Character-level Deep Conflation for Business Data Analytics

Presented by Xiaodong He

Joint work with Zhe Gan, P. D. Singh, Ameet Joshi, Jianshu Chen, Jianfeng Gao, Li Deng

Microsoft Research & Duke University

The Problem of Conflation

Conflation: connecting different text attributes associated with the same entity, so as to merge two or more tables.

- Of great importance in business data analytics

Entity	Attr. 1	Attr. 2
Emilio yentsch	111	222
enrique hafner	333	444

Entity	Attr. A	Attr. B
Mr. halner exrique	aaa	bbb
ydntsch emilip	ccc	ddd

Entity	Attr. 1	Attr. 2	Attr. A	Attr. B
emilio yentsch	111	222	ccc	ddd
enrique hafner	333	444	aaa	bbb

The Major Challenges

Entity in Table A	Entity in Table B
emilio yentsch	ydntsch emilip
enrique hafner	Mr. halner exrique
javier creswell	Prof. crrxwell javzfr

- Irregular vocabulary and frequent misspelling
- Non-monotonic word ordering, plus ins/del
- Very short context, weak language info.

Traditional methods

- Rule based entity name matching
  - Only suitable for very limited variations

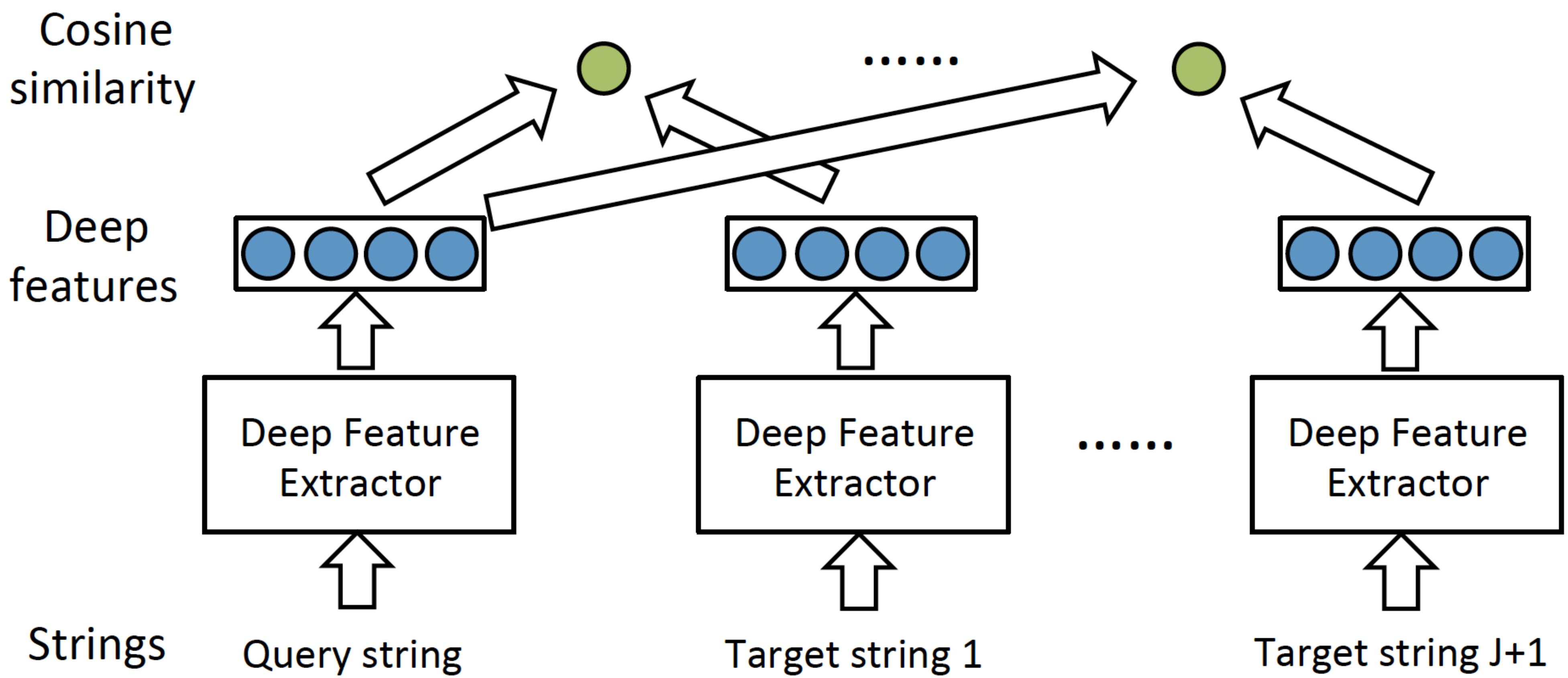
- Statistical model based entity name matching
  - Using hand-crafted features
  - Using various distance metrics, e.g., Hamming distance

