Neural Collective Entity Linking

Yixin Cao, Lei Hou,[*] Juanzi Li, Zhiyuan Liu
Dept. of Computer Science and Technology, Tsinghua University, China 100084
{caoyixin2011,greener2009,lijuanzi2008}@gmail.com
liuzy@tsinghua.edu.cn

Abstract

Entity Linking aims to link entity mentions in texts to knowledge bases, and neural models have achieved recent success in this task. However, most existing methods rely on local contexts to resolve entities independently, which may usually fail due to the data sparsity of local information. To address this issue, we propose a novel neural model for collective entity linking, named as NCEL. NCEL applies Graph Convolutional Network to integrate both local contextual features and global coherence information for entity linking. To improve the computation efficiency, we approximately perform graph convolution on a subgraph of adjacent entity mentions instead of those in the entire text. We further introduce an attention scheme to improve the robustness of NCEL to data noise and train the model on Wikipedia hyperlinks to avoid overfitting and domain bias. In experiments, we evaluate NCEL on five publicly available datasets to verify the linking performance as well as generalization ability. We also conduct an extensive analysis of time complexity, the impact of key modules, and qualitative results, which demonstrate the effectiveness and efficiency of our proposed method.

1 Introduction

Entity linking (EL), mapping entity mentions in texts to a given knowledge base (KB), serves as a fundamental role in many fields, such as question answering (Zhang et al., 2016), semantic search (Blanco et al., 2015), and information extraction (Ji et al., 2015; Ji et al., 2016). However, this task is non-trivial because entity mentions are usually ambiguous. As shown in Figure 1, the mention *England* refers to three entities in KB, and an entity linking system should be capable of identifying the correct entity as *England cricket team* rather than *England* and *England national football team*.

Entity linking is typically broken down into two main phases: (i) candidate generation obtains a set of referent entities in KB for each mention, and (ii) named entity disambiguation selects the possible candidate entity by solving a ranking problem. The key challenge lies in the ranking model that computes the relevance between candidates and the corresponding mentions based on the information both in texts and KBs (Nguyen et al., 2016). In terms of the features used for ranking, we classify existing EL models into two groups: **local models** to resolve mentions independently relying on textual context information from the surrounding words (Chen and Ji, 2011; Chisholm and Hachey, 2015; Lazic et al., 2015; Yamada et al., 2016), and **global (collective) models**, which are the main focus of this paper, that encourage the target entities of all mentions in a document to be topically coherent (Han et al., 2011; Cassidy et al., 2012; He et al., 2013b; Cheng and Roth, 2013; Durrett and Klein, 2014; Huang et al., 2014).

Global models usually build an entity graph based on KBs to capture coherent entities for all identified mentions in a document, where the nodes are entities, and edges denote their relations. The graph provides highly discriminative semantic signals (e.g., entity relatedness) that are unavailable to local model (Eshel et al., 2017). For example (Figure 1), an EL model seemly cannot find sufficient disambiguation clues for the mention *England* from its surrounding words, unless it utilizes the coherence

[*]Corresponding author.

information of consistent topic “cricket” among adjacent mentions *England*, *Hussain*, and *Essex*. Although the global model has achieved significant improvements, its limitation is threefold:

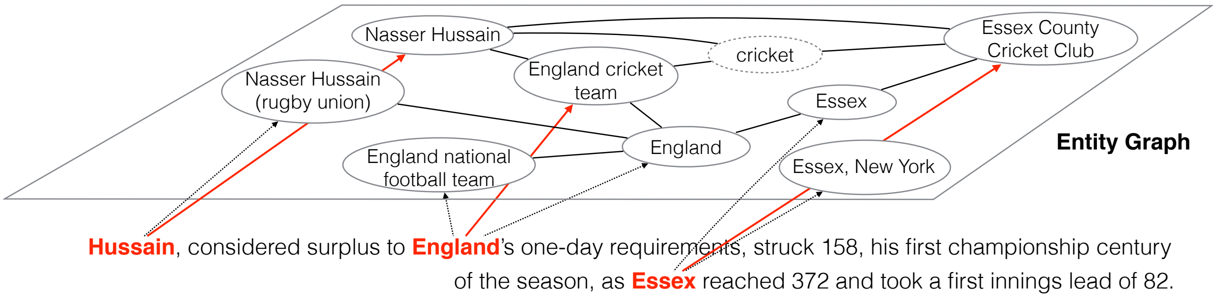
Nasser Hussain
Essex County Cricket Club
cricket
Nasser Hussain (rugby union)
England cricket team
Essex
England
Entity Graph
England national football team
Essex, New York
Hussain, considered surplus to England's one-day requirements, struck 158, his first championship century of the season, as Essex reached 372 and took a first innings lead of 82.

Figure 1: Illustration of named entity disambiguation for three mentions *England*, *Hussain*, and *Essex*. The nodes linked by arrowed lines are the candidate entities, where red solid lines denote target entities.

1. The global approach suffers from the data sparsity issue of unseen words/entities, and the failure to induce underlying discriminative features for EL.

2. The joint inference mechanism in the global approach leads to expensive computations, especially when the entity graph may contain hundreds of nodes in case of long documents.

3. The annotated EL training data is usually expensive to obtain or only available in narrow domains, which results in possible overfitting issue or domain bias.

