GNEM: A Generic One-to-Set Neural Entity Matching Framework

Runjin Chen
chenrunjin@sjtu.edu.cn
Shanghai Jiao Tong University

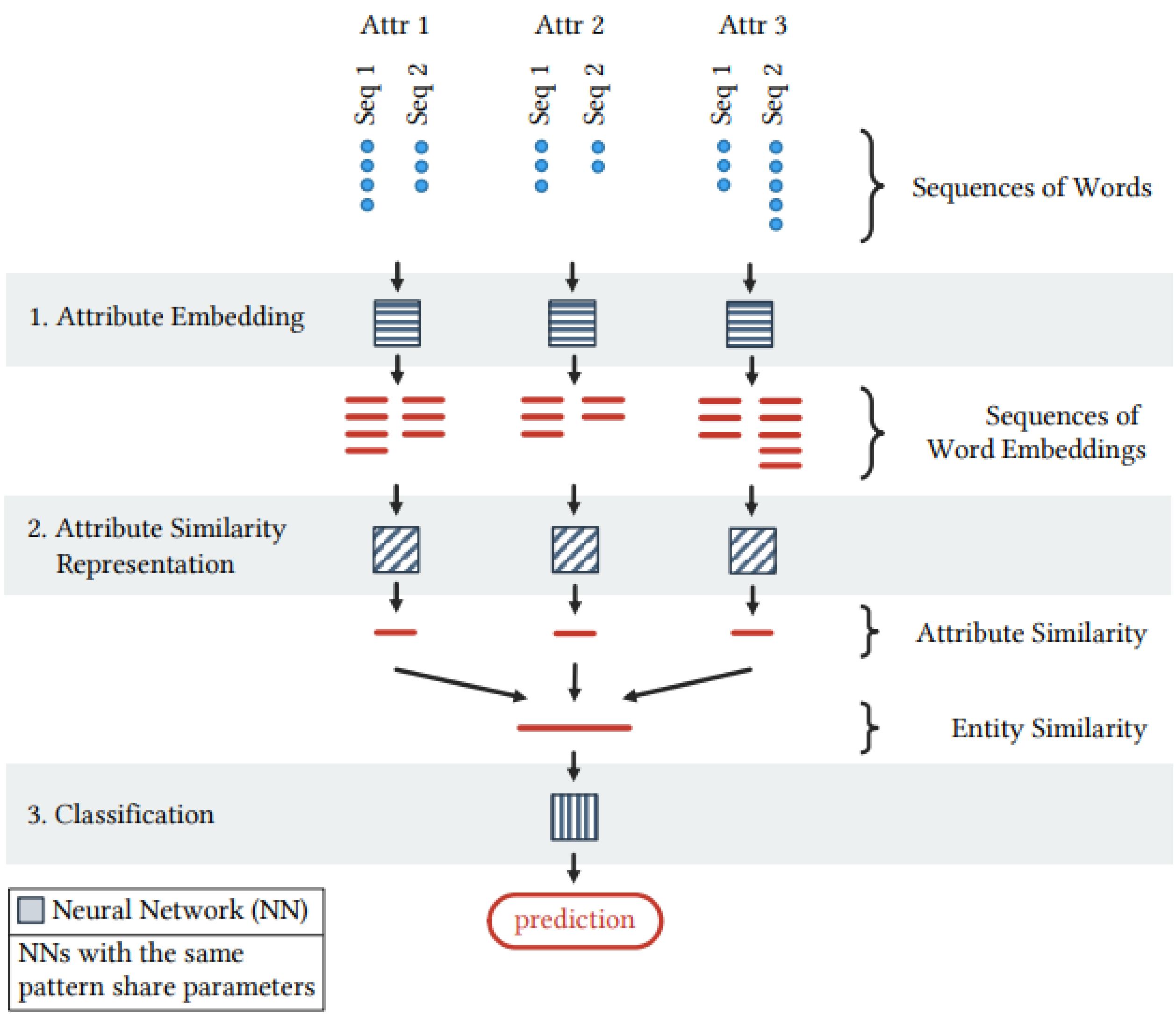
Yanyan Shen*
shenyy@sjtu.edu.cn
Shanghai Jiao Tong University

Dongxiang Zhang
zhangdongxiang@zju.edu.cn
Zhejiang University

WWW2021

Matching two tables typically consists of the following three steps:

1. **Reading the input tables**
2. **Blocking the input tables to get a candidate set**
3. **Matching the tuple pairs in the candidate set**


Attr 1
Attr 2
Attr 3
Seq 1
Seq 2
Sequences of Words
1. Attribute Embedding
Sequences of Word Embeddings
2. Attribute Similarity Representation
Attribute Similarity
Entity Similarity
3. Classification
prediction
Neural Network (NN)
NNs with the same pattern share parameters