- Success is limited

Deep Conflation Models (DCM)

- We proposed new Deep Conflation Models
  - Motivated by the Deep Structured Semantic Model (DSSM)
  - Character-level models, each entity name is represented as a sequence of characters.
  - Using neural networks to project an entity name into a continuous semantic vector.
  - Treat the conflation problem as a ranking problem
    - Train the model so that the correct one ranked the closest to the query entity, when measured by similarity in the vector space.

Architecture of the DCM



Inspired by the Deep Structured Semantic Model [Huang, He, Gao, Deng, Heck, Acero, CIKM2013]

Vocabulary – Just the alphabet!

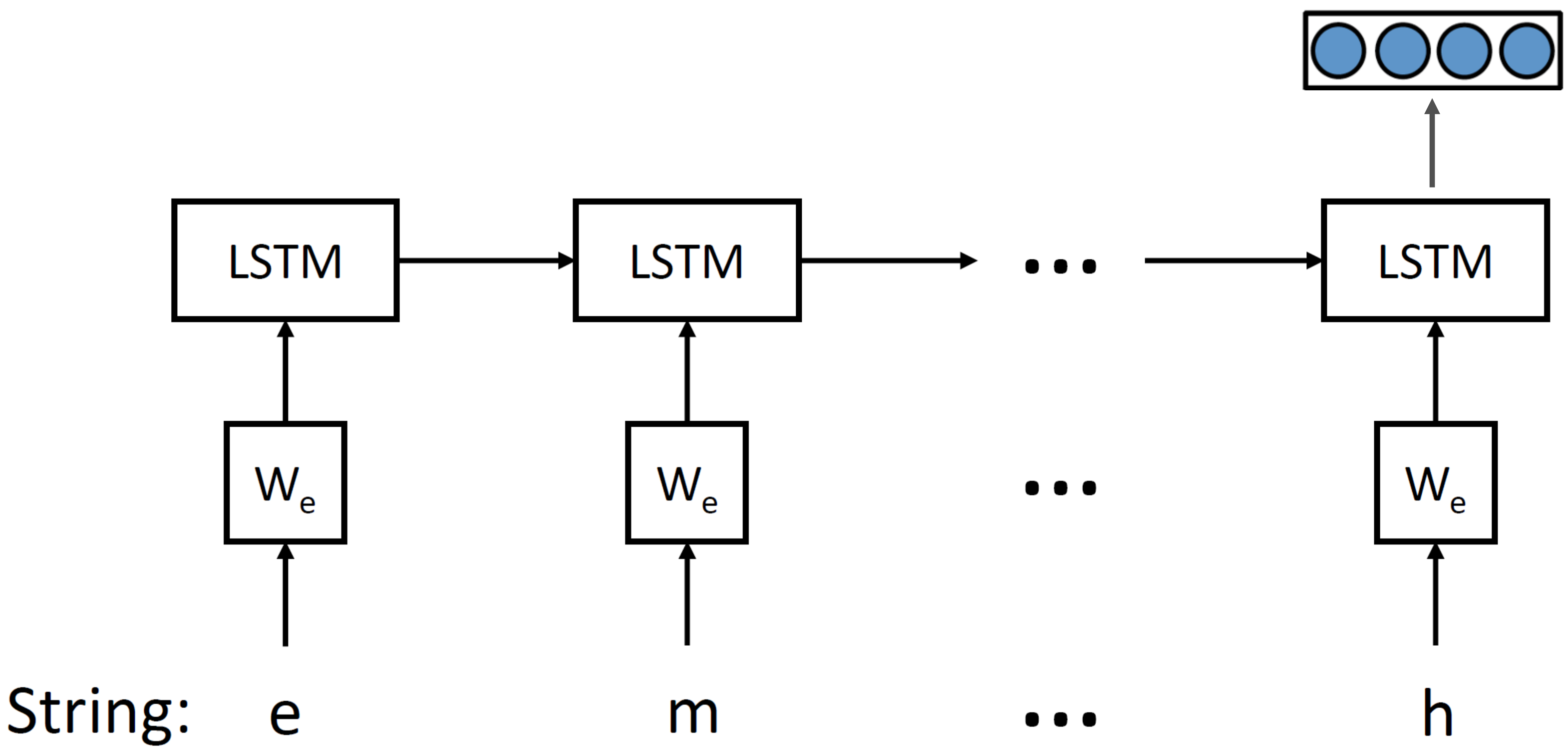
We include only the following 32 characters in the vocabulary

