Toward Socially-Infused Information Extraction: Embedding Authors, Mentions, and Entities

Yi Yang
Georgia Institute of Technology
Atlanta, GA 30308, USA
yiyang@gatech.edu

Ming-Wei Chang
Microsoft Research
Redmond, WA 98052, USA
minchang@microsoft.com

Jacob Eisenstein
Georgia Institute of Technology
Atlanta, GA 30308, USA
jacobe@gatech.edu

Abstract

Entity linking is the task of identifying mentions of entities in text, and linking them to entries in a knowledge base. This task is especially difficult in microblogs, as there is little additional text to provide disambiguating context; rather, authors rely on an implicit common ground of shared knowledge with their readers. In this paper, we attempt to capture some of this implicit context by exploiting the social network structure in microblogs. We build on the theory of *homophily*, which implies that socially linked individuals share interests, and are therefore likely to mention the same sorts of entities. We implement this idea by encoding authors, mentions, and entities in a continuous vector space, which is constructed so that socially-connected authors have similar vector representations. These vectors are incorporated into a neural structured prediction model, which captures structural constraints that are inherent in the entity linking task. Together, these design decisions yield F1 improvements of 1%-5% on benchmark datasets, as compared to the previous state-of-the-art.

1 Introduction

Entity linking on short texts (e.g., Twitter messages) is of increasing interest, as it is an essential step for many downstream applications, such as market research (Asur and Huberman, 2010), topic detection and tracking (Mathioudakis and Koudas, 2010), and question answering (Yih et al., 2015). Tweet entity linking is a particularly difficult problem, because the short context around an entity mention is often insufficient for entity disambiguation. For example, as shown in Figure 1, the entity mention 'Giants' in tweet t_1 can refer to the NFL football team *New York Giants* or the MLB baseball team *San Francisco Giants*. In this example, it is impossible to disambiguate between these entities solely based on the individual text message.

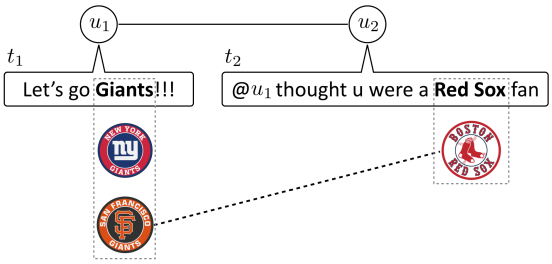

Figure 1: Illustration on leveraging social relations for entity disambiguation. Socially connected users u_1 and u_2 tend to talk about similar entities (baseball in the example).

We propose to overcome the difficulty and improve the entity disambiguation capability of the entity linking system by employing social network structures. The sociological theory of *homophily* asserts that socially connected individuals are more likely to have similar behaviors or share similar interests (McPherson et al., 2001). This property has been used to improve many natural language processing tasks such as sentiment analysis (Tan et al., 2011; Yang and Eisenstein, 2015), topic classification (Hovy, 2015) and user attribute inference (Li et al., 2015). We assume Twitter users will have similar interests in real world entities to their near neighbors — an assumption of *entity homophily* — which

is demonstrated in Figure 1. The social relation between users u_1 and u_2 may lead to more coherent topics in tweets t_1 and t_2. Therefore, by successfully linking the less ambiguous mention 'Red Sox' in tweet t_2 to the *Boston Red Sox* baseball team, the tweet entity linking system will be more confident on linking 'Giants' to the *San Francisco Giants* football team in tweet t_1.

To exploit social information, we adopt the recent advance on embedding information networks (Tang et al., 2015), which induces low-dimensional representations for author nodes based on the network structure. By learning the semantic interactions between the author embeddings and the pre-trained Freebase entity embeddings, the entity linking system can incorporate more disambiguating context from the social network. We also consider low-dimensional representations of mentions, another source of related information for entity linking, with the intuition that semantically related mentions can refer to similar entities. Previously proposed approaches (Guo et al., 2013a; Yang and Chang, 2015) are based on hand-crafted features and off-the-shelf machine learning algorithms. Our preliminary study suggests that simply augmenting the traditional surface features with the distributed representations barely improves the performance of these entity linking systems. Therefore, we propose NTEL, a **N**eural model for **T**weet **E**ntity **L**inking, to leverage the distributed representations of authors, mentions, and entities. NTEL can not only make efficient use of statistical surface features built from a knowledge base, but also learn the interactions between these distributed representations.

Our contributions are summarized as follows:

- We present a novel model for entity linking that exploits distributed representations of users, mentions, and entities.
- We combine this distributed model with a feed-forward neural network that learns non-linear combinations of surface features.
- We perform message-level inference using a dynamic program to avoid overlapping mentions. The architecture is trained with loss-augmented decoding, a large margin learning technique for structured prediction.
- The complete system, NTEL, outperforms the previous state-of-the-art (Yang and Chang, 2015) by 3% average F1 on two benchmark datasets.

Data	# Tweet	# Entity	Date
NEEL-train	2,340	2,202	Jul. - Aug. 2011
NEEL-test	1,164	687	Jul. - Aug. 2011
TACL	500	300	Dec. 2012

Table 1: Statistics of data sets.

2 Data

Two publicly available datasets for tweet entity linking are adopted in the work. NEEL is originally collected and annotated for the Named Entity Extraction & Linking Challenge (Cano et al., 2014), and TACL is first used and released by Fang and Chang (2014). The datasets are then cleaned and unified by Yang and Chang (2015). The statistics of the datasets are presented in Table 1.

3 Testing Entity Homophily

The hypothesis of *entity homophily*, as presented in the introduction, is that socially connected individuals are more likely to mention similar entities than disconnected individuals. We now test the hypothesis on real data before we start building our entity linking systems.

Twitter social networks We test the assumption on the users in the NEEL-train dataset. We construct three author social networks based on the follower, mention and retweet relations between the 1,317 authors in the NEEL-train dataset, which we refer as FOLLOWER, MENTION and RETWEET. Specifically, we use the Twitter API to crawl the friends of the NEEL users (individuals that they follow) and the mention/retweet links are induced from their most recent 3,200 tweets.[1] We exploit bi-directed links to create the undirected networks, as bi-directed links result in stronger social network ties than directed links (Kwak et al., 2010; Wu et al., 2011). The numbers of social relations for the networks are 1,604, 379 and 342 respectively.

[1] We are able to obtain at most 3,200 tweets for each Twitter user, due to the Twitter API limits.

Network	$sim(i \leftrightarrow j)$	$sim(i \not\leftrightarrow j)$
FOLLOWER	0.128	0.025
MENTION	0.121	0.025
RETWEET	0.173	0.025

Table 2: The average entity-driven similarity results for the networks.

Metrics We propose to use the *entity-driven similarity* between authors to test the hypothesis of entity homophily. For a user u_i, we employ a Twitter NER system (Ritter et al., 2011) to detect entity mentions in the timeline, which we use to construct a user entity vector $\mathbf{u}_i^{(ent)}$, so that $u_{i,j}^{(ent)} = 1$ iff user i has mentioned entity j.[2] The entity-driven similarity between two users u_i and u_j is defined as the cosine similarity score between the vectors $\mathbf{u}_i^{(ent)}$ and $\mathbf{u}_j^{(ent)}$. We evaluate the three networks by calculating the average entity-driven similarity of the connected user pairs and that of the disconnected user pairs, which we name as $sim(i \leftrightarrow j)$ and $sim(i \not\leftrightarrow j)$.

Results The entity-driven similarity results of these networks are presented in Table 2. As shown, $sim(i \leftrightarrow j)$ is substantially higher than $sim(i \not\leftrightarrow j)$ on all three social networks, indicating that socially connected individuals clearly tend to mention more similar entities than disconnected individuals. Note that $sim(i \not\leftrightarrow j)$ is approximately equal to the same base rate defined by the average entity-driven similarity of all pairs of users, because the vast majority of user pairs are disconnected, no matter how to define the network. Among the three networks, RETWEET offers slightly higher $sim(i \leftrightarrow j)$ than FOLLOWER and MENTION. The results verify our hypothesis of entity homophily, which forms the basis for this research. Note that all social relation data was acquired in March 2016; by this time, the authorship information of 22.1% of the tweets in the NEEL-train dataset was no longer available, because the tweets or user accounts had been deleted.

