Knowledge and Information Systems (2023) 65:1221–1242
https://doi.org/10.1007/s10115-022-01793-3

REGULAR PAPER

Exploiting anonymous entity mentions for named entity linking

Feng Hou[1] · Ruili Wang[1] · See-Kiong Ng[2] · Michael Witbrock[3] · Fangyi Zhu[2] · Xiaoyun Jia[4]

Received: 30 January 2022 / Revised: 31 October 2022 / Accepted: 7 November 2022 /
Published online: 7 December 2022

Abstract
Named entity linking or named entity disambiguation is to link entity mentions to corresponding entities in a knowledge base for resolving the ambiguity of entity mentions. Recently, collective linking methods exploit document-level coherence of the referenced entities by computing a pairwise score between candidates of a pair of named entity mentions (e.g., *Raytheon* and *Boeing*) in a document. However, in a document, named entity mentions are significantly less frequent than anonymous entity mentions (e.g., *defense contractor* and *the company*). In this paper, we propose a method, DOCument-level Anonymous Entity Type words relatedness (DOC-AET), to exploit the document-level coherence between candidate entities and anonymous entity mentions. We use the anonymous entity type (AET) words to extract anonymous entity mentions. We learn embeddings of AET words from their inter-paragraph co-occurrence matrix; thus, the document-level entity-type relatedness is encoded in the AET word embeddings. Then, we compute the coherence scores between candidate

✉ Ruili Wang
ruili.wang@massey.ac.nz

Feng Hou
f.hou@massey.ac.nz

See-Kiong Ng
seekiong@nus.edu.sg

Michael Witbrock
m.witbrock@auckland.ac.nz

Fangyi Zhu
fyzhu@nus.edu.sg

Xiaoyun Jia
Dr.SophiaJia@outlook.com

1 School of Mathematical and Computational Sciences, Massey University, Palmerston North, New Zealand

2 Institute of Data Science, National University of Singapore, Queenstown, Singapore

3 School of Computer Science, University of Auckland, Auckland, New Zealand

4 Institute of Governance and School of Politics and Public Administration, Shandong University, Jinan, China

entities and anonymous entity mentions using the AET entity embeddings and document context embeddings. By incorporating such coherence scores for candidates ranking, DOC-AET has achieved new state-of-the-art results on two of the five out-domain test sets for named entity linking.

Keywords Entity linking · Fine-grained entity types · Anonymous entity type words · Entity embeddings

1 Introduction

Named entity linking (NEL) or named entity disambiguation (NED) is the task of automatically resolving the ambiguity of entity mentions in textual documents by linking them to the corresponding concrete entities in a knowledge base (KB). For example, in Fig. 1, the referenced entity of the mention "Nardelli" should be the American *businessman* "Robert Nardelli" in Wikipedia. NEL has been used in pre-processing for tasks such as information extraction [19], information retrieval [9] and question-answering [43], [7].

Entity linking systems typically consist of two sequential modules: candidate entities generation and candidate entities ranking [36]. Candidate entities are selected by the mention-entity prior and the local context-entity similarity score [13]. Coarse-grained entity type information (e.g., *person, organization, location*) has been used for candidate entities selection [13], [8]. Research in NEL has largely focused on two types of contextual information for candidate entities ranking: local information and global information. Local information is based on words that appear in the context window around an entity mention. For global information, the document-level coherence of the referenced entities is exploited to make compatible linking decisions collectively [15], [39], [34]. Recently, deep learning-based entity linking methods use pre-trained entity embeddings [39], [40], [13] and on-site local and global score functions to rank the candidate entities [13], [25].

Currently, there are mainly two approaches for improving entity linking:

- Exploiting fine-grained type information of candidate entities. This approach implicitly embeds type information in entity embeddings using human curated type labels [17], [3], such as the FIGER type taxonomy [27].

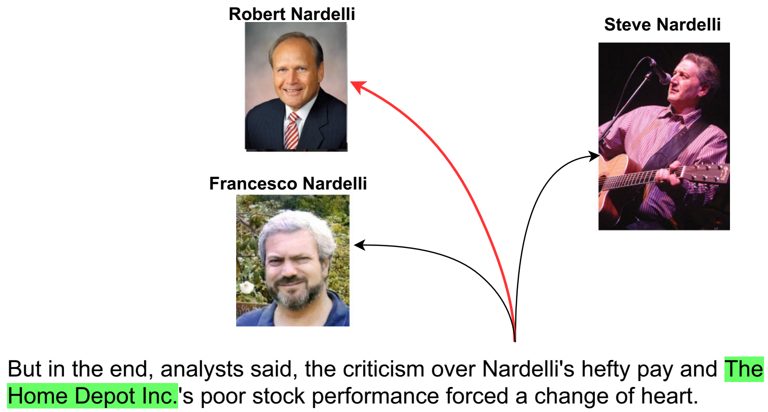


Fig. 1 Local model for candidate ranking for entity linking

- Collective linking by using global information. For example, the end-to-end deep collective linking model [13] achieves differentiable message passing by casting loopy belief propagation [32] as a rolled-out deep network. Multiple latent relations between mentions in a document [25] are also exploited to capture document-level coherence . Another way of using global information is to sequentially link and accumulate dynamic context information from linked entities [42].

However, the issues with the above two approaches include:

- The type information by human curated type labels is unable to efficiently mesh with the local context, which is plain vocabulary words. The human curated type labels are still too coarse-grained. Embedding the human curated type information cannot learn the contextual commonality of entities that are of the same fine-grained types. For example, all the entities of *footballer* and *golfer* are typed as *person/athlete* using FIGER [27] type taxonomy, but the contexts of *footballer* and *golfer* are different.
- All the aforementioned collective linking methods exploit the coherence of candidate entities of named entity mentions (e.g., "Nardelli" and "Home Depot Inc"). However, such named entity mentions appear less frequently than anonymous entity mentions (e.g., *the company*[1] in Fig. 2). Thus, such methods can only use limited global information, but the more frequently occurring anonymous entity mentions are ignored. The anonymous entity mentions are more likely to be plain vocabulary words (e.g., the *company, Canadian singer, service provider, news agency*, etc.) than human curated type labels.

We observe that many plain vocabulary words appear frequently as appositions of entities (e.g., *Defense contractor* Raytheon), coreferences of entities (e.g., the *company*) or anonymous entity mentions (e.g., *American defense firms*). These plain vocabulary words provide more fine-grained semantic types of entities and can help (i) anchor diversified contexts of entities of the same type, e.g., the contexts of vocabulary word *company* are similar to the contexts of entities of *company*; (ii) capture the document-level relatedness of entities of different types, e.g., the *company* and *executive officer* are more likely to be in the same document. These words are parts of anonymous entity mentions, and we call such words **Anonymous Entity Type** (**AET**) words.

Our hypothesis is that these AET words provide more effective contextual information for entity linking. We can use the anonymous entity mentions in a document to infer the types of the named entity mentions. For example, in Fig. 2, *company* and *chief executive* are highly related to each other in documents; when ranking the candidate entities of "Nardelli," the entity "Robert Nardelli" with type *chief executive* is more coherent with the document that has many anonymous *company* mentions. If a linking model learns the contextual commonality of *chief executive* entities, it can correctly select entities of similar types using similar contextual information.