Motivation

Table 1: Motivating Examples for One-to-Set EM

(a) (r_1^a, r_1^b) matched $\wedge$ (r_1^b, r_2^b) matched $\Rightarrow$ (r_1^a, r_2^b) matched

No.	Name	Gender	City	Occupation	$(r_1^a, \cdot)$
r_1^a	John Smith	female	Seattle, Washington	–	–
r_1^b	J. Smith	female	Seattle, Washington	teacher	*matched*
r_2^b	J. Smith	–	Seattle, WA	teacher	*matched*

(b) (r_2^a, r_3^b) unmatched $\wedge$ (r_3^b, r_4^b) matched $\Rightarrow$ (r_2^a, r_4^b) unmatched

No.	Title	Artist	Genre	Tag	$(r_2^a, \cdot)$
r_2^a	My Love	Westlife	pop	Band, Heart Touching	–
r_3^b	My Love	Sia	Indie rock	OST, Heart Touching	*unmatched*
r_4^b	My Love	–	–	OST, Heart Touching	*unmatched*

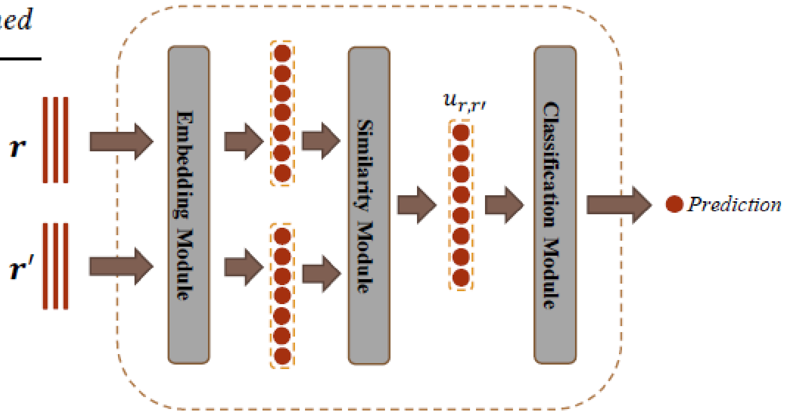


Figure 1: Pairwise EM neural methods.

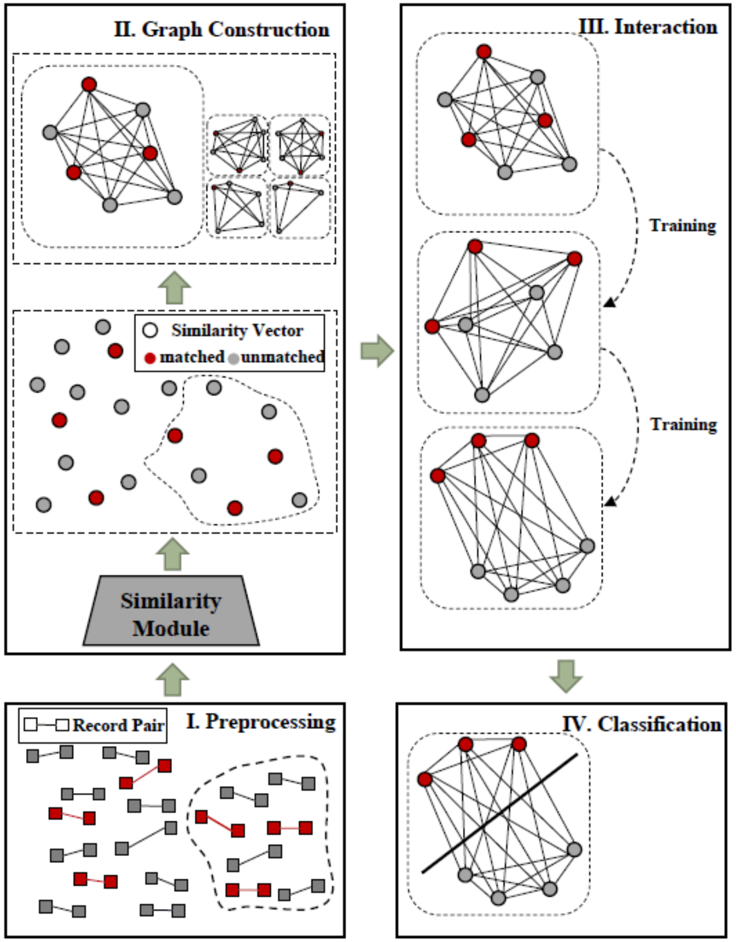


Figure 2: Framework of GNEM.