To mitigate the first limitation, recent EL studies introduce neural network (NN) models due to its amazing feature abstraction and generalization ability. In such models, words/entities are represented by low dimensional vectors in a continuous space, and features for mention as well as candidate entities are automatically learned from data (Nguyen et al., 2016). However, existing NN-based methods for EL are either local models (Yamada et al., 2017; Gupta et al., 2017) or merely use word/entity embeddings for feature extraction and rely on another modules for collective disambiguation, which thus cannot fully utilize the power of NN models for collective EL (Globerson et al., 2016; Guo and Barbosa, 2017; Phan et al., 2018).

The second drawback of the global approach has been alleviated through approximate optimization techniques, such as PageRank/random walks (Pershina et al., 2015), graph pruning (Hoffart et al., 2011), ranking SVMs (Ratinov et al., 2011), or loopy belief propagation (LBP) (Globerson et al., 2016; Ganea and Hofmann, 2017). However, these methods are not differentiable and thus difficult to be integrated into neural network models (the solution for the first limitation).

To overcome the third issue of inadequate training data, (Gupta et al., 2017) has explored a massive amount of hyperlinks in Wikipedia, but these potential annotations for EL contain much noise, which may distract a naive disambiguation model (Chisholm and Hachey, 2015).

In this paper, we propose a novel **N**eural **C**ollective **E**ntity **L**inking model (NCEL), which performs global EL combining deep neural networks with Graph Convolutional Network (GCN) (Defferrard et al., 2016; Kipf and Welling, 2017) that allows flexible encoding of entity graphs. It integrates both local contextual information and global interdependence of mentions in a document, and is efficiently trainable in an end-to-end fashion. Particularly, we introduce attention mechanism to robustly model local contextual information by selecting informative words and filtering out the noise. On the other hand, we apply GCNs to improve discriminative signals of candidate entities by exploiting the rich structure underlying the correct entities. To alleviate the global computations, we propose to convolute on the subgraph of adjacent mentions. Thus, the overall coherence shall be achieved in a chain-like way via a sliding window over the document. To the best of our knowledge, this is the first effort to develop a unified model for neural collective entity linking.

In experiments, we first verify the efficiency of NCEL via theoretically comparing its time complexity with other collective alternatives. Afterwards, we train our neural model using collected Wikipedia hyperlinks instead of dataset-specific annotations, and perform evaluations on five public available benchmarks. The results show that NCEL consistently outperforms various baselines with a favorable generalization ability. Finally, we further present the performance on a challenging dataset WW (Guo and Barbosa, 2017) as well as qualitative results, investigating the effectiveness of each key module.

2 Preliminaries and Framework

We denote $M = \{m_i\}$ as a set of entity mentions in a document $D = \langle x_1, \ldots, x_i, \ldots, x_{|D|} \rangle$, where x_i is either a word w_i or a mention m_i. $G = (E, R)$ is the entity graph for document D derived from the given knowledge base, where $E = \{e_i\}$ is a set of entities, $R = \{r^i_j \in (0, 1]\}$ denotes the relatedness between $\langle e_i, e_j \rangle$ and higher values indicate stronger relations. Based on G, we extract a subgraph G^{ij} for $e_j \in \Phi(m_i)$, where $\Phi(m_i)$ denotes the set of candidate entities for m_i. Note that we don't include the relations among candidates of the same mention in G^{ij} because these candidates are mutually exclusive in disambiguation.

Formally, we define the entity linking problem as follows: Given a set of mentions M in a document D, and an entity graph G, the goal is to find an assignment[1] $\Gamma : M \rightarrow E$.

To collectively find the best assignment, NCEL aims to improve the discriminability of candidates' local features by using entity relatedness within a document via GCN, which is capable of learning a function of features on the graph through shared parameters over all nodes. Figure 2 shows the framework of NCEL including three main components:

1. **Candidate Generation**: we use a pre-built dictionary to generate a set of entities as candidates to be disambiguated for each mention, e.g., for mention *England*, we have $\Phi(m_i) = \{e^i_1, e^i_2, e^i_3\}$, in which the entities refer to *England national football team*, *England* and *England cricket team* in Figure 1, respectively.

2. **Feature Extraction**: based on the document and its entity graph, we extract both local features and global features for each candidate entity to feed our neural model. Concretely, local features reflect the compatibility between a candidate and its mention within the contexts, and global features are to capture the topical coherence among various mentions. These features, including vectorial representations of candidates and a subgraph indicating their relatedness, are highly discriminative for tackling ambiguity in EL.

3. **Neural Model**: given feature vectors and subgraphs of candidates, we first encode the features to represent nodes (i.e., candidates) in the graph, then improve them for disambiguation via multiple graph convolutions by exploiting the structure information, in which the features for correct candidates that are strongly connected (i.e., topical coherent) shall enhance each other, and features for incorrect candidates are weakened due to their sparse relations. Finally, we decode the features of nodes to output a probability indicating how possible the candidate refers to its mention.

Example As shown in Figure 2, for the current mention *England*, we utilize its surrounding words as local contexts (e.g., *surplus*), and adjacent mentions (e.g., *Hussian*) as global information. Collectively, we utilize the candidates of *England* $e^i_j \in \Phi(m_i), j = 1, 2, 3$ as well as those entities of its adjacencies $\Phi(m_{i-1}) \cup \Phi(m_{i+1})$ to construct feature vectors for e^i_j and the subgraph of relatedness as inputs of our neural model. Let darker blue indicate higher probability of being predicted, the correct candidate e^i_3 becomes bluer due to its bluer neighbor nodes of other mentions m_{i-1}, m_{i+1}. The dashed lines denote entity relations that have indirect impacts through the sliding adjacent window , and the overall structure shall be achieved via multiple sub-graphs by traversing all mentions.