In this paper, we propose a method, **DOC**ument-level **A**nonymous **E**ntity **T**ype words relatedness (DOC-AET), to exploit the affluent anonymous entity mentions. DOC-AET incorporates a new candidate ranking score by computing the coherence score between candidate entities and anonymous entity mentions. We use AET words to extract anonymous entity mentions. DOC-AET learns embeddings of AET words from the document-level AET words inter-paragraph co-occurrence matrix, and the document-level relatedness of AET words is encoded in the AET word embeddings. The coherence scores between candidate entities and anonymous entities are computed based on the AET word embeddings.

[1] We use italic font to represent anonymous entity mentions.

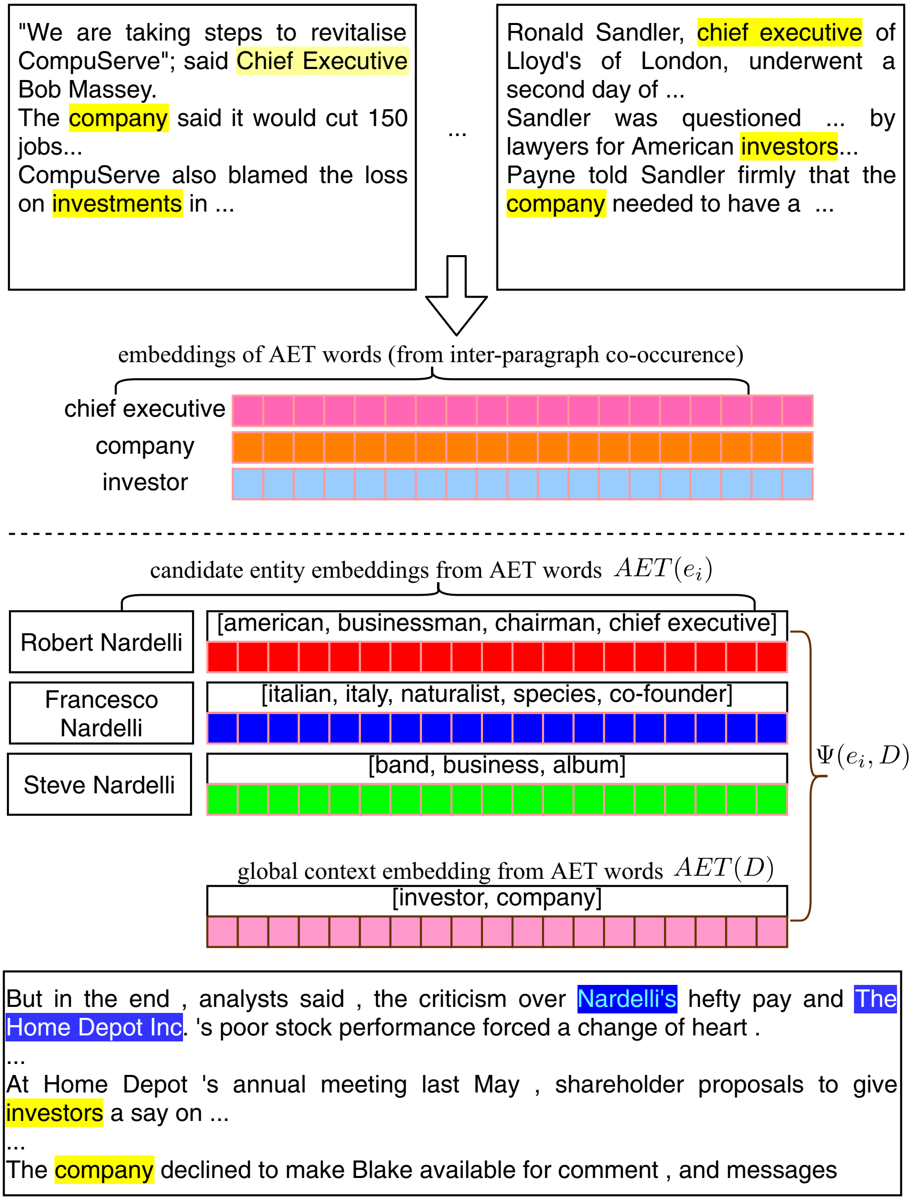


Fig. 2 The process of incorporating the coherence score between entity candidates and Anonymous Entity Type (AET) words (anonymous entity mentions). The AET words are highlighted

We evaluate our method on standard benchmark datasets and achieve new state-of-the-art performance on three of the five out-domain test sets for entity linking. Our contributions can be summarized as follows:

- For the first time, the document-level relatedness of fine-grained entity types is explored. We propose a novel method to capture the relatedness of AET words from document-

level context, i.e., extracting AET words' inter-paragraph co-occurrence and learning AET word embeddings. The document-level relatedness of AET words is encoded in the AET word embeddings.
- We incorporate a new coherence score based on AET entity embeddings and document's AET context embeddings. This coherence score can be combined with any candidate entities ranking methods.
- We verify the effectiveness of our method on standard benchmark datasets.

2 Background

2.1 Named entity linking

Formally, given a knowledge base (KB) that contains a set of entities E and a document D in which a set of named entity mentions M are identified in advance, the goal of entity linking is to link each entity mention $m_i \in M$ to its corresponding entity $e_i \in E$. It is possible that an entity mention does not have its corresponding entity in the given KB (i.e., e_i =NIL).

Because $|E|$ can be very large, entity linking systems typically consist of two modules: candidate entity generation and candidate entity ranking. Candidate entities generation is to select possibly referenced entities E_m in the KB for mention m. Candidate entities ranking is to rank the candidate entities in E_m to find out which entity $e \in E_m$ is the most likely referenced entity. Research on NEL has largely focused on the following two types of candidate ranking scores.

2.2 Local models for candidate ranking

Local models rely only on local contexts of mentions and completely ignore interdependencies between the linking decisions in the document (these interdependencies are usually referred to as coherence). Suppose a document D contains a list of mentions $m_1, \ldots, m_n$. Let c_i be a local context of mention m_i and $\Psi(e_i, c_i)$ be a local score function. A local model [24] [40], [13], [25] then tackles the problem by searching the highest scored candidate

$$e_i^* = \underset{e_i \in E_{m_i}}{arg\ max}\ \Psi(e_i, c_i) \tag{1}$$

for each mention $m_i, i \in \{1, \ldots, n\}$.

The local score $\Psi(e_i, c_i)$ measures the relevance of entity candidates of each mention independently. Neural network-based NEL models usually compute Ψ as follows:

$$\Psi(e_i, c_i) = \mathbf{e}_i^\top \mathbf{B}\ f(c_i) \tag{2}$$

where $\mathbf{e}_i \in \mathbb{R}^d$ is the embedding of candidate entity e_i; $\mathbf{B} \in \mathbb{R}^{d\times d}$ is a diagonal matrix; $f(c_i) \in \mathbb{R}^d$ is a feature representation of local context c_i surrounding mention m_i.

The local context score is combined with the context-independent mention-entity prior $\hat{p}(e|m)$ [13] as follows:

$$\Psi(e_i, c_i, m_i) = g(\Psi(e_i, c_i), \hat{p}(e_i|m_i)) \tag{3}$$

where g is a neural network with two fully connected layers and ReLU activation function.