Preprocessing:

obtain a candidate set $\mathrm{C} \subseteq \mathrm{R}^{\mathrm{a}} \times \mathrm{R}^{\mathrm{b}}$

To derive the one-to-set matching instances, for each record r in $\mathrm{R}^{\mathrm{a}} \cup \mathrm{R}^{\mathrm{b}}$, we retrieve all the pairs in C involving r, i.e., $\{(r_i, r_j) \in \mathrm{C} \mid r_i = r \vee r_j = r\}$. We now recognize the set V_r of records that are relevant to r as follows:

$$V_r = \{r' \mid (r, r') \in \mathrm{C} \vee (r', r) \in \mathrm{C}\} \quad (3)$$

a one-to-set matching instance: (r, V_r)

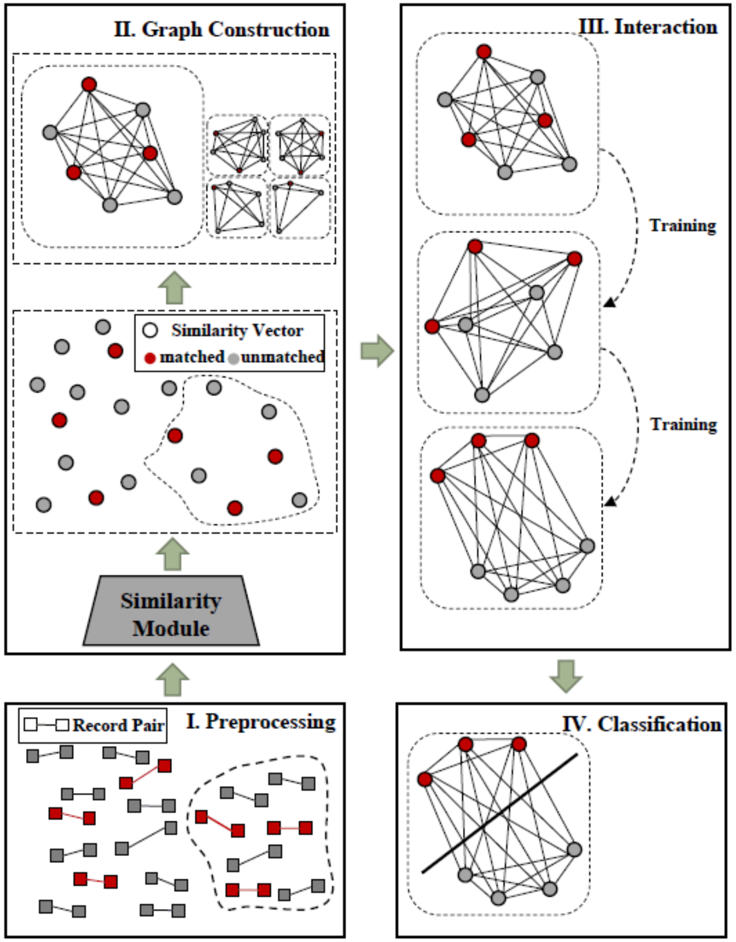


Figure 2: Framework of GNEM.

Graph Construction

Node: $(r, r' \in V_r)$
Transform the one-to-set matching problem into a set of node classify-cation problems

Edge: a complete graph

$$\mathcal{A}_r^{ij} = \mathcal{F}(Abs(\mathrm{f}_{r_i} - \mathrm{f}_{r_j}))$$

$\mathcal{F}$ is a stack of L fully-connected layers

$$\mathcal{A}_r^{ij} = \mathcal{F}(\mathrm{u}_{r,r_i} \oplus \mathrm{u}_{r,r_j})$$