4 Method

In this section, we present, NTEL, a novel neural based tweet entity linking framework that is able to leverage social information. We first formally define the task of tweet entity linking. Assume we are given an entity database (e.g., Wikipedia or Freebase), and a lexicon that maps a surface form into a set of entity candidates. For each input tweet, we consider any n-grams of the tweet that match the lexicon as mention candidates.[3] The entity linking system maps every mention candidate (e.g., 'Red Sox') in the message to an entity (e.g., *Boston Red Sox*) or to **Nil** (i.e., not an entity). There are two main challenges in the problem. First, a mention candidate can often potentially link to multiple entities according to the lexicon. Second, as shown in Figure 2, many mention candidates overlap with each other. Therefore, the entity linking system is required to disambiguate entities and produce non-overlapping entity assignments with respect to the mention candidates in the tweet.

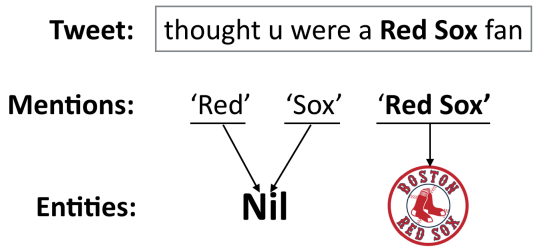

Figure 2: Illustration of the non-overlapping structure for the task of tweet entity linking. In order to link 'Red Sox' to a real entity, 'Red' and 'Sox' should be linked to **Nil**.

We formalize this task as a structured learning problem. Let $\mathbf{x}$ be the tweet, u be the author, and $\mathbf{y} = \{y_t\}_{t=1}^{T}$ be the entity assignments of the T mention candidates in the tweet. The overall scoring function $s(\mathbf{x}, \mathbf{y}, u)$ can be decomposed as follows,

$$s(\mathbf{x}, \mathbf{y}, u) = \sum_{t=1}^{T} g(\mathbf{x}, y_t, u, t), \qquad (1)$$

where $g(\mathbf{x}, y_t, u, t)$ is the scoring function for the t-th mention candidate choosing entity y_t. Note that the system needs to produce non-overlapping entity assignments, which will be resolved in the inference algorithm.

The overview of NTEL is illustrated in Figure 3. We further break down $g(\mathbf{x}, y_t, u, t)$ into two scoring

[2] We assume each name corresponds to a single entity for this metric, so this metric only approximates entity homophily.

[3] We adopted the same entity database and lexicon as those used by Yang and Chang (2015).

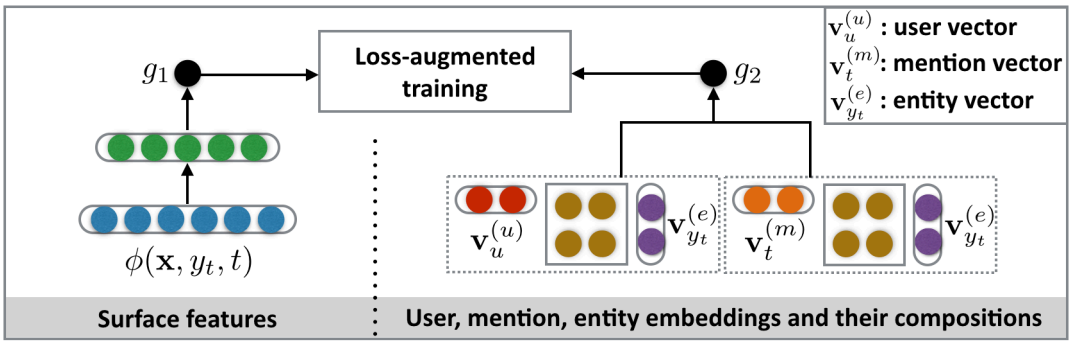

Figure 3: The proposed neural network approach for tweet entity linking. A composition model based on bilinear functions is used to learn the semantic interactions of user, mention, and entity.

functions:

$$g(\mathbf{x}, y_t, u, t; \Theta_1, \Theta_2) = g_1(\mathbf{x}, y_t, t; \Theta_1) + g_2(\mathbf{x}, y_t, u, t; \Theta_2), \quad (2)$$

where g_1 is the scoring function for our basic surface features, and g_2 is the scoring function for modeling user, mention, entity representations and their compositions. Θ_1 and Θ_2 are model parameters that will be detailed below. We choose to use a multilayer perceptron (MLP) to model $g_1(\mathbf{x}, y_t, t; \Theta_1)$, and we employ simple yet efficient bilinear functions to learn the compositions of user, mention, and entity representations $g_2(\mathbf{x}, y_t, u, t; \Theta_2)$. Finally, we present a training algorithm based on