Table 1 Definitions of used notations

Notation	Definition and description
E	The set of entities of a knowledge base
D	A document contains a set of entity mentions
E_m	Candidate entities for mention m
m_i	The ith entity mention in a document needs to be resolved
c_i	The local context of entity mention m_i
$\mathbf{e}_i$	The entity embedding of entity e_i
$\mathbf{a}_e$	The AET embedding of entity e
$\mathbf{a}_D$	The AET embedding of document D
X_{ij}	The co-occurrence count of AET word i and j
f	A function for context representations
g	A neural network for computing the local coherence score
$\hat{p}(e_i\|m_i)$	Mention-entity prior, i.e., the prior probability of e_i being the referent of mention m_i
$\Psi(e_i, c_i)$	The coherence score between the candidate entity e_i and the context c_i of mention m_i
$\Psi(e_i, D)$	The coherence score between candidate e_i and the AET words in document D
$\Phi(e_i, e_j, D)$	The coherence score between entity e_i and e_j, which are candidate entities of mention m_i and m_j in document D, respectively

2.3 Global models for candidate ranking

Global models make collective linking decisions by taking into account the coherence among the referent entities in a document [28], [23]. Besides using local score $\Psi(e_i, c_i)$, a global model incorporates a global coherence score function $\Phi(E, D)$:

$$E^* = \underset{E \in E_{m_1} \times \ldots \times E_{m_n}}{arg\ max} \left(\sum_{i=1}^{n} \Psi(e_i, c_i) + \Phi(E, D) \right) \tag{4}$$

where $E = (e_1, \ldots, e_n)$. The global coherence score, in the simplest form, is a sum over all pairwise scores $\Phi(e_i, e_j, D)$ ([12], [16], [15], [39]) as follows:

$$E^* = \underset{E \in E_{m_1} \times \ldots \times E_{m_n}}{arg\ max} \left(\sum_{i=1}^{n} \Psi(e_i, c_i) + \sum_{i \neq j} \Phi(e_i, e_j, D) \right) \tag{5}$$

Deep learning-based entity linking models compute the pairwise score $\Phi(e_i, e_j, D)$ as follows:

$$\Phi(e_i, e_j, D) = \frac{1}{n-1} \mathbf{e}_i^{\top} \mathbf{C}\, \mathbf{e}_j \tag{6}$$

where $\mathbf{e}_i$ and $\mathbf{e}_j \in \mathbb{R}^d$ are the embeddings of entity e_i, e_j, which are candidates for mention m_i and m_j, respectively; $\mathbf{C} \in \mathbb{R}^{d \times d}$ is a diagonal matrix. The pairwise score of [25] considers K latent relations between entities.

$$\Phi(e_i, e_j, D) = \sum_{k=1}^{K} \alpha_{ijk}\, \mathbf{e}_i^{\top} \mathbf{R}_k\, \mathbf{e}_j \tag{7}$$

where α_{ijk} is the weight for relation k and $\mathbf{R}_k$ is a diagonal matrix for measuring relations k between two entities.

However, finding the exact solution of Eq. (5) is NP-hard [25]. Previous work has investigated different approximation techniques. For example, Ganea and Hofmann [13] use loopy belief propagation (LBP), an approximate inference method based on message passing [32]. Differentiable message passing is performed by casting loopy belief propagation (LBP) as a rolled-out deep network. The linking model directly optimizes the marginal likelihoods, using the same networks for learning and prediction. They use truncated fitting of LBP to a fixed number of message passing iterations.

3 Related work

Our research focuses on improving NEL by exploiting the plain vocabulary words that can be used as anonymous entity mentions and fine-grained entity types. DOC-AET exploits the (AET) coherence between candidate entities' type and anonymous entities' type to rank candidate entities.

3.1 NEL using entity type information

Coarse-grained entity type information (e.g., *person, organization, location*) has been used for candidate entities selection [13], [8]. Fine-grained entity type information is usually encoded into entity embeddings. Entity embeddings are the vector representations of entities built from entity–entity co-occurrences [39], [11], [45], or canonical Wikipedia articles and local context surrounding anchor links [13]. [17] map entities' Freebase types to the FIGER [27] types, and learn entity embeddings and type embeddings jointly on the training data. [3] extract latent entity type information from the embeddings generated by applying the pre-trained BERT encoder to the Wikipedia context of entities.

These efforts focus on the type information of named entity mentions. As such, we aim to exploit the coherence between candidate entities and anonymous entity types (mentions) in the documents.

3.2 Word embeddings and entity embeddings

3.2.1 Word embeddings

Word embeddings, such as Word2Vec [29] and GloVe [33], exclusively exploit the intra-sentence context of words to capture the semantic and syntactic similarities. In this paper, we use the inter-paragraph co-occurrence of AET words to capture the document-level relatedness of anonymous entity types.

3.2.2 Entity embeddings

Similar to word embeddings, entity embeddings are the vector representations of entities. The methods of [11], [39], and [45] use data about entity–entity co-occurrences to learn entity embeddings and often suffer from sparsity of co-occurrence statistics. [13] learned entity embeddings using words from canonical Wikipedia articles and local context surrounding anchor links. They used Word2Vec vectors [29] of positive words and random negative words as input to the learning objective. Thus, their entity embeddings are aligned with the Word2Vec word embeddings.

3.3 Embeddings aggregation

Linear aggregations of vector representations have been used to fuse information from different sources (e.g., polysemous words have multiple sense embeddings). [1] hypothesizes that the global word embedding is a linear combination of its sense embeddings. They show that senses can be recovered through sparse coding. [31] show that senses and word embeddings are linearly related and sense sub-spaces tend to intersect over a line. [38] probe the aggregated word embeddings of polysemous words for semantic classes. They create a WIKI-PSE corpus, where word and semantic class pairs are annotated using Wikipedia anchor links, e.g., “apple” has two semantic classes: *food* and *organization*. A separate embedding for each semantic class was learned based on the WIKI-PSE corpus. They find that the linearly aggregated embeddings of polysemous words represent well their semantic classes. In this paper, we use linear aggregations of AET word embeddings to generate AET-based representations of entities and a document.

3.4 Fine-grained entity typing

To use entity type information, the entities must firstly be typed. Fine-grained entity typing (FGET) is a task of classifying entities into fine-grained types [27] or ultra-fine-grained semantic labels [6]. Mention-level FGET only infers the entity types that are coherent with a specific context [14], [21], while entity-level FGET considers all possible types [37]. Gupta *et al.* [17] mapped the Freebase types of entities to FIGER [27] types, but this method is less credible, as noted in [14]. [2] used a memory-based network to generate a short description of an entity, e.g., “Roger Federer” is described as “Swiss tennis player.” In this paper, we extract a flat list of AET words from the first paragraph of Wikipedia articles and use these AET words to generate AET-based entity embeddings.

4 DOC-AET: method overview

4.1 Motivation

Our DOC-AET method aims at exploiting the coherence between anonymous entity mentions and candidate entities’ types to improve NEL. We observe that fine-grained AET words appear frequently as apposition (e.g., *Defense contractor* Raytheon), coreference (e.g., the *company*) or anonymous entities (e.g., *American defense firms*) in documents. All these can be viewed as anonymous entity mentions. We can use the AET words from unlabeled documents to capture the document-level relatedness of anonymous entity types. But to capture the longer contexts or document-level relatedness, we only consider the **inter-paragraph co-occurrence**, i.e., we only count the co-occurrence of AET words in neighboring paragraphs instead of in the same paragraph.

4.2 AET words dictionary

We use the 3,227 fine-grained type words of our previous work [20] as AET words. These words are of the following categories:

- Profession/subject, e.g., *footballer*, *soprano, biology, rugby*.

- Title, e.g., *president, CEO, head, director*.
- Industry/genre, e.g., *carmaker*, *manufacturer, defense contractor*, *hip hop*.
- Geospatial, e.g., *Canada, Asian*, *Australian*.
- Ideology/religion, e.g., *communism, Buddhism*.
- Miscellaneous, e.g., *book, film, tv, ship, language*.