Initializing node representations for $\mathcal{V}_r$

$$r.\mathrm{x}_i = \phi(\mathrm{w}_1, \cdots, \mathrm{w}_k)$$

$$\mathrm{u}_{r,r'} = \varphi(\mathrm{r}, \mathrm{r}')$$

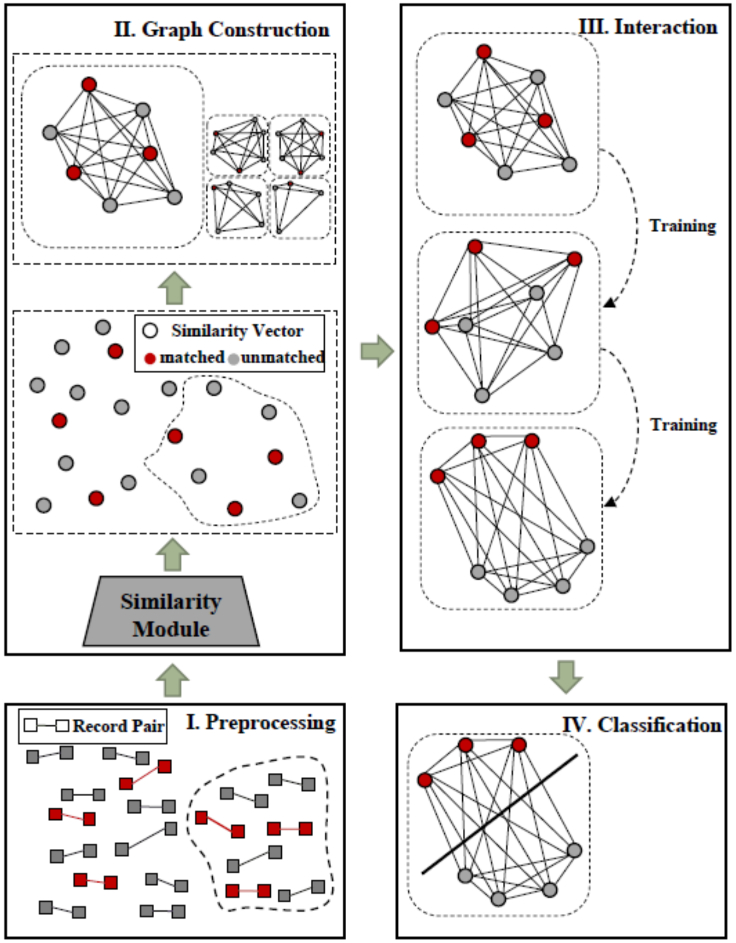


Figure 2: Framework of GNEM.

Interaction via Graph Neural Network

$$z^{(l)} = \sigma_1\left(W_{z,s}^{(l)} H^{(l-1)} + W_{z,n}^{(l)} \tilde{\mathcal{A}}_r H^{(l-1)}\right)$$

$$\tilde{H}^{(l)} = \sigma_2\left(W_{o,s}^{(l)} H^{(l-1)} + W_{o,n}^{(l)} \tilde{\mathcal{A}}_r H^{(l-1)}\right)$$

$$H^{(l)} = z^{(l)} \odot \tilde{H}^{(l)} + (1 - z^{(l)}) \odot H^{(l-1)}$$

Classification

$$\tilde{X}_r = H^{(L')}$$

$s_{r,r'}$: generated by the fully connected layer given $\tilde{X}_{r,r'}$

$$Pr(y_{r,r'} \mid \tilde{X}_{r,r'}) = \text{softmax}(W s_{r,r'} + b)$$

$$Pr(\hat{y} \mid r, r') = \text{Average}\left(Pr(y_{r,r'} \mid \tilde{X}_{r,r'}), Pr(y_{r',r} \mid \tilde{X}_{r',r})\right)$$

Table 3: Performance comparison results.

		SIF	RNN	Attention	Hybrid	BERT	XLNet	DistilBERT	RoBERTa
Abt-Buy	origin	35.1	39.4	56.8	62.8	85.9	86.8	83.3	90.9
	GNEM (w/o interaction)	29.7	41.3	55.5	65.8	85.4	87.8	81.2	91.3
	GNEM	**44.2**	**45.5**	**61.5**	**71.9**	**87.7**	**88.7**	**83.6**	**93.0**
Amazon-Google	origin	**60.6**	59.9	61.1	69.3	71.3	71.6	69.4	70.4
	GNEM (w/o interaction)	48.5	58.5	62.3	67.6	72.5	75.4	72.0	72.5
	GNEM	59.6	**65.3**	**64.1**	**69.4**	**74.7**	**77.6**	**73.0**	**76.0**
Walmart-Amazon	origin	65.1	67.6	50.0	66.9	83.9	78.2	82.3	84.9
	GNEM (w/o interaction)	65.9	**69.9**	54.2	63.7	82.6	**82.4**	79.8	85.9
	GNEM	**66.7**	69.3	**58.7**	**68.7**	**86.7**	81.6	**85.0**	**86.5**

