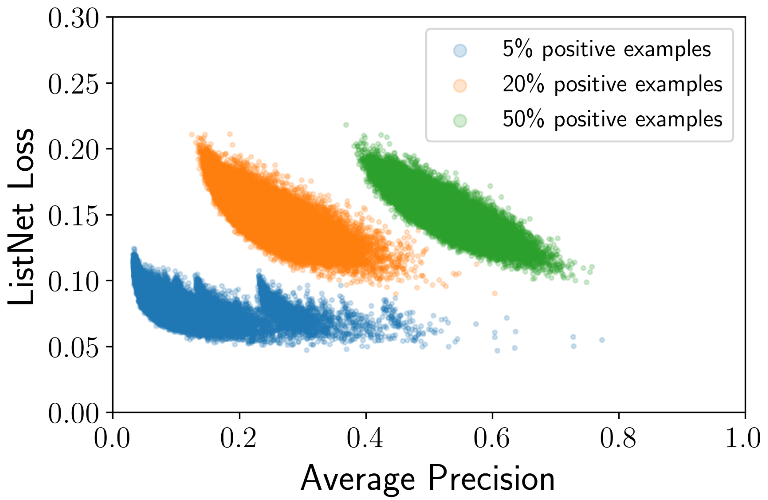
Generalizability

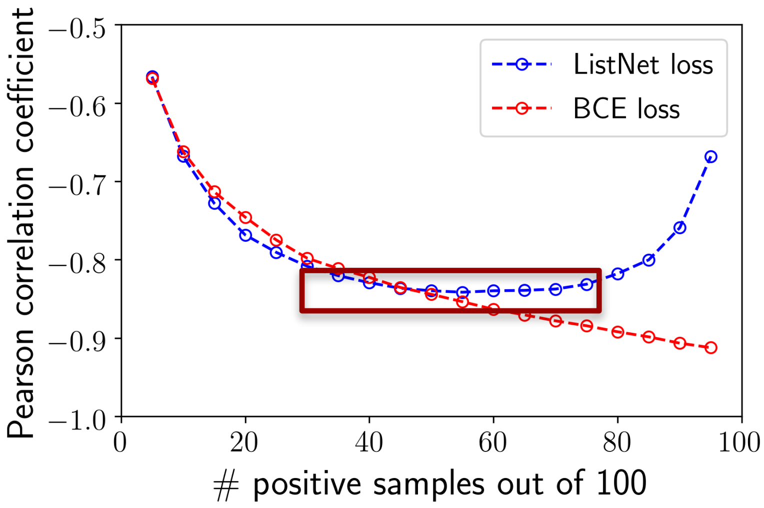
- Models trained on DBpedia-sets data perform well on DBpedia-sets test data (✓)
- Models trained on DBpedia-sets data perform well on non-KB entities
  - Use noun phrases as approximation to entity mentions
  - They also have context features and entity embeddings (compatible w/ our model!)
  - Hard to evaluate: same entity have different surface names
    - Tottenham Hotspur F.C.: "Tottenham Hotspur", "Tottenham", "the spurs", ……
    - We adopt small-scale human evaluation

Sets	NBA teams		TV channels		European capitals	
Query	q1	q2	q3	q4	q5	q6
GloVe	.625	.673	.059	.125	.050	.313
CaSE-uni	.656	.647	.178	.254	.524	.454
NESE	**.733**	**.733**	**.254**	**.313**	**.551**	**.524**

Side Experiments: list-wise learning to rank

- Relation between “correlation of Listnet loss and MAP” and “ratio of positive docs / entities”





Summary

- Cast corpus-based set expansion as list-wise learning-to-rank
- Corpus-dependent dataset for training set expansion models
- Linearly mapping entity embeddings + unigram PPMI features bring significant improvement

Future work:

- Better ways of using BERT?
    - Source sentence selection / weighting to generated “query-dependent” contextualized entity embeddings

UMassAmherst | College of Information & Computer Sciences
Center for Intelligent Information Retrieval

Puxuan Yu

cs.umass.edu/~pxyu

pxyu@cs.umass.edu