Before introducing our model, we first describe the component of candidate generation.

[1] Normally, an entity linking system outputs NIL for a mention when no assignment score is higher than a threshold. This is application-specific and thus outside of the scope of this work.

2.1 Candidate Generation

Similar to previous work (Ganea and Hofmann, 2017), we use the prior probability $\hat{p}(e_i|m_j)$ of entity e_i conditioned on mention m_j both as a local feature and to generate candidate entities: $\Phi(m_j) = \{e_i|\hat{p}(e_i|m_j) > 0\}$. We compute $\hat{p}(\cdot)$ based on statistics of mention-entity pairs from: (i) Wikipedia page titles, redirect titles and hyperlinks, (ii) the dictionary derived from a large Web Corpus (Spitkovsky and Chang, 2012), and (iii) the YAGO dictionary with a uniform distribution (Hoffart et al., 2011). We pick up the maximal prior if a mention-entity pair occurs in different resources. In experiments, to optimize for memory and run time, we keep only top n entities based on $\hat{p}(e_i|m_j)$. In the following two sections, we will present the key components of NECL, namely feature extraction and neural network for collective entity linking.

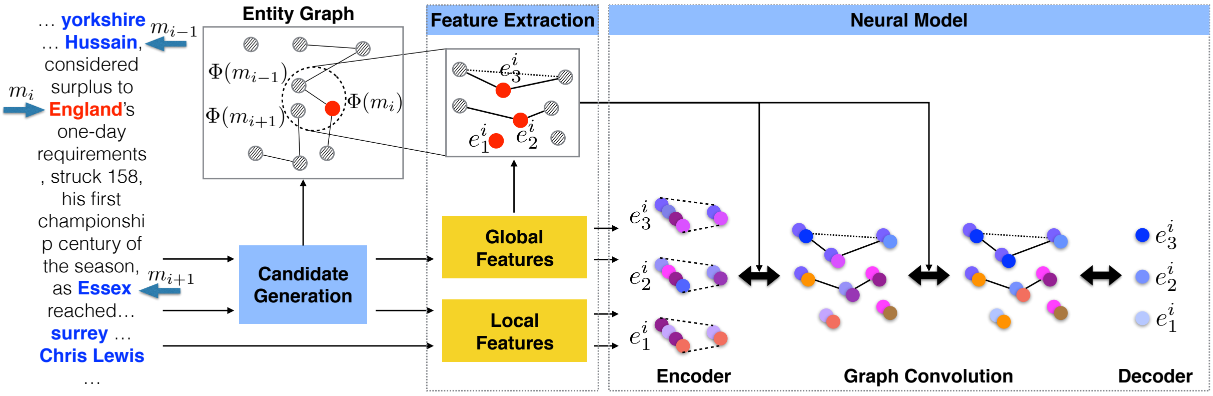

Figure 2: Framework of NCEL. The inputs of a set of mentions in a document are listed in the left side. The words in red indicate the current mention m_i, where m_{i-1}, m_{i+1} are neighbor mentions, and $\Phi(m_i) = \{e_1^i, e_2^i, e_3^i\}$ denotes the candidate entity set for m_i.

3 Feature Extraction

The main goal of NCEL is to find a solution for collective entity linking using an end-to-end neural model, rather than to improve the measurements of local textual similarity or global mention/entity relatedness. Therefore, we use joint embeddings of words and entities at sense level (Cao et al., 2017) to represent mentions and its contexts for feature extraction. In this section, we give a brief description of our embeddings followed by our features used in the neural model.

3.1 Learning Joint Embeddings of Word and Entity

Following (Cao et al., 2017), we use Wikipedia articles, hyperlinks, and entity outlinks to jointly learn word/mention and entity embeddings in a unified vector space, so that similar words/mentions and entities have similar vectors. To address the ambiguity of words/mentions, (Cao et al., 2017) represents each word/mention with multiple vectors, and each vector denotes a sense referring to an entity in KB. The quality of the embeddings is verified on both textual similarity and entity relatedness tasks.

Formally, each word/mention has a global embedding $\mathbf{w}_i/\mathbf{m}_i$, and multiple sense embeddings $\mathcal{S}(x_i) = \{\mathbf{s}_j\}$. Each sense embedding $\mathbf{s}_j$ refers to an entity embedding $\mathbf{e}_j$, while the difference between $\mathbf{s}_j$ and $\mathbf{e}_j$ is that $\mathbf{s}_j$ models the co-occurrence information of an entity in texts (via hyperlinks) and $\mathbf{e}_j$ encodes the structured entity relations in KBs. More details can be found in the original paper.

3.2 Local Features

Local features focus on how compatible the entity is mentioned in a piece of text (i.e., the mention and the context words). Except for the prior probability (Section 2.1), we define two types of local features for each candidate entity $e_j \in \Phi(m_i)$:

String Similarity Similar to (Yamada et al., 2017), we define string based features as follows: the edit distance between mention's surface form and entity title, and boolean features indicating whether they are equivalent, whether the mention is inside, starts with or ends with entity title and vice versa.