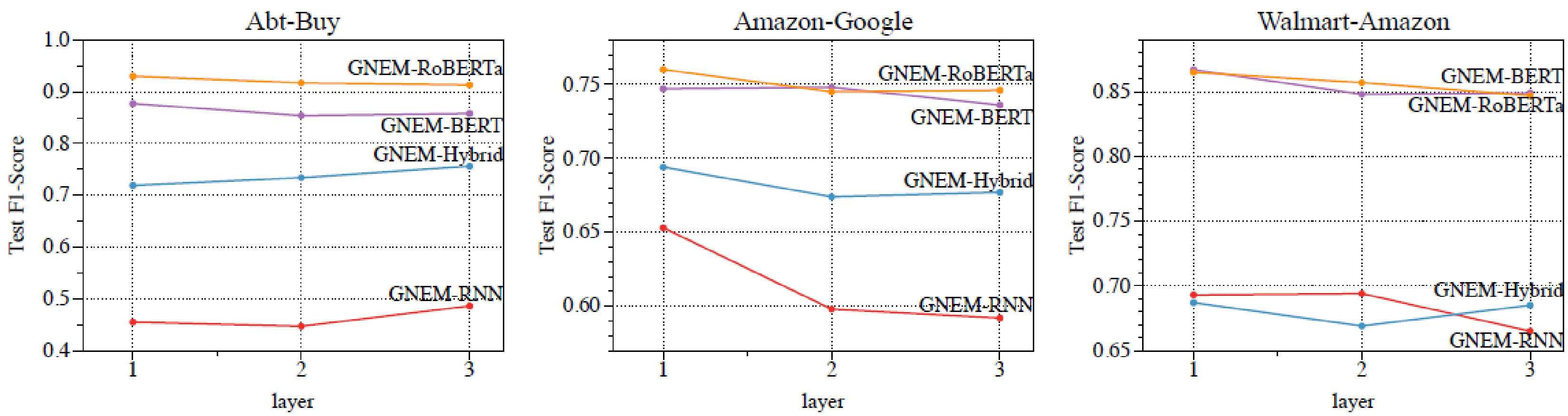


Figure 3: Effects of the number of gated graph convolution layers.

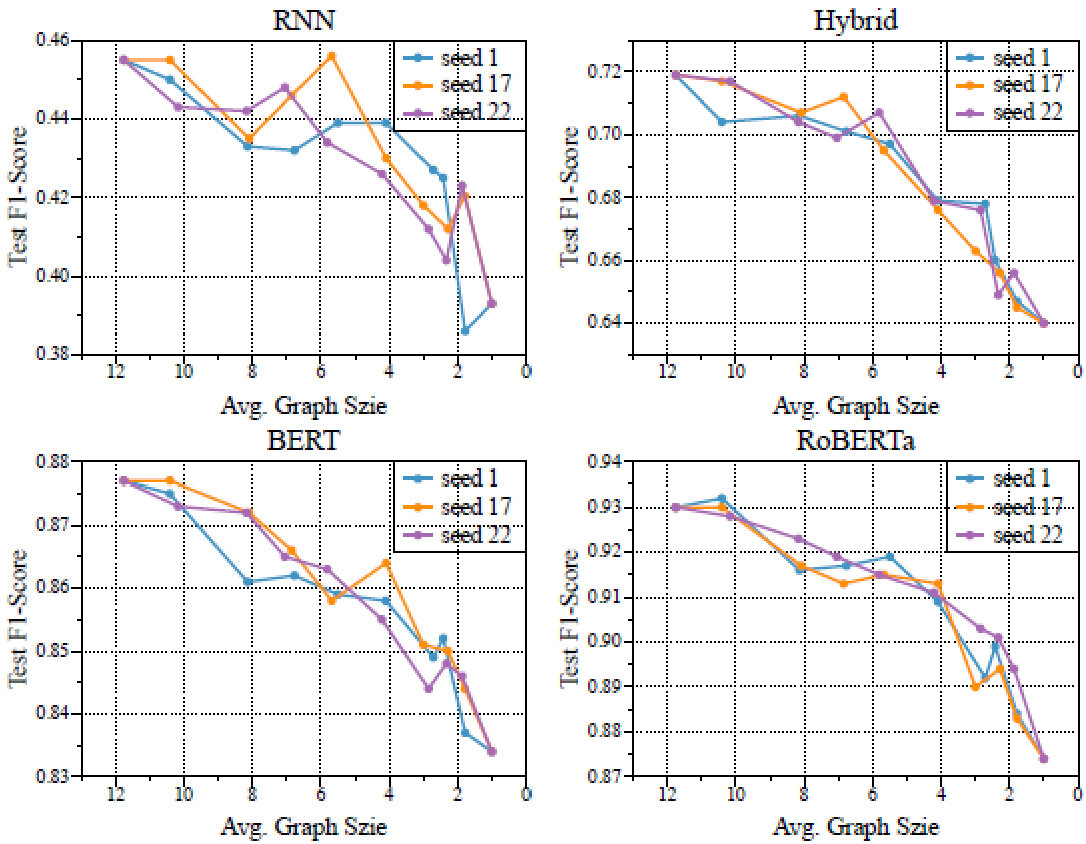


Figure 4: Effects of different graph sizes (Abt-Buy).