- sufficient for the Deep Conflation Model

`DMPSabcdefghijklmnopqrstuvwxyz.`

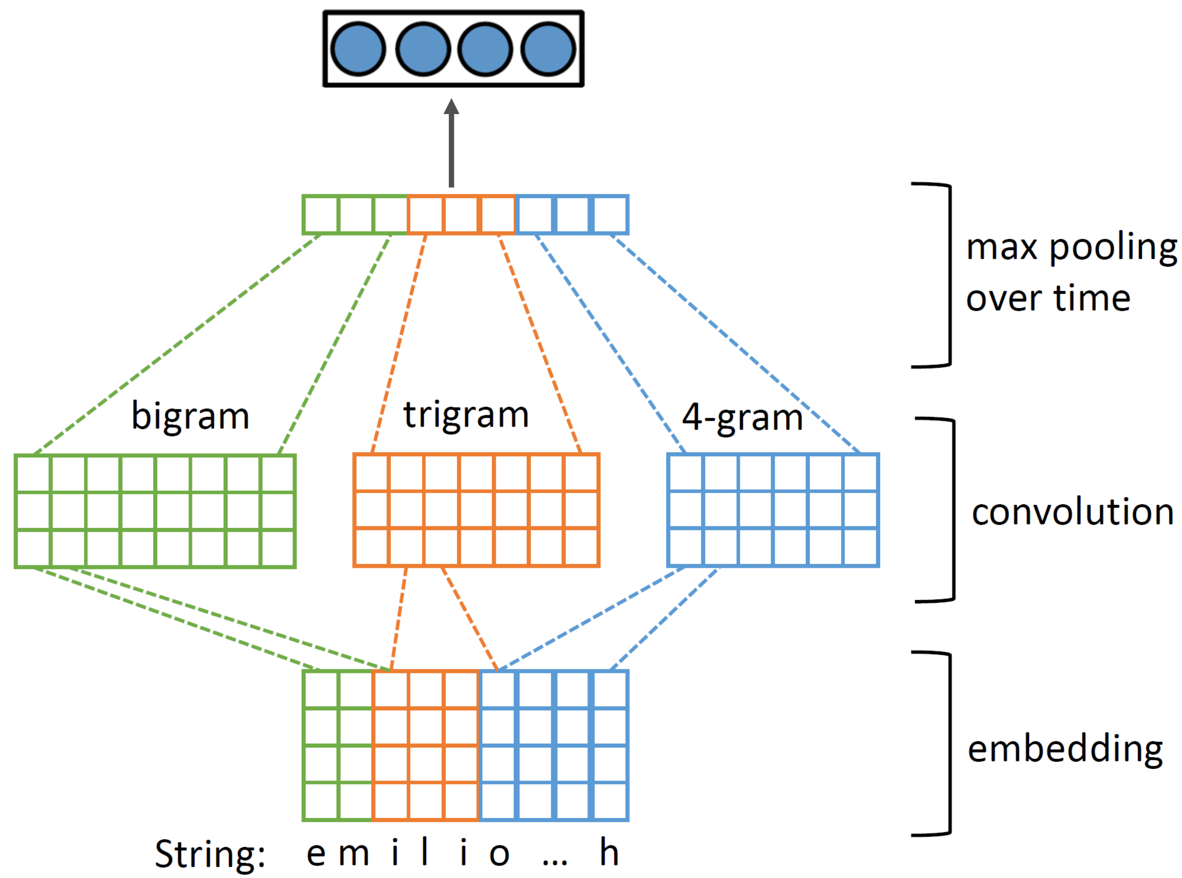
LSTM-based Deep Feature Generator

Code the string of the entity name
into a vector using a LSTM



CNN-based Deep Feature Generator

As an alternative, code the string of the entity name into a vector using a CNN