Compatibility We also measure the compatibility of e_j with the mention's context words $\mathcal{C}(m_i)$ by computing their similarities based on joint embeddings: $sim(\mathbf{e}_j, \mathbf{c}_{m_i,e_j})$ and $sim(\mathbf{s}_j, \mathbf{c}_{m_i,e_j})$, where $\mathbf{c}_{m_i,e_j}$ is the context embedding of m_i conditioned on candidate e_j and is defined as the average sum of word global vectors weighted by attentions:

$$\mathbf{c}_{m_i,e_j} = \sum_{w_k \in \mathcal{C}(m_i)} \alpha_{kj} \mathbf{w}_k$$

where α_{kj} is the k-th word's attention from e_j. In this way, we automatically select informative words by assigning higher attention weights, and filter out irrelevant noise through small weights. The attention α_{kj} is computed as follows:

$$\alpha_{kj} \propto sim(\mathbf{w}_k, \mathbf{e}_j)$$

where sim is the similarity measurement, and we use cosine similarity in the presented work. We concatenate the prior probability, string based similarities, compatibility similarities and the embeddings of contexts as well as the entity as the local feature vectors.

3.3 Global Features

The key idea of collective EL is to utilize the topical coherence throughout the entire document. The consistency assumption behind it is that: *all mentions in a document shall be on the same topic*. However, this leads to exhaustive computations if the number of mentions is large. Based on the observation that the consistency attenuates along with the distance between two mentions, we argue that the adjacent mentions might be sufficient for supporting the assumption efficiently.

Formally, we define neighbor mentions as q adjacent mentions before and after current mention m_i: $\mathcal{N}(m_i) = \{m_{i-q}, \ldots, m_{i-1}, m_{i+1}, \ldots, m_{i+q}\}$, where $2q$ is the pre-defined window size. Thus, the topical coherence at document level shall be achieved in a chain-like way. As shown in Figure 2 ($q = 1$), mentions *Hussain* and *Essex*, a cricket player and the cricket club, provide adequate disambiguation clues to induce the underlying topic "cricket" for the current mention *England*, which impacts positively on identifying the mention *surrey* as another cricket club via the common neighbor mention *Essex*.

A degraded case happens if q is large enough to cover the entire document, and the mentions used for global features become the same as the previous work, such as (Pershina et al., 2015). In experiments, we heuristically found a suitable $q = 3$ which is much smaller than the total number of mentions. The benefits of efficiency are in two ways: (i) to decrease time complexity, and (ii) to trim the entity graph into a fixed size of subgraph that facilitates computation acceleration through GPUs and batch techniques, which will be discussed in Section 5.2.

Given neighbor mentions $\mathcal{N}(m_i)$, we extract two types of vectorial global features and structured global features for each candidate $e_j \in \Phi(m_i)$:

Neighbor Mention Compatibility Suppose neighbor mentions are topical coherent, a candidate entity shall also be compatible with neighbor mentions if it has a high compatibility score with the current mention, otherwise not. That is, we extract the vectorial global features by computing the similarities between e_j and all neighbor mentions: $\{sim(\mathbf{e}_j, \mathbf{m}_i) | m_i \in \mathcal{N}(m_i)\}$, where $\mathbf{m}_i$ is the mention embedding by averaging the global vectors of words in its surface form: $\mathbf{m}_j = \sum_{w_l \in \mathcal{T}(m_j)} \mathbf{w}_l$, where $\mathcal{T}(m_j)$ are tokenized words of mention m_j.

Subgraph Structure The above features reflect the consistent semantics in texts (i.e., mentions). We now extract structured global features using the relations in KB, which facilitates the inference among candidates to find the most topical coherent subset. For each document, we obtain the entity graph G by taking candidate entities of all mentions $\Phi(M)$ as nodes, and using entity embeddings to compute their similarities as edges $R = \{r_j^i | r_j^i = sim(\mathbf{e}_i, \mathbf{e}_j)\}$. Then, we extract the subgraph structured features $\mathbf{g}^{i*}$ for each entity $e_*^i \in \Phi(m_i), m_i \in M$ for efficiency.

Formally, we define the subgraph as: $G^{i*} = (e_*^i \bigcup \Phi(\mathcal{N}(m_i)), R^{i*})$, where $R^{i*} = \{r_{jk}^{i*} | e_k^j \in \Phi(m_j), j \in [i-q, i+q] \setminus i\}$. For example (Figure 1), for entity *England cricket team*, the subgraph contains the relation from it to all candidates of neighbor mentions: *England cricket team*, *Nasser Hussain (rugby union)*, *Nasser Hussain*, *Essex*, *Essex County Cricket Club* and *Essex, New York*. To support batch-wise acceleration, we represent G^{i*} in the form of adjacency table based vectors: $\mathbf{g}^{i*} = [r_{i-q,1}^{i*}, \cdots, r_{i+q,n}^{i*}]^T \in \mathbb{R}^{2qn}$, where n is the number of candidates per mention.

Finally, for each candidate e_j^i, we concatenate local features and neighbor mention compatibility scores as the feature vector $\mathbf{f}^{ij}$, and construct the subgraph structure representation $\mathbf{g}^{ij}$ as the inputs of NCEL.

4 Neural Collective Entity Linking

NCEL incorporates GCN into a deep neural network to utilize structured graph information for collectively feature abstraction, while differs from conventional GCN in the way of applying the graph. Instead of the entire graph, only a subset of nodes is "visible" to each node in our proposed method, and then the overall structured information shall be reached in a chain-like way. Fixing the size of the subset, NCEL is further speeded up by batch techniques and GPUs, and is efficient to large-scale data.