LATTE: Latent Type Modeling for Biomedical Entity Linking

Ming Zhu,[1*] Busra Celikkaya,[2] Parminder Bhatia,[2] Chandan K. Reddy[1]
[1]Department of Computer Science, Virginia Tech, Arlington, VA 22203
[2]AWS AI, Seattle, WA 98121
mingzhu@vt.edu, {busrac, parmib}@amazon.com, reddy@cs.vt.edu

AAAI2020

Two challenges

Similar surface level features

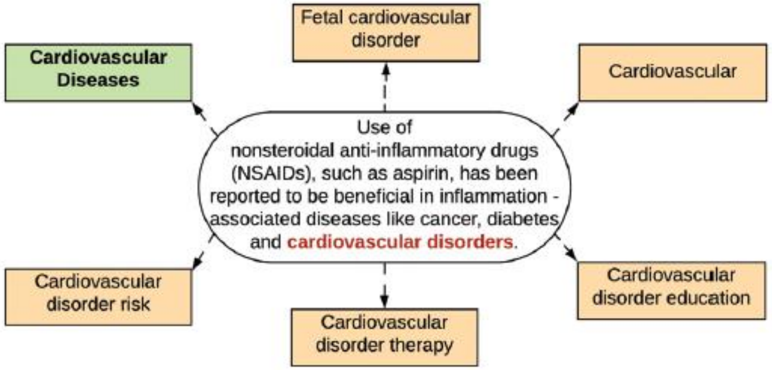


Figure 1: An example of biomedical entity linking. Phrase shown in red is the extracted mention, the orange boxes refer to the top candidate entities retrieved from the biomedical knowledge-base, and the green box is the ground truth entity for this mention.