Design the learning objective

The semantic relevance score between a query and a reference is

$$R(Q, D_j) = \frac{\boldsymbol{y}_Q^\top \boldsymbol{y}_{D_j}}{||\boldsymbol{y}_Q|| \cdot ||\boldsymbol{y}_{D_j}||} ,$$

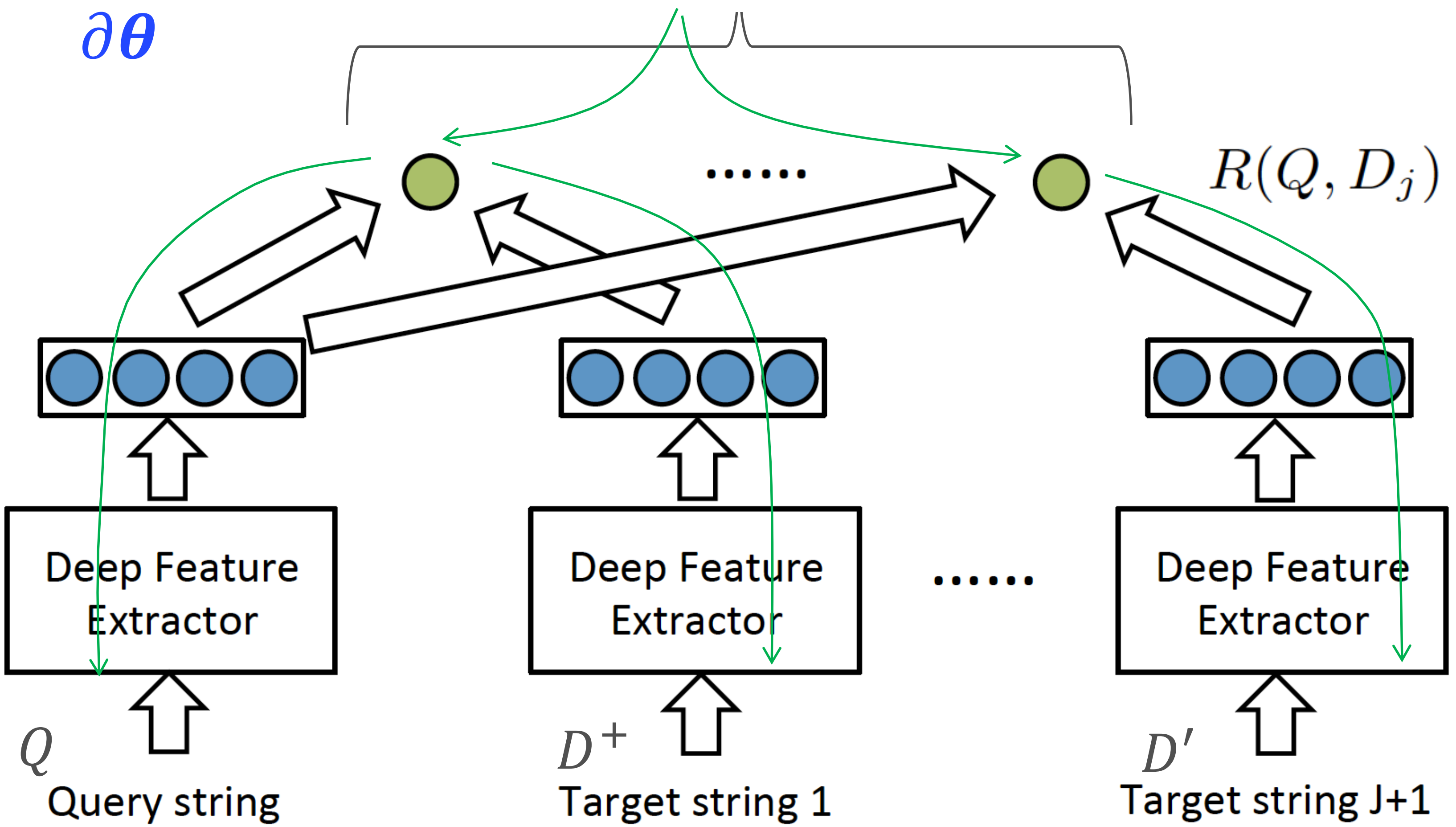
In training, we maximize the probability of the reference given the query:

$$P(D^+|Q) = \frac{\exp(\gamma R(Q, D^+))}{\sum_{D' \in \mathbf{D}} \exp(\gamma R(Q, D'))} ,$$

Learning the DCM by SGD

$$\boldsymbol{\theta}^{t} = \boldsymbol{\theta}^{t-1} + \lambda \frac{\partial P(D^{+}|Q)}{\partial \boldsymbol{\theta}}$$

$$P(D^{+}|Q) = \frac{\exp(\gamma R(Q, D^{+}))}{\sum_{D' \in \mathbf{D}} \exp(\gamma R(Q, D'))}$$



Evaluation

Experimental setting

Training set:

- 8,000 pairs of entity names

Validation set:

- 1,000 pairs of entity names

Test set:

- 1,000 pairs of entity names

Results

Comparison of various feature generators

CNN-based feature generator is most effective

- the local (regional) sequential order information (captured by CNN) is more important than the global sequential order information (captured by LSTM) in matching two names.

Model	**R@1**	**R@3**	**R@10**	**Med r**	**Mean r**	**Harmonic Mean r**
Using correct names to query mis-spelled names						
BoC	82.09± 1.59	92.30± 0.76	96.83± 0.36	1.0± 0.0	2.380± 0.218	1.138± 0.009
LSTM	86.66± 0.90	95.38± 0.53	98.54± 0.20	1.0± 0.0	1.609± 0.092	1.095± 0.007
CNN	98.90± 0.18	99.97± 0.05	100.00± 0.00	1.0± 0.0	1.012± 0.003	1.006± 0.001
Using mis-spelled names to query correct names						
BoC	83.56± 1.42	93.06± 0.80	97.35± 0.27	1.0± 0.0	2.158± 0.128	1.131± 0.011
LSTM	87.63± 0.92	95.50± 0.45	98.67± 0.21	1.0± 0.0	1.584± 0.055	1.088± 0.007
CNN	99.25± 0.43	99.98± 0.06	100.00± 0.00	1.0± 0.0	1.008± 0.005	1.004± 0.002

Results

Range of cosine similarity scores for correct and wrong matching (CNN-DCM)

Table 3: Average scores for each of the top four retrieved items.

top 1	**top 2**	**top 3**	**top 4**
0.792± 0.086	0.448± 0.072	0.397±0.050	0.371±0.042

- Set the threshold to be 0.62 (median between top 1 and top 2).
- When the similarity score between two strings is higher than 0.62, we can safely conflate the entities.

Case study

Table 4: An example of the mistakenly retrieved cases.

query string	palmer mehaffey	
ground truth	Mr mehaffep paleer	score
1st result	paleer mehaffep	0.882
2nd result	Mr mehaffep paleer	0.877
3rd result	fendlasyn pdlmer	0.427
4th result	zalwzar sharley	0.420

Conclusion

We propose a character-level deep conflation model for business data analytics

- The model is extremely compact. It solves three problems:
    1. spelling check for irregular words
    2. Handling non-monotonic word ordering
    3. Working with very short context

Using CNN as feature extractors perform the best for entity name conflation