4.1 Graph Convolutional Network

GCNs are a type of neural network model that deals with structured data. It takes a graph as an input and output labels for each node. As a simplification of spectral graph convolutions, the main idea of (Kipf and Welling, 2017) is similar to a propagation model: to enhance the features of a node according to its neighbor nodes. The formulation is as follows:

$$H^{l+1} = \sigma(\tilde{A} H^l W^l)$$

where $\tilde{A}$ is a normalized adjacent matrix of the input graph with self-connection, H^l and W^l are the hidden states and weights in the l-th layer, and $\sigma(\cdot)$ is a non-linear activation, such as *ReLu*.

4.2 Model Architecture

As shown in Figure 2, NCEL identifies the correct candidate e_1^i for the mention m_i by using vectorial features as well as structured relatedness with candidates of neighbor mentions $\Phi(m_{i-1}), \Phi(m_{i+1})$. Given feature vector $\mathbf{f}^{ij} \in \mathbb{R}^{d_0}$ and subgraph representation $\mathbf{g}^{ij} \in \mathbb{R}^{2qn}$ of each candidate $e_j^i \in \Phi(m_i)$, we stack them as inputs for mention[2] m_i: $\mathbf{f} = [\mathbf{f}_{i1}, \cdots, \mathbf{f}_{in}]^T \in \mathbb{R}^{n \times d_0}$, and the adjacent matrix $A = [\hat{\mathbf{g}}^1, \cdots, \hat{\mathbf{g}}_n]^T \in \mathbb{R}^{n \times (2qn+1)}$, where $\hat{\mathbf{g}}^j = [\mathbf{g}^j, 1]^T \in \mathbb{R}^{2qn+1}$ denotes the subgraph with self-connection. We normalize A such that all rows sum to one, denoted as $\tilde{A}$, avoiding the change in the scale of the feature vectors.

Given $\mathbf{f}$ and $\tilde{A}$, the goal of NCEL is to find the best assignment:

$$\Gamma^*(m_i) = \underset{\hat{y}}{\text{argmax}}\ P(\hat{y}; \mathbf{f}, \tilde{A}, \omega)$$

where $\hat{y}$ is the output variable of candidates, and $P(\cdot)$ is a probability function as follows:

$$P(\hat{y}; \mathbf{f}, \tilde{A}, \omega) \propto \exp(F(\mathbf{f}, \tilde{A}, \hat{y}; \omega))$$

where $F(\mathbf{f}, \tilde{A}, \hat{y}; \omega)$ is the score function parameters by $\omega \in \mathbb{R}^{\omega}$. NCEL learns the mapping $F(\cdot)$ through a neural network including three main modules: encoder, sub-graph convolution network (sub-GCN) and decoder. Next, we introduce them in turn.

Encoder The function of this module is to integrate different features by a multi-layer perceptron (MLP):

$$h^1 = \sigma(\mathbf{f} W^1 + b^1)$$

[2] For clarity, we omit the superscripts indicating the mention.

where h^1 is the hidden states of the current mention, $W^1 \in \mathbb{R}^{d_0 \times d_1}$ and $b^1 \in \mathbb{R}^{d_1}$ are trainable parameters and bias. We use ReLu as the non-linear activation $\sigma(\cdot)$.

Sub-Graph Convolution Network Similar to GCN, this module learns to abstract features from the hidden state of the mention itself as well as its neighbors. Suppose $h^t_{m_k}$ is the hidden states of the neighbor m_k, we stack them to expand the current hidden states of m_i as $\tilde{h}^t \in \mathbb{R}^{(2qn+1) \times d_t}$, such that each row corresponds to that in the subgraph adjacent matrix $\tilde{A}$. We define sub-graph convolution as:

$$h^{t+1} = \sigma(\tilde{A}\tilde{h}^t W^t)$$

where $W^t \in \mathbb{R}^{d_t \times d_{t+1}}$ is a trainable parameter.

Decoder After T iterations of sub-graph convolution, the hidden states integrate both features of m_i and its neighbors. A fully connected decoder maps h^{t+1} to the number of candidates as follows:

$$F = h^{T+1} W^{T+1}$$

where $W^{T+1} \in \mathbb{R}^n$.

4.3 Training

The parameters of network are trained to minimize cross-entropy of the predicted and ground truth y^g:

$$\mathcal{L}_m = -\sum_{j=1}^{n} y^g_j log(P(\hat{y} = e_j; \mathbf{f}, \tilde{A}, \omega))$$

Suppose there are $D \in \mathcal{D}$ documents in training corpus, each document has a set of mentions M, leading to totally $M \in \mathcal{M}$ mention sets. The overall objective function is as follows:

$$\mathcal{L} = \sum_{M \in \mathcal{M}} \sum_{m \in M} \mathcal{L}_m$$

5 Experiments

To avoid overfitting with some dataset, we train NCEL using collected Wikipedia hyperlinks instead of specific annotated data. We then evaluate the trained model on five different benchmarks to verify the linking precision as well as the generalization ability. Furthermore, we investigate the effectiveness of key modules in NCEL and give qualitative results for comprehensive analysis[3].

5.1 Baselines and Datasets

We compare NCEL with the following state-of-the-art EL methods including three local models and three types of global models:

1. Local models: He (He et al., 2013a) and Chisholm (Chisholm and Hachey, 2015) beat many global models by using auto-encoders and web links, respectively, and NTEE (Yamada et al., 2017) achieves the best performance based on joint embeddings of words and entities.

2. Iterative model: AIDA (Hoffart et al., 2011) links entities by iteratively finding a dense subgraph.

3. Loopy Belief Propagation: Globerson (Globerson et al., 2016) and PBoH (Ganea et al., 2016) introduce LBP (Murphy et al., 1999) techniques for collective inference, and Ganea (Ganea and Hofmann, 2017) solves the global training problem via truncated fitting LBP.

4. PageRank/Random Walk: Boosting (Kulkarni et al., 2009), AGDISTISG (Usbeck et al., 2014), Babelfy (Moro et al., 2014), WAT (Piccinno and Ferragina, 2014), xLisa (Zhang and Rettinger, 2014) and WNED (Guo and Barbosa, 2017) performs PageRank (Page et al., 1999) or random walk (Tong et al., 2006) on the mention-entity graph and use the convergence score for disambiguation.

[3] Our codes can be found in `https://github.com/TaoMiner/NCEL`

For fairly comparison, we report the original scores of the baselines in the papers. Following these methods, we evaluate NCEL on the following five datasets: (1) **CoNLL-YAGO** (Hoffart et al., 2011): the CoNLL 2003 shared task including testa of 4791 mentions in 216 documents, and testb of 4485 mentions in 213 documents. (2) **TAC2010** (Ji et al., 2010): constructed for the Text Analysis Conference that comprises 676 mentions in 352 documents for testing. (3) **ACE2004** (Ratinov et al., 2011): a subset of ACE2004 co-reference documents including 248 mentions in 35 documents, which is annotated by Amazon Mechanical Turk. (4) **AQUAINT** (Milne and Witten, 2008): 50 news articles including 699 mentions from three different news agencies. (5) **WW** (Guo and Barbosa, 2017): a new benchmark with balanced prior distributions of mentions, leading to a hard case of disambiguation. It has 6374 mentions in 310 documents automatically extracted from Wikipedia.

5.2 Training Details and Running Time Analysis

Training We collect 50,000 Wikipedia articles according to the number of its hyperlinks as our training data. For efficiency, we trim the articles to the first three paragraphs leading to 1,035,665 mentions in total. Using CoNLL-Test A as the development set, we evaluate the trained NCEL on the above benchmarks. We set context window to 20, neighbor mention window to 6, and top $n = 10$ candidates for each mention. We use two layers with 2000 and 1 hidden units in MLP encoder, and 3 layers in sub-GCN. We use early stop and fine tune the embeddings. With a batch size of 16, nearly 3 epochs cost less than 15 minutes on the server with 20 core CPU and the GeForce GTX 1080Ti GPU with 12Gb memory. We use standard Precision, Recall and F1 at mention level (Micro) and at the document level (Macro) as measurements.

Complexity Analysis Compared with local methods, the main disadvantage of collective methods is high complexity and expensive costs. Suppose there are k mentions in documents on average, among these global models, NCEL not surprisingly has the lowest time complexity $\mathcal{O}(T * kn^2)$ since it only considers adjacent mentions, where T is the number of sub-GCN layers indicating the iterations until convergence. AIDA has the highest time complexity k^3n^3 in worst case due to exhaustive iteratively finding and sorting the graph. The LBP and PageRank/random walk based methods achieve similar high time complexity of $\mathcal{O}(T * k^2n^2)$ mainly because of the inference on the entire graph.

5.3 Results on GERBIL

GERBIL (Usbeck et al., 2015) is a benchmark entity annotation framework that aims to provide a unified comparison among different EL methods across datasets including ACE2004, AQUAINT and CoNLL. We compare NCEL with the global models that report the performance on GERBIL.

Datasets	AGDISTIS	AIDA	Babelfy	WAT	xLisa	PBoH	WNED	NCEL
ACE2004	0.66	0.80	0.61	0.76	0.81	0.79	0.81	**0.88**
	0.78	0.89	0.76	0.85	0.88	0.86	**0.90**	0.89
AQUAINT	0.73	0.57	0.70	0.75	0.79	0.84	0.83	**0.87**
	0.59	0.56	0.70	0.76	0.77	0.83	0.83	**0.88**
CoNLL-Test A	0.56	0.74	0.74	0.78	0.52	**0.80**	0.79	0.79
	0.49	0.71	0.68	0.76	0.48	**0.77**	0.76	**0.77**
CoNLL-Test B	0.55	0.77	0.76	**0.80**	0.54	**0.80**	0.79	**0.80**
	0.54	0.78	0.70	**0.80**	0.53	0.79	0.79	**0.80**
Average	0.63	0.72	0.70	0.77	0.67	0.81	0.81	**0.84**
	0.60	0.74	0.71	0.79	0.67	0.81	0.82	**0.84**

Table 1: Micro F1 (above) and Macro F1 (bottom) on GERBIL.

As shown in Table 1, NCEL achieves the best performance in most cases with an average gain of 2% on Micro F1 and 3% Macro F1. The baseline methods also achieve competitive results on some datasets but fail to adapt to the others. For example, AIDA and xLisa perform quite well on ACE2004 but poorly on other datasets, or WAT, PBoH, and WNED have a favorable performance on CoNLL but lower values on ACE2004 and AQUAINT. Our proposed method performs consistently well on all datasets that demonstrates the good generalization ability.

5.4 Results on TAC2010 and WW

In this section, we investigate the effectiveness of NCEL in the “easy” and “hard” datasets, respectively. Particularly, TAC2010, which has two mentions per document on average (Section 5.1) and high prior probabilities of correct candidates (Figure 3), is regarded as the “easy” case for EL, and WW is the “hard” case since it has the most mentions with balanced prior probabilities (Guo and Barbosa, 2017). Besides, we further compare the impact of key modules by removing the following part from NCEL: global features (NCEL-local), attention (NCEL-noatt), embedding features (NCEL-noemb), and the impact of the prior probability (prior).

Table 2: Precision on WW

AIDA	Ganea	WNED	NCEL-local	NCEL
0.63	0.78	0.84	0.81	**0.86**

	Prec	Micro F1	Macro F1
Chisholm	0.81	-	-
He	0.81	-	-
NTEE	0.88	-	-
NCEL-local	0.89	0.89	0.88
AIDA	-	0.55	0.51
Babelfy	-	0.63	0.62
WAT	-	0.75	0.73
Globerson	-	0.84	-
Boosting	-	0.86	0.85
NCEL	**0.91**	**0.91**	**0.92**

Table 3: Results on TAC2010

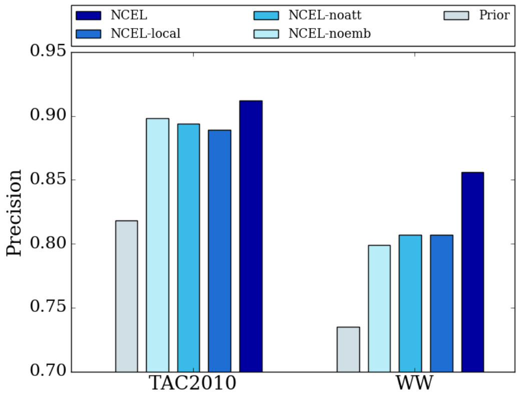

Figure 3: Impacts of NCEL modules

The results are shown in Table 2 and Table 3. We can see the average linking precision (Micro) of WW is lower than that of TAC2010, and NCEL outperforms all baseline methods in both easy and hard cases. In the “easy” case, local models have similar performance with global models since only little global information is available (2 mentions per document). Besides, NN-based models, NTEE and NCEL-local, perform significantly better than others including most global models, demonstrating that the effectiveness of neural models deals with the first limitation in the introduction.

Impact of NCEL Modules

As shown in Figure 3, the prior probability performs quite well in TAC2010 but poorly in WW. Compared with NCEL-local, the global module in NCEL brings more improvements in the “hard” case than that for “easy” dataset, because local features are discriminative enough in most cases of TAC2010, and global information becomes quite helpful when local features cannot handle. That is, our propose collective model is robust and shows a good generalization ability to difficult EL. The improvements by each main module are relatively small in TAC2010, while the modules of attention and embedding features show non-negligible impacts in WW (even worse than local model), mainly because WW contains much noise, and these two modules are effective in improving the robustness to noise and the ability of generalization by selecting informative words and providing more accurate semantics, respectively.

5.5 Qualitative Analysis

<table>
<tr><td colspan="4">Hussain, considered surplus to Englands one-day requirements, struck 158, his first championship century of the season, as Essex reached 372 and took a first innings lead of 82.</td></tr>
<tr><td colspan="2">NCEL</td><td colspan="2">NCEL-local</td></tr>
<tr><td>England</td><td>0.23</td><td>England</td><td>0.42</td></tr>
<tr><td>England cricket team</td><td>0.72</td><td>England cricket team</td><td>0.20</td></tr>
<tr><td>Essex County Cricket Club</td><td>0.99</td><td>Essex County Cricket Club</td><td>0.97</td></tr>
</table>

Table 4: Qualitative Analysis of the Example *England*.

The results of example in Figure 1 are shown in Table 4, which is from CoNLL testa dataset. For mention *Essex*, although both NCEL and NCEL-local correctly identify entity *Essex County Cricket*

Club, NCEL outputs higher probability due to the enhancement of neighbor mentions. Moreover, for mention *England*, NCEL-local cannot find enough disambiguation clues from its context words, such as *surplus* and *requirements*, and thus assigns a higher probability of 0.42 to the country *England* according to the prior probability. Collectively, NCEL correctly identifies England cricket team with a probability of 0.72 as compared with 0.20 in NCEL-local with the help of its neighbor mention *Essex*.

6 Conclusion

In this paper, we propose a neural model for collective entity linking that is end-to-end trainable. It applies GCN on subgraphs instead of the entire entity graph to efficiently learn features from both local and global information. We design an attention mechanism that endows NCEL robust to noisy data. Trained on collected Wikipedia hyperlinks, NCEL outperforms the state-of-the-art collective methods across five different datasets. Besides, further analysis of the impacts of main modules as well as qualitative results demonstrates its effectiveness.

In the future, we will extend our method into cross-lingual settings to help link entities in low-resourced languages by exploiting rich knowledge from high-resourced languages, and deal with NIL entities to facilitate specific applications.

7 Acknowledgments

The work is supported by National Key Research and Development Program of China (2017YFB1002101), NSFC key project (U1736204, 61661146007), and THUNUS NExT Co-Lab.

References

Roi Blanco, Giuseppe Ottaviano, and Edgar Meij. 2015. Fast and space-efficient entity linking for queries. In *WSDM*.

Yixin Cao, Lifu Huang, Heng Ji, Xu Chen, and Juan-Zi Li. 2017. Bridge text and knowledge by learning multi-prototype entity mention embedding. In *ACL*.

Taylor Cassidy, Heng Ji, Lev-Arie Ratinov, Arkaitz Zubiaga, and Hongzhao Huang. 2012. Analysis and enhancement of wikification for microblogs with context expansion. In *COLING*.

Zheng Chen and Heng Ji. 2011. Collaborative ranking: A case study on entity linking. In *EMNLP*.

Xiao Cheng and Dan Roth. 2013. Relational inference for wikification. In *EMNLP*.

Andrew Chisholm and Ben Hachey. 2015. Entity disambiguation with web links. *TACL*.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *NIPS*.

Greg Durrett and Dan Klein. 2014. A joint model for entity analysis: Coreference, typing, and linking. *TACL*.

Yotam Eshel, Noam Cohen, Kira Radinsky, Shaul Markovitch, Ikuya Yamada, and Omer Levy. 2017. Named entity disambiguation for noisy text. In *CoNLL*.

Octavian-Eugen Ganea and Thomas Hofmann. 2017. Deep joint entity disambiguation with local neural attention. In *EMNLP*.

Octavian-Eugen Ganea, Marina Ganea, Aurélien Lucchi, Carsten Eickhoff, and Thomas Hofmann. 2016. Probabilistic bag-of-hyperlinks model for entity linking. In *WWW*.

Amir Globerson, Nevena Lazic, Soumen Chakrabarti, Amarnag Subramanya, Michael Ringgaard, and Fernando Pereira. 2016. Collective entity resolution with multi-focal attention. In *ACL*.

Zhaochen Guo and Denilson Barbosa. 2017. Robust named entity disambiguation with random walks.

Nitish Gupta, Sameer Singh, and Dan Roth. 2017. Entity linking via joint encoding of types, descriptions, and context. In *EMNLP*.

Xianpei Han, Le Sun, and Jun Zhao. 2011. Collective entity linking in web text: a graph-based method. In *SIGIR*.

Zhengyan He, Shujie Liu, Mu Li, Ming Zhou, Longkai Zhang, and Houfeng Wang. 2013a. Learning entity representation for entity disambiguation. In *ACL*.

Zhengyan He, Shujie Liu, Yang Song, Mu Li, Ming Zhou, and Houfeng Wang. 2013b. Efficient collective entity linking with stacking. In *EMNLP*.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *EMNLP*.

Hongzhao Huang, Yunbo Cao, Xiaojiang Huang, Heng Ji, and Chin-Yew Lin. 2014. Collective tweet wikification based on semi-supervised graph regularization. In *ACL*.

Heng Ji, Ralph Grishman, Hoa Trang Dang, Kira Griffitt, and Joe Ellis. 2010. Overview of the tac 2010 knowledge base population track.

Heng Ji, Joel Nothman, Ben Hachey, and Radu Florian. 2015. Overview of tac-kbp2015 tri-lingual entity discovery and linking.

Heng Ji, Joel Nothman, H Trang Dang, and Sydney Informatics Hub. 2016. Overview of tac-kbp2016 tri-lingual edl and its impact on end-to-end cold-start kbp. *Proceedings of TAC*.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. *ICLR*.

Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan, and Soumen Chakrabarti. 2009. Collective annotation of wikipedia entities in web text. In *KDD*.

Nevena Lazic, Amarnag Subramanya, Michael Ringgaard, and Fernando Pereira. 2015. Plato: A selective context model for entity resolution. *TACL*.

David N. Milne and Ian H. Witten. 2008. Learning to link with wikipedia. In *CIKM*.

Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014. Entity linking meets word sense disambiguation: a unified approach. *TACL*.

Kevin P. Murphy, Yair Weiss, and Michael I. Jordan. 1999. Loopy belief propagation for approximate inference: An empirical study. In *UAI*.

Thien Huu Nguyen, Nicolas Fauceglia, Mariano Rodriguez-Muro, Oktie Hassanzadeh, Alfio Massimiliano Gliozzo, and Mohammad Sadoghi. 2016. Joint learning of local and global features for entity linking via neural networks. In *COLING*.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: Bringing order to the web.

Maria Pershina, Yifan He, and Ralph Grishman. 2015. Personalized page rank for named entity disambiguation. In *HLT-NAACL*.

Minh C. Phan, Aixin Sun, Yi Tay, Jialong Han, and Chenliang Li. 2018. Pair-linking for collective entity disambiguation: Two could be better than all. *CoRR*.

Francesco Piccinno and Paolo Ferragina. 2014. From tagme to wat: a new entity annotator. In *ERD@SIGIR*.

Lev-Arie Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 2011. Local and global algorithms for disambiguation to wikipedia. In *ACL*.

Valentin I. Spitkovsky and Angel X. Chang. 2012. A cross-lingual dictionary for english wikipedia concepts. In *LREC*.

Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. 2006. Fast random walk with restart and its applications. *ICDM*.

Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Michael Röder, Daniel Gerber, Sandro Coelho, Sören Auer, and Andreas Both. 2014. Agdistis - graph-based disambiguation of named entities using linked data. In *International Semantic Web Conference*.

Ricardo Usbeck, Michael Röder, Axel-Cyrille Ngonga Ngomo, Ciro Baron, Andreas Both, Martin Brümmer, Diego Ceccarelli, Marco Cornolti, Didier Cherix, Bernd Eickmann, et al. 2015. Gerbil: general entity annotator benchmarking framework. In *WWW*.

Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2016. Joint learning of the embedding of words and entities for named entity disambiguation. In *CoNLL*.

Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2017. Learning distributed representations of texts and entities from knowledge base. *TACL*.

Lei Zhang and Achim Rettinger. 2014. X-lisa: Cross-lingual semantic annotation. *PVLDB*.

Yuanzhe Zhang, Shizhu He, Kang Liu, and Jun Zhao. 2016. A joint model for question answering over multiple knowledge bases. In *AAAI*.