Same type

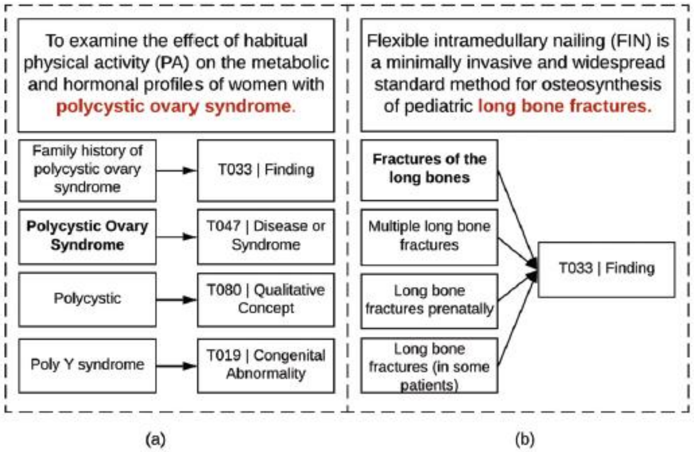
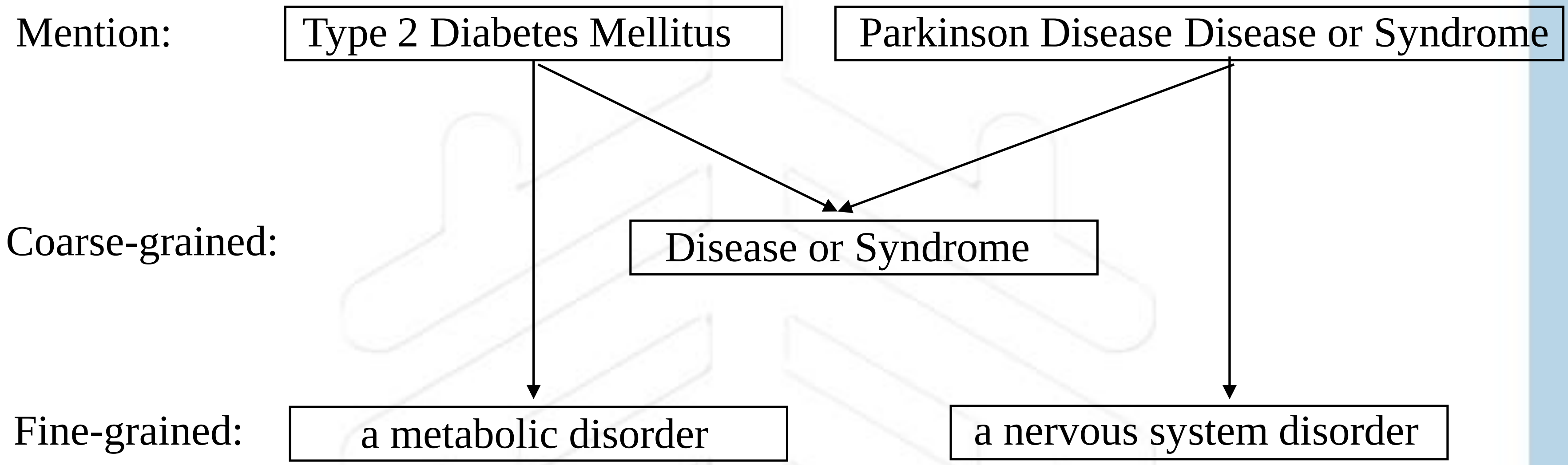
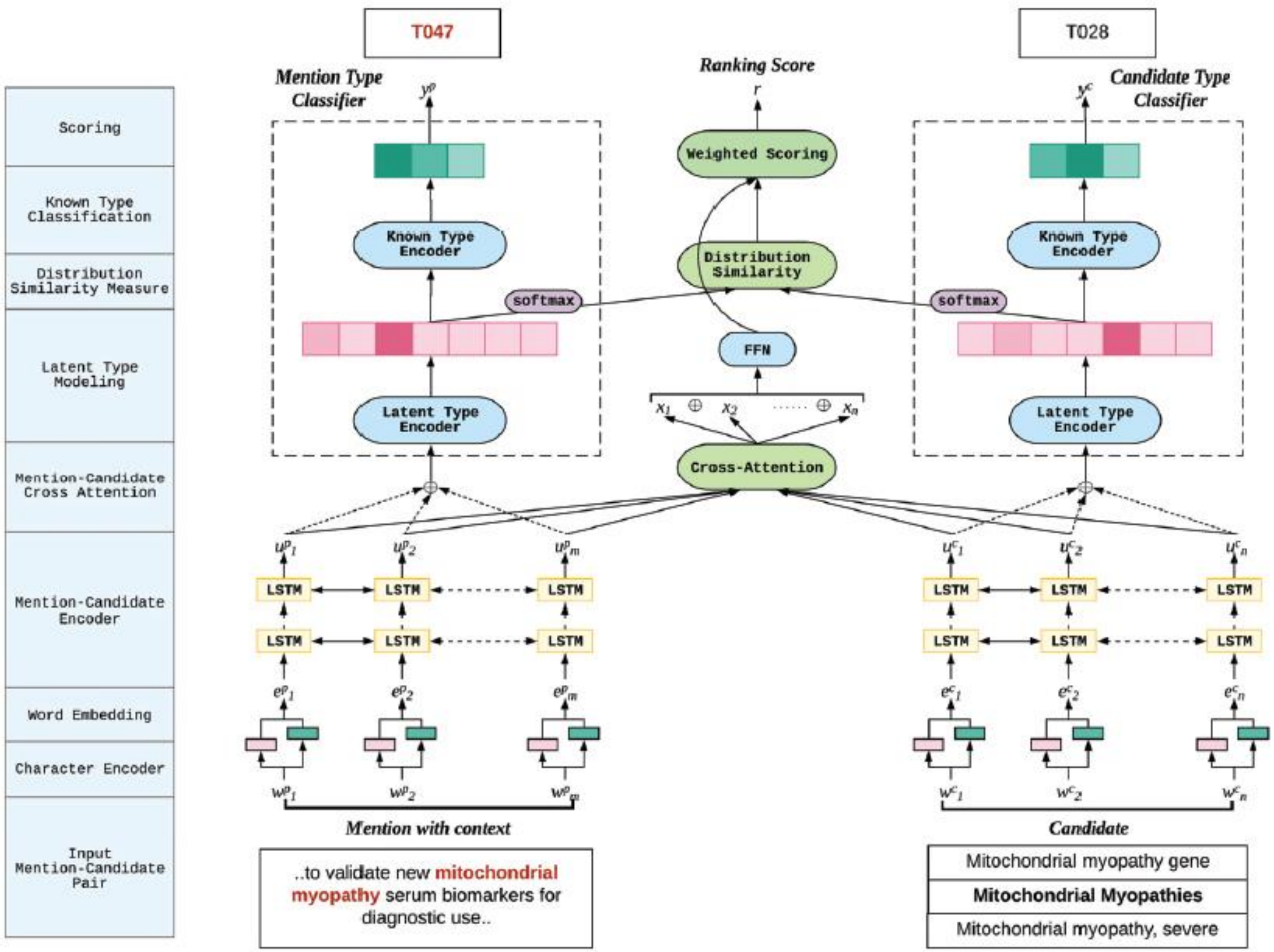


Figure 2: Examples of biomedical entity linking with type information.