Each candidate entity will be typed by extracting AET words from the first sentence of the Wikipedia article. Each document will be represented by the AET words extracted from the document.

4.3 Process of DOC-AET method

Exploiting the coherence between anonymous entity mentions and candidate entities' types is not trivial. As shown in Fig. 1, the general process can be summarized as follows:

Step 1: Extract anonymous entity mentions (highlighted AET words) from unlabeled documents and build document-level inter-paragraph co-occurrence matrix of AET words; then learn inter-paragraph AET words embeddings from the co-occurrence matrix. This step is shown in the upper part above the dashed line. This step is to **learn and encode the document-level relatedness of AET words in their embeddings**. We describe this step in Sect. 5.

Step 2: Generate AET entity embeddings using the AET words extracted from the first paragraph of Wikipedia articles. For example, three AET types are extracted from the Wikipedia article of "Steve Nardelli": *band*, *business* and *album*, then the entity embedding of "Steve Nardelli" is generated by averaging the embeddings of these three AET words.

Step 3: Incorporate a coherence score $\Psi(e_i, D)$ between candidate entities' embeddings and document AET context embeddings. For example, the document has two AET words: *investor* and *company*, the AET context embedding is generated by averaging the embeddings of these two AET words. Details of this step is given in Sect. 6.

5 Generate AET word embeddings

We build AET words' inter-paragraph co-occurrence matrix from unlabeled documents and then learn the word embeddings from the inter-paragraph co-occurrence matrix. This process is similar to the method of GloVe [33]. The document-level relatedness of AET words will be encoded in the embeddings.

5.1 Document-level inter-paragraph co-occurrence of AET words

The local context score in Equation (2) captures the local context within a sentence, while our DOC-AET score aims at capturing the entity type relatedness across paragraphs (i.e. longer context). We only extract inter-paragraph co-occurrence of AET words, instead of the immediate neighboring context words.

For each document, we extract a list of AET words from each paragraph. Each document is transformed into a structure of a two-dimensional list of AET words. For example, the document in Fig. 3 can be represented as: [[*'online'*, *'service'*], [], [*'chief'*, *'executive'*], [*'company'*, *'programme'*]].

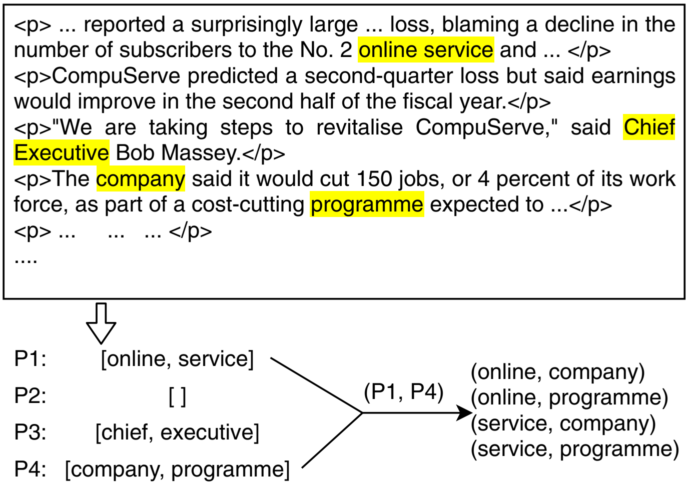


Fig. 3 Building AET words inter-paragraph co-occurrence matrix. AET words are highlighted

As shown in Fig. 3, we build AET words' inter-paragraph co-occurrence matrix from the structured representations of documents. Each paragraph is treated as a word. For a pair of paragraphs within context window, we pick one AET word from the left paragraph and right paragraph, respectively, to count co-occurrence. For example, if paragraph [*'online'*, *'service'*] and paragraph [*'company'*, *'programme'*] are picked, the co-occurrences of (*online*, *company*), (*online*, *programme*), (*service*, *company*) and (*service*, *programme*) are updated. Paragraph pairs that are p paragraphs apart contribute $1/p$ to the total count. We build a symmetric co-occurrence matrix.

5.2 Learn AET word embeddings

We use the weighted least squares regression model of GloVe [33] to learn the AET word embeddings. The cost function is as follows:

$$J = \sum_{i,j=1}^{V} h(X_{ij})(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij})$$

where X_{ij} is the co-occurrence count of word i and j, $w \in \mathbb{R}^a$ are the AET word embeddings. The weighting function $h(x)$ is defined as follows:

$$h(x) = \begin{cases} (x/x_{max})^{\alpha}, & \text{if } x < x_{max} \\ 1, & \text{otherwise} \end{cases} \tag{8}$$

The model generates two sets of word vectors w and $\tilde{w}$, $\tilde{w} \in \mathbb{R}^a$ are separate word embeddings (w and $\tilde{w}$ are equivalent as our X is symmetric) .

5.3 Differences to GloVe and Word2Vec

Both GloVe [33] and Word2Vec [29] use the neighboring words within a fixed window as context for learning word embeddings. Thus, the word embeddings of GloVe and Word2Vec are good at encoding the local contexts of words. In contrast, our AET word embeddings are learned using the AET words in neighboring paragraphs. Thus, our AET word embeddings are good at encoding the document-level relatedness of AET words.

6 Incorporating AET scores

6.1 Entity embeddings from AET words

To represent entities using AET words, we extract AET words from the first paragraph of Wikipedia articles. Suppose entity e has T AET words, the AET entity embedding $\mathbf{a}_e$ of e is generated by averaging the AET word embeddings of these T words.

$$\mathbf{a}_e = \frac{1}{T}\sum_{i=1}^{T} w_i \tag{9}$$

where $w \in \mathbb{R}^a$ are the AET word embeddings.

6.2 Document context embeddings from AET words

The document AET context embedding $\mathbf{a}_D$ is generated similarly by averaging the embeddings of AET words extracted from the document. Suppose L AET words are extracted from document D, the AET context embedding of D is

$$\mathbf{a}_D = \frac{1}{L}\sum_{i=1}^{L} w_i$$

where $w \in \mathbb{R}^a$ are the AET word embeddings.

6.3 Local AET scores using document context

The AET coherence score of entity e_i is computed as follows:

$$\Psi(e_i, D) = \mathbf{a}_i^\top \mathbf{A}\, \mathbf{a}_D \tag{10}$$

where $\mathbf{a}_i \in \mathbb{R}^a$ is the AET embedding of candidate entity e_i; $\mathbf{A} \in \mathbb{R}^{a\times a}$ is a diagonal matrix; $\mathbf{a}_D \in \mathbb{R}^a$ is the AET context embedding of document D. After incorporating this score, Equation (3) becomes:

$$\Psi(e_i, c_i, m_i, D) = f(\Psi(e_i, c_i), \Psi(e_i, D), \hat{p}(e_i|m_i)) \tag{11}$$

6.4 Model training

Following [25], we use Equation (11) and Equation (7) to define a conditional random field (CRF) as follows:

$$q(E_D|D) \propto \left\{ \sum_{i=1}^{n} \Psi(e_i, c_i, m_i, D) + \sum_{i\neq j} \Phi(e_i, e_j, D) \right\} \tag{12}$$

The max-marginal probability for each mention-candidate is estimated using max-product loopy belief propagation (LBP):

$$\hat{q}_i(e_i|D) \approx \max_{\substack{e_1,\ldots,e_{i-1},\\ e_{i+1},\ldots,e_n}} q(E_D|D) \tag{13}$$

The final score for ranking entity candidates is defined as follows:

$$\rho_i(e) = g(\hat{q}_i(e|D), \hat{p}(e, m_i))$$

where g is a two-layer neural network and $\hat{p}(e|m)$ is the context-independent mention-entity prior.

The other parts of training the model are the same as [25]. The key aspects are as follows:

- The model is trained by minimizing the marginal ranking loss as follows:

$$L(\theta) = \sum_{D\in\mathcal{D}} \sum_{m_i\in D} \sum_{e\in E_{m_i}} max(0, \gamma - \rho_i(e_i^*) + \rho(e)) \tag{14}$$

  where θ are the model parameters and $\mathcal{D}$ is the collection of training documents.
- To encourage diversity, a regularization term is added to the loss function in Equation 14.
- Adam [22] is used as an optimizer.

7 Experimental evaluation

7.1 Datasets for AET word embeddings

We use the RCV1, TREC-Disk5 (LA TIMES), and TREC-Disk4 (FINANTIAL TIMES) as training corpus for learning AET word embeddings. These datasets have paragraph segments and suit our method. We obtain 1,072,120 documents, and 3,140 AET words appear in these documents.

7.2 Datasets for NEL

We validate the effectiveness of our method on the benchmark datasets shared by [13], [25] and [26]. The details about the datasets are as follows:

AIDA-CoNLL [18] is one of the biggest manually annotated entity linking datasets. It contains AIDA train for training, AIDA-A for dev, and AIDA-B for testing, having, respectively, 946, 216, and 231 documents.

MSNBC, AQUAINT, ACE2004 were cleaned and updated by Guo and Barbosa [16] and have, respectively, 20, 50, and 36 documents for test only. **MSNBC** is created from MSNBC news articles. **AQUAINT** is created from Xinhua News Service, the New York Times, and Associated Press news corpus. **ACE2004** is a subset of ACE2004 co-reference documents annotated by Amazon Mechanical Turk.

WNEDCWEB (CWEB), WNED-WIKI (WIKI) were automatically extracted from ClueWeb and Wikipedia [16], [10] and have 320 documents each for test only.

Following previous works, we use the preprocessed data shared by [13], [25] and [26]. We use the **AIDA-CoNLL** [18] dataset for in-domain training and validation. We use the other datasets as out-domain datasets for evaluating our linking model based on DOC-AET score.

7.3 Evaluation metrics and baselines

We use the standard Micro-F1 score as an evaluation metric. The method of computing **Recall**, **Precision** and **Micro-F1** can be found in the survey of [36].

Our research is following the studies of [13], [25] and [20]. Thus, we use their linking methods (named **DeepEd**, **DeepEd+MulRel**, **DeepEd+MulRel+FGS2EE**, respectively) as baselines. We also compare our method with other state-of-the-art entity linking models.

7.4 Experimental settings

For AET word embeddings, we set the dimension a to 100, and the x_{max} and α in Equation (8) are set to 100 and 0.75, respectively. The context window for building inter-paragraph co-occurrence is set to 10. We use the Wikipedia entity-AET list [2] shared by our previous work to generate the AET entity embeddings in Equation (9).

In Equation (2) and Equation (7), we use the semantic reinforced entity embeddings of our previous work [20] to capture local contexts.

We modify the PyTorch code of **MulRel** [25] [3] to incorporate the AET coherence score. Following [25], we use the following parameter values: $\gamma = 0.01$ (in Equation 14), the number of LBP loops is 10; the f in Equation 10 is a neural network with two fully connected layers of 100 hidden units and ReLU nonlinearities. We select **ment-norm**, $K = 3$ (in Equation 7). The learning rate starts with 10^{-4} and changes to 10^{-5} when the F1 score on the dev set reaches 91.5%. The model is trained and evaluated on a single GTX 1080 Ti GPU.

Similar to [13] and [25, 26], we run our NEL system 5 times on the same datasets and report the mean and 95% confidence interval of the Micro-F1 score.

Our data, source code, and trained models are publicly available at https://github.com/fhou80/DOC-AET.

7.5 Results and discussion

The results on six test sets are shown in Table 2. The linking methods are categorized into four types.

Firstly, we compare our system to fully supervised systems, which were trained on AIDA-CoNLL documents. Recall that every mention in these documents has been manually annotated or validated by a human expert. Compared with all the fully supervised systems, including our direct baselines **DeepEd** [13], **DeepEd+MulRel** [25], and **DeepEd+MulRel+FGS2EE** [20], our approach is very effective and achieved significant improvement on three of the five out-domain test sets. The three out-domain test sets, **MSNBC**, **AQUAINT**, and **ACE2004**, are small data sets manually cleaned and labeled from news articles. The writing styles of these news articles are similar to our datasets for learning AET word embeddings. Compared with **DeepEd+MulRel+FGS2EE** [20], it is fair to say that incorporating the AET coherence scores can improve performance on all out-domain test sets with a slight drop on the in-domain test set.

We then compare our system to the systems that relied on Wikipedia and those which used Wikipedia along with unlabeled data ('Wikipedia + unlabeled data'). Our model outperformed all of them on the in-domain test set and two of the five out-domain test sets.

[2] download from https://drive.google.com/open?id=1OtLnrH4SpDzdNNcuca-DdXCMwsDPsG3B.

[3] https://github.com/lephong/mulrel-nel.

Table 2 F1 scores on six test sets The AIDA-B dataset is the in-domain test set, while the other five datasets are out-domain test sets. Best results are in boldface and second best results are underlined. The methods in bold are our direct baselines

Linking methods	AIDA-B	MSNBC	AQUAINT	ACE2004	CWEB	WIKI
Wikipedia						
Milne and Witten (2008) [30]	-	78	85	81	64.1	81.7
Ratinov et al. (2011) [35]	-	75	83	82	56.2	67.2
Hoffart et al. (2011) [18]	-	79	56	80	58.6	63
Cheng and Roth (2013) [4]	-	90	90	86	67.5	73.4
Chisholm and Hachey (2015) [5]	84.9	-	-	-	-	-
Wiki + Unlabeled data						
Lazic et al. (2015) [24]	86.4	-	-	-	-	-
wnel [26]	89.66±0.16	92.2±0.2	**90.7±0.2**	88.1±0.0	<u>78.2±0.2</u>	81.7±0.1
Wiki + Extra supervision						
Chisholm and Hachey (2015) [5]	88.7	-	-	-	-	-
Fully supervised(Wiki+ AIDA train)						
Guo and Barbosa (2016) [16]	89.0	92	87	88	77	**84.5**
Globerson et al. (2016) [15]	91.0	-	-	-	-	-
Yamada et al. (2016) [39]	91.5	-	-	-	-	-
DCA-RL [42]	**93.73±0.2**	93.80±0.0	88.25±0.4	90.14±0.0	75.59±0.3	78.84±0.2
RMA [44]	91.5	93.2	88.3	89.3	**79.3**	<u>82.2</u>
DeepEd [13]	92.22±0.14	93.7±0.1	88.5±0.4	88.5±0.3	77.9±0.1	77.5±0.1
+**MulRel** [25]	93.07±0.27	93.9±0.2	88.3±0.6	89.9±0.8	77.5±0.1	78.0±0.1
+**FGS2EE** [20]	92.63±0.14	<u>94.26±0.17</u>	88.47±0.23	<u>90.7±0.28</u>	77.41±0.21	77.66±0.23
+ DOC-AET	<u>93.29±0.18</u>	**94.55±0.11**	<u>88.96±0.41</u>	**91.27±0.14**	77.56±0.14	77.75±0.24

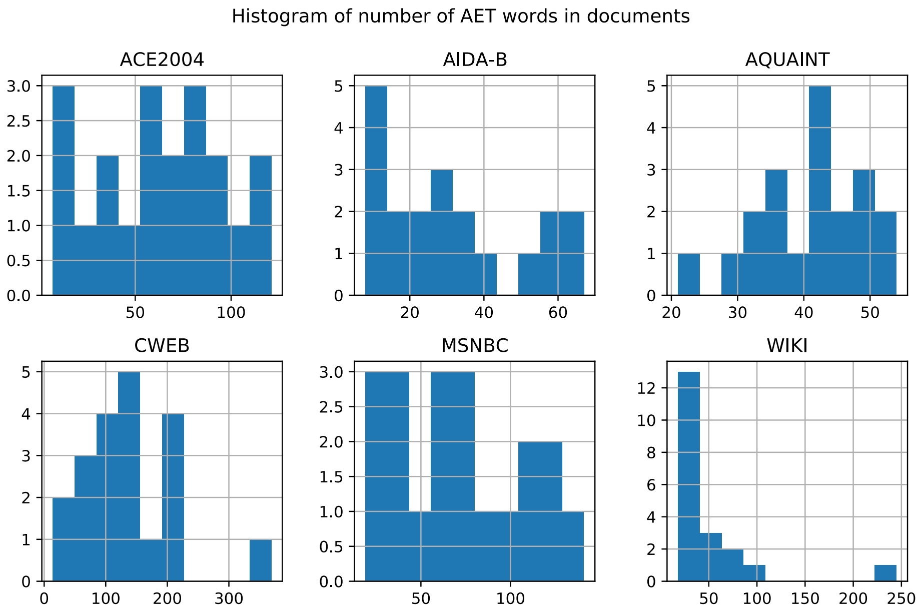


Fig. 4 Histogram of number of AET words in documents of the six datasets

It is seen that the method of [26] outperformed our model on three out-domain test sets: **AQUAINT**, **CWEB**, and **WIKI**. But it should be noted that **CWEB** and **WIKI** are believed to be less reliable [13], as they are automatically extracted (all entity linking systems perform comparatively poor on both test sets). Moreover, their model is trained on a large training set with 30,000 documents, while the AIDA-CoNLL training set only has 946 documents. Our method only extracts the inter-paragraph co-occurrence of 3,140 words.

We find that the effectiveness of our method might be affected by the number of AET words in documents. As shown in Fig. 4, most documents in CWEB contain AET words more than 100, while most documents in WIKI contain AET words less than 30. Our method performs worse on these two datasets than on the other datasets (most documents in the other datasets have 30-100 AET words). We conjecture that too many AET words might introduce noisy information for entity linking; and too scarce AET words might cannot provide sufficient global clues for entity linking.

Our method simply extracts AET words from the whole document. Thus, our method lacks the deep understanding of the document. There are concurrent studies using BERT to generate deeper understanding of context for entity linking. For example, the BERT-based entity linking model by Yamada et al. [41] is trained by predicting randomly masked entities on a large corpus extracted from Wikipedia. Even though our method is much simpler, our method outperforms Yamada et al. [41] on two datasets: MSNBC and ACE 2004. Obviously, it is because they used Wikipedia to train the model that their entity linking method performs especially well on WIKI.

The AET words extracted from documents contain some nominal coreferences (e.g., the company) of entities. Thus, it is fair to say that our method performs implicit and fuzzy coreference resolution for entity linking. Explicitly performing high-quality coreference resolution will improve the accuracy of entity linking, as shown by the joint model for entity recognition, resolution and linking by Durrett and Klein [8]. However, the scarcity of annotated parallel

Table 3 Ablation analysis on the effectiveness of our proposed AET coherence scores. The "Average F1" denotes the averaged F1 on the five out-domain test sets

Linking methods	Average F1
DeepEd [13]	85.22
+MulRel [25]	85.5
+FGS2EE [20]	85.7
+ DOC-AET	**86.02**

datasets for multiple tasks (e.g., coreference resolution and entity linking) means the joint model cannot be trained sufficiently. We believe that fine-tuning a pre-trained BERT-based model alternately on datasets of individual tasks can alleviate the scarcity of parallel datasets.

7.6 Ablation analysis

As we mentioned, our research can be seen as novel but along the line of research by [13], [25] and [20]. Thus, we perform ablation analysis to gauge the contributions of our research.

The method of **DeepEd** [13] is the first to leverage learned neural representations instead of manually designed features. Their deep learning architecture for NEL combines entity embeddings, a neural attention mechanism over local context windows, and unrolled differentiable message passing for global inference. The **MulRel** method [25] improved **DeepEd** by modeling latent multiple relations between textual mentions, i.e., the coherence scores of entity candidates are computed using Eq. 7 instead of Eq. 6. Our previous work [20] injects fine-grained semantic information into entity embeddings to facilitate the learning of contextual commonality.

We use the average F1 score on the five out-domain test sets to conduct ablation comparison, as listed in Table 3. The **MulRel** improved the average F1 on the five out-domain test sets by +0.28. Using the semantic reinforced entity embeddings of **FGS2EE** boost the average F1 by +0.2. Incorporating the coherence score between entity candidates and anonymous entity mentions improved the average F1 by +0.32.

7.7 Model complexity

Compared with the model of **MulRel** [25], our model added the following 200 parameters: (1) 100 parameters are the diagonal matrix **A** in Eq. 10; (2) 100 more parameters are integrated in the f function in Eq. 11 to incorporate the AET coherence score $\Psi(e_i, D)$.

Thus, the complexity of our model should be slightly more expensive than **MulRel** [25] and **DeepEd** [13]. However, our model converges faster than **MulRel**: on average our model needs 80 epochs, while **MulRel** needs 120 epochs and **DeepEd** needs 1250 epochs. In terms of wall clock time, our model requires less than 1 hour to train on a single GTX 1080, and the difference in training time between our model and **MulRel** is negligible. The training of **MulRel** is ten times faster than that of **DeepEd**.

7.8 AET word embeddings evaluation

Our AET word embeddings are learnt from AET words' inter-paragraph co-occurrence; thus, they can capture the related anonymous entity mentions from longer contexts that may span

Table 4 Cosine similarity between "*investor*" and other AET words using different embeddings

AET Words	AET similarity	GloVe similarity	Word2Vec similarity
Investment	0.9381	0.7935	0.6319
Stock	0.9341	0.4737	0.4529
Trading	0.9236	0.5374	0.3381
Equity	0.9230	0.7325	0.5259
Market	0.9098	0.5695	0.4209
Finance	0.8875	0.5504	0.3015
Fund	0.8851	0.6151	0.3248
Bank	0.8829	0.5119	0.2584
Portfolio	0.8826	0.5637	0.3864
Firm	0.8816	0.4944	0.3218

several paragraphs. Such AET word embeddings are used to compute the coherence scores between entity candidates and other anonymous entity mentions in the same document.

In contrast, the GloVe [33] and Word2Vec embeddings are learnt from the local context. Such embeddings can only be used to compute the coherence between entity candidates and local context (Equation 2).

To demonstrate the difference between our AET word embeddings and GloVe/Word2Vec, we list the cosine similarities between *investor* and other AET words using different word embeddings in Table 4. The words in the left column are the top-10 most similar words of *investor* using AET embeddings. The documents they appear in are similar to the documents where *investor* appears; thus, the AET cosine similarities are higher. In contrast, the local contexts where they appear are different; thus, the GloVe and Word2Vec cosine similarities are smaller and cannot capture the document-level relatedness of anonymous type words.

8 Conclusion

In this paper, we investigate using the more fine-grained anonymous entity type (AET) words as anonymous entity mentions for named entity linking. We propose a method, DOCument-level AET words relatedness (DOC-AET), to: (i) learn AET word embeddings to encode the document-level relatedness of AET words; and (ii) incorporate a new coherence score by exploiting the document-level relatedness of AET words.

DOC-AET exploits coherence between named entity mentions (represented by AET words) and contextual AET words/mentions for improving entity linking. We show that incorporating the coherence score between candidate entities and AET mentions can significantly improve NEL performance. DOC-AET used the fine-grained type words of our previous work [20] as AET vocabulary to extract anonymous entity mentions. The document-level relatedness between AET words is encoded into the AET word embeddings which are learnt from the AET words' inter-paragraph co-occurrence matrix. AET entity embeddings and document AET context embeddings are computed using the AET word embeddings. The coherence scores between candidate entities and anonymous entities are computed using the AET entity embeddings and document context embeddings. By incorporating such coherence scores for candidate ranking, we achieve state-of-the-art performance on three of the five out-domain datasets.

For future work, we plan to apply the document-level entity type coherence to the task of fine-grained entity typing. We will also explore the methods to incorporate deeper understanding of context, for example, by using the BERT embeddings of the AET words.

Acknowledgements This work is supported by the 2020 Catalyst: Strategic NZ-Singapore Data Science Research Programme Fund, MBIE, New Zealand.

References

1. Arora S, Li Y, Liang Y, Ma T, Risteski A (2018) Linear algebraic structure of word senses, with applications to polysemy. Trans Assoc Comput Linguist 6:483–495. https://doi.org/10.1162/tacl_a_00034
2. Bhowmik R, de Melo G (2018) Generating fine-grained open vocabulary entity type descriptions. In: Proceedings of the 56th annual meeting of the association for computational linguistics. https://doi.org/10.18653/v1/P18-1081
3. Chen S, Wang J, Jiang F, Lin CY (2020) Improving entity linking by modeling latent entity type information. In: Proceedings of the 34th AAAI conference on artificial intelligence, pp 7529–7537 . https://doi.org/10.1609/aaai.v34i05.6251
4. Cheng X, Roth D (2013) Relational inference for wikification. In: Proceedings of the 2013 conference on empirical methods in natural language processing, pp 1787–1796. https://www.aclweb.org/anthology/D13-1184
5. Chisholm A, Hachey B (2015) Entity disambiguation with web links. Trans Assoc Comput Linguist 3:145–156. https://doi.org/10.1162/tacl_a_00129
6. Choi E, Levy Omer, Choi Yejin, Zettlemoyer Luke (2018) Ultra-fine entity typing. In: Proceedings of the 56th annual meeting of the association for computational linguistics. Assoc Comput Linguist, pp 87–96. https://doi.org/10.18653/v1/P18-1009
7. Cui H, Peng T, Feng L, Bao T, Liu L (2021) Simple question answering over knowledge graph enhanced by question pattern classification. Knowl Inf Syst 63(10):2741–2761
8. Durrett G, Klein D (2014) A joint model for entity analysis: coreference, typing, and linking. Trans Assoc Comput Linguist
9. Ensan F, Du W (2019) Ad hoc retrieval via entity linking and semantic similarity. Knowl Inf Syst 58(3):551–583
10. Gabrilovich Evgeniy, Ringgaard Michael, Subramanya Amarnag (2013) FACC1: Freebase annotation of ClueWeb corpora, version 1 (release date 2013-06-26, format version 1, correction level 0)
11. Fang W, Zhang J, Wang D, Chen Z, Li M (2016) Entity disambiguation by knowledge and text jointly embedding. In: Proceedings of The 20th SIGNLL conference on computational natural language learning, pp 260–269. Association for computational linguistics, Berlin, Germany. https://doi.org/10.18653/v1/K16-1026.https://www.aclweb.org/anthology/K16-1026
12. Ganea OE, Ganea M, Lucchi A, Eickhof, C, Hofmann T (2016) Probabilistic bag-of-hyperlinks model for entity linking. In: Proceedings of the 25th international conference on world wide web, pp 927–938. International World Wide Web Conferences Steering Committee
13. Ganea OE, Hofmann T (2017) Deep joint entity disambiguation with local neural attention. In: Proceedings of the 2017 conference on empirical methods in natural language processing, pp 2619–2629. Association for Computational Linguistics, Copenhagen, Denmark . https://doi.org/10.18653/v1/D17-1277.https://www.aclweb.org/anthology/D17-1277
14. Gillick D, Lazic N, Ganchev K, Kirchner J, Huynh D (2014) Contextdependent fine-grained entity type tagging. arXiv preprint arXiv:1412.1820
15. Globerson A, Lazic N, Chakrabarti S, Subramanya A, Ringaard M, Pereira F (2016) Collective entity resolution with multi-focal attention. In: Proceedings of the 54th annual meeting of the association for computational linguistics (vol 1: Long Papers), pp 621–631. https://doi.org/10.18653/v1/P16-1059.https://www.aclweb.org/anthology/P16-1059
16. Guo Z, Barbosa D (2018) Robust named entity disambiguation with random walks. Sem Web (Preprint) 9(4):459–479
17. Gupta N, Singh S, Roth D (2017) Entity linking via joint encoding of types, descriptions, and context. In: Proceedings of the 2017 conference on empirical methods in natural language processing, pp 2681–2690. Association for Computational Linguistics, Copenhagen, Denmark . https://doi.org/10.18653/v1/D17-1284.https://www.aclweb.org/anthology/D17-1284
18. Hoffart J, Yosef MA, Bordino I, Fürstenau H, Pinkal M, Spaniol M, Taneva B, Thater S, Weikum G (2011) Robust disambiguation of named entities in text. In: Proceedings of the 2011 conference on empirical

methods in natural language processing, pp 782–792. Association for Computational Linguistics . http://www.aclweb.org/anthology/D11-1072
19. Hoffmann R, Zhang C, Ling X, Zettlemoyer L, Weld DS (2011) Knowledge-based weak supervision for information extraction of overlapping relations. In: Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies, pp 541–550. Association for Computational Linguistics, Portland, Oregon, USA. https://www.aclweb.org/anthology/P11-1055
20. Hou F, Wang R, He J, Zhou Y (2020) Improving entity linking through semantic reinforced entity embeddings. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 6843–6848. Association for Computational Linguistics, Online . https://doi.org/10.18653/v1/2020.acl-main.612.https://www.aclweb.org/anthology/2020.acl-main.612
21. Hou F, Wang R, Zhou Y (2021) Transfer learning for fine-grained entity typing. Knowl Inf Syst 63(4):845–866
22. Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980
23. Kouki P, Pujara J, Marcum C, Koehly L, Getoor L (2019) Collective entity resolution in multi-relational familial networks. Knowl Inf Syst 61(3):1547–1581
24. Lazic N, Subramanya A, Ringgaard M, Pereira F (2015) Plato: a selective context model for entity resolution. Trans Assoc Comput Linguist 3:503–515. https://doi.org/10.1162/tacl_a_00154https://www.aclweb.org/anthology/Q15-1036
25. Le P, Titov I (2018) Improving entity linking by modeling latent relations between mentions. In: Proceedings of the 56th annual meeting of the association for computational linguistics (vol 1: Long Papers), pp 1595–1604. Association for Computational Linguistics, Melbourne, Australia . https://doi.org/10.18653/v1/P18-1148.https://www.aclweb.org/anthology/P18-1148
26. Le P, Titov I (2019) Boosting entity linking performance by leveraging unlabeled documents. In: Proceedings of the 57th annual meeting of the association for computational linguistics, pp 1935–1945. Association for Computational Linguistics, Florence, Italy. https://www.aclweb.org/anthology/P19-1187
27. Ling X, Weld DS (2012) Fine-grained entity recognition. In: Proceedings of association for the advancement of artificial intelligence
28. Liu M, Zhao Y, Qin B, Liu T (2019) Collective entity linking: a random walk-based perspective. Knowl Inf Syst 60(3):1611–1643
29. Mikolov T, Sutskever Ilya, Chen Kai, Corrado Greg, Dean Jeffrey (2013) Distributed representations of words and phrases and their compositionality. In: Adv Neural Inf Process Syst, pp 3111–3119
30. Milne D, Witten IH (2008) Learning to link with wikipedia. In: Proceedings of the 17th ACM conference on information and knowledge management, pp 509–518. ACM
31. Mu J, Bhat S, Viswanath P (2017) Geometry of polysemy. In: Proceedings of the 5th international conference on learning representations
32. Murphy KP (2012) Machine learning: a probabilistic perspective. MIT press
33. Pennington J, Socher R, Manning C (2014) GloVe: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1532–1543. Association for Computational Linguistics, Doha, Qatar. https://doi.org/10.3115/v1/D14-1162.https://www.aclweb.org/anthology/D14-1162
34. Phan MC, Sun A, Tay Y, Han J, Li C (2019) Pair-linking for collective entity disambiguation: two could be better than all. IEEE Trans Knowl Data Eng 31(7):1383–1396
35. Ratinov L, Roth D, Downey D, Anderson M (2011) Local and global algorithms for disambiguation to wikipedia. In: Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies, vol 1, pp 1375–1384. Association for Computational Linguistics. https://www.aclweb.org/anthology/P11-1138
36. Shen W, Wang J, Han J (2015) Entity linking with a knowledge base: issues, techniques, and solutions. IEEE Trans Knowl Data Eng 27(2):443–460
37. Yaghoobzadeh Y, Adel H, Schutze H (2018) Corpus-level fine-grained entity typing. J Artif Intell Res 61:835–862
38. Yaghoobzadeh Y, Kann K, Hazen TJ, Agirre E, Schütze H (2019) Probing for semantic classes: Diagnosing the meaning content of word embeddings. In: Proceedings of the 57th Annual meeting of the association for computational linguistics, pp 5740–5753. Association for Computational Linguistics, Florence, Italy. https://www.aclweb.org/anthology/P19-1574
39. Yamada I, Shindo H, Takeda H, Takefuji Y (2016) Joint learning of the embedding of words and entities for named entity disambiguation. In: Proceedings of The 20th SIGNLL conference on computational natural language learning, pp 250–259. Association for Computational Linguistics, Berlin, Germany. https://doi.org/10.18653/v1/K16-1025.https://www.aclweb.org/anthology/K16-1025
40. Yamada I, Shindo H, Takeda H, Takefuji Y (2017) Learning distributed representations of texts and entities from knowledge base. Trans Assoc Comput Linguist 5:397–411

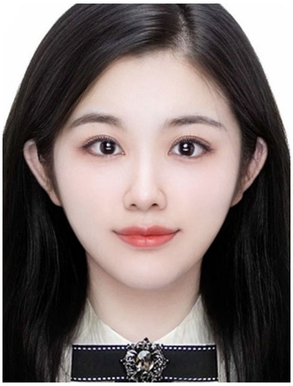
41. Yamada I, Washio K, Shindo H, Matsumoto Y (2020) Global entity disambiguation with pretrained contextualized embeddings of words and entities. arXiv preprint arXiv:1909.00426
42. Yang X, Gu X, Lin S, Tang S, Zhuang Y, Wu F, Chen Z, Hu G, Ren X (2019) Learning dynamic context augmentation for global entity linking. In: Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP), pp 271–281. Association for Computational Linguistics, Hong Kong, China. https://doi.org/10.18653/v1/D19-1026.https://www.aclweb.org/anthology/D19-1026
43. Yih Wt, Chang MW, He X, Gao J (2015) Semantic parsing via staged query graph generation: Question answering with knowledge base. In: Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (vol 1: Long Papers), pp 1321–1331. Association for Computational Linguistics, Beijing, China. https://doi.org/10.3115/v1/P15-1128.https://www.aclweb.org/anthology/P15-1128
44. Zhou X, Miao Y, Wang W, Qin J (2020) A recurrent model for collective entity linking with adaptive features. In: Proceedings of the 34th AAAI Conference on Artificial Intelligence, vol. 34, pp. 329–336
45. Zwicklbauer S, Seifert C, Granitzer M (2016) Robust and collective entity disambiguation through semantic embeddings. In: Proceedings of the 39th international ACM SIGIR conference, pp 425–434. ACM

Feng Hou received the Ph.D. degree in Computer Science from Massey University, New Zealand. He is now a Research Fellow at Massey University, New Zealand. His research interests include machine learning methods for natural language processing.

Ruili Wang received the Ph.D. degree in Computer Science from Dublin City University, Dublin, Ireland. He is currently a Professor of Artificial Intelligence and Chair of Research in the School of Natural and Computational Sciences, Massey University, Auckland, New Zealand, where he is the Director of the Centre of Language and Speech Processing. His current research interests include speech processing, language processing, video processing, data mining, and intelligent systems. Dr. Wang serves as a member and an Associate Editor of the editorial boards for international journals, such as the journals of IEEE Transactions on Emerging Topics in Computational Intelligence, Knowledge and Information Systems and Applied Soft Computing.

See-Kiong Ng is a Professor of Practice at the School of Computing, National University of Singapore, and the Deputy Director of the university's Institute of Data Science. See-Kiong obtained his Ph.D. in Computer Science from Carnegie Mellon University and started his transdisciplinary research career in data science and AI as an early bioinformatician in the 1990s. He has since been applying what he had learned from bioinformatics to a wide array of real-life application domains for smart cities. See-Kiong is currently working with his NZ collaborators on natural language processing for QA in indigenous/vernacular languages.

Michael Witbrock is a Professor of Computer Science at Waipapa Taumata Rau, The University of Auckland. Michael's research group, the Strong AI Lab, integrates machine learning, reasoning, and natural language understanding, with an additional focus on maximizing the near-term benefit of AI to NZ entrepreneurs and business, and more generally achieving the best social and civilizational impacts of increasingly powerful AI. Michael obtained his Ph.D. in Computer Science from Carnegie Mellon, and he holds a B.Sc. (Hons) in Psychology from Otago University in NZ. Before joining the University, he was a Distinguished Research Staff Member at IBM T J Watson Research Center in Yorktown Heights, NY.



Fangyi Zhu is a research fellow at the National University of Singapore. Most of her research centers around techniques for question-answering, including question-answering for low-resource languages, i.e., Malay Bahasa and Thai. Additionally, her research interests include multi-modal learning, particularly exploring the connections between vision and language.

Xiaoyun Jia received her master degree from University of Auckland (New Zealand) and obtained her Ph.D from Massey University (New Zealand). With a diverse background, her current research interests focus on online behavior analysis, user studies, E-commerce, data mining, human–computer interactions, information systems, etc.