Coarse-grained: UMLS(Unified Medical Language System) Sementic Types

Fine-grained: Latent Types



Scoring
Known Type Classification
Distribution Similarity Measure
Latent Type Modeling
Mention-Candidate Cross Attention
Mention-Candidate Encoder
Word Embedding
Character Encoder
Input Mention-Candidate Pair
T047
T028
Mention Type Classifier
Ranking Score
Candidate Type Classifier
Weighted Scoring
Known Type Encoder
Distribution Similarity
softmax
FFN
Latent Type Encoder
Cross-Attention
LSTM
Mention with context
Candidate
..to validate new mitochondrial myopathy serum biomarkers for diagnostic use..
Mitochondrial myopathy gene
Mitochondrial Myopathies
Mitochondrial myopathy, severe

$$u_i^p = [\overrightarrow{u_i^p}; \overleftarrow{u_i^p}] \quad u_i^c = [\overrightarrow{u_i^c}; \overleftarrow{u_i^c}]$$

Cross-Attention Layer:

$$S \in \mathbb{R}^{m \times n} \quad s_{ij} = w_a^T \cdot [u_i^c; u_j^p; u_i^c \odot u_j^p]$$

$$S^\alpha = \underset{row}{\mathrm{softmax}}(S),$$

$$\bar{S}^\beta = \underset{col}{\mathrm{softmax}}(S), \text{and} \quad S^\beta = S^\alpha \cdot \bar{S}^{\beta^T}.$$

$$a_j^\alpha = \sum_i s_{ij}^\alpha u_i^c, \qquad a_j^\beta = \sum_i s_{ij}^\beta u_i^p,$$

$$x_j = [u_j^p; a_j^\alpha; u_j^p \odot a_j^\alpha; u_j^c \odot a_j^\beta].$$

Sim. score: $f = ReLU(w_f \cdot X + b_f)$.

Latent Type Similarity

$$v^p = w_l \cdot u^p + b_l, \quad \hat{v^p} = \mathrm{softmax}(v^p),$$

$$v^c = w_l \cdot u^c + b_l, \quad \hat{v^c} = \mathrm{softmax}(v^c),$$

$$g = \frac{\hat{v^p} \cdot \hat{v^c}}{||\hat{v^p}|| \, ||\hat{v^c}||}.$$

Type distribution $\begin{cases} y^p = ReLU(w_k \cdot v^p + b_k) \\ y^c = ReLU(w_k \cdot v^c + b_k) \end{cases}$

Final score: $r = w_r^f \cdot f + w_r^g \cdot g$

Optimization

Type Classification loss: To incorporate the knowledge about the *known* categorical types into the semantic representation of mentions and the entities, we minimize the categorical cross-entropy loss. Given the known type $y \in \{y^p, y^c\}$ of a mention or a candidate, and its predicted type distribution $\hat{y}$, the loss is calculated as follows:

$$\mathcal{L}^{type} = -\sum_{j=1}^{K} y_j \log(\hat{y_j}) \qquad (9)$$

Mention-Candidate Ranking loss: For a given mention, we want to ensure that the correct candidate c_{pos} gets a higher score compared to the incorrect candidates c_{neg}. Hence, we use max-margin loss as the objective function for this task. Given the final scores $r_{p,c_{pos}}$ and $r_{p,c_{neg}}$ of p with respect to c_{pos} and c_{neg} respectively, the ranking loss is calculated as follows:

$$\mathcal{L}^{rank} = \max\{0, M - r_{p,c_{pos}} + r_{p,c_{neg}}\} \qquad (10)$$

Datasets

Dataset	Statistics	Train	Dev	Test
Med Mentions	#Documents	2,635	878	879
	#Mentions	210,891	71,013	70,364
	#Entities	25,640	12,586	12,402
3DNotes	#Documents	2,133	525	745
	#Mentions	22,266	5,373	8,065
	#Entities	2,026	1,030	1,209

Table 1: Statistics of the datasets used. Note that the "#Entities" refers to the number of unique entities.

	MedMentions		3DNotes	
Model name	**P@1**	**MAP**	**P@1**	**MAP**
TF-IDF	61.39	67.74	56.89	69.45
ARC-I	71.50	81.78	84.73	90.35
ARC-II	72.56	82.36	86.12	91.38
KNRM	74.92	83.47	84.32	90.04
Duet	76.19	84.92	86.11	91.19
MatchPyramid	78.15	86.31	85.97	91.32
MV-LSTM	80.26	87.58	87.90	92.44
Conv-KNRM	83.08	89.34	86.92	92.08
LATTE-NKT	86.09	91.27	86.40	91.09
LATTE	**88.46**	**92.81**	**87.98**	**92.49**

	MedMentions		3DNotes	
Model name	**P@1**	**MAP**	**P@1**	**MAP**
LATTE_base	80.02	86.94	84.08	90.15
LATTE_base+LT	86.09	91.27	86.40	91.09
LATTE_base+KT	87.73	92.33	87.80	**92.66**
LATTE	**88.46**	**92.81**	**87.98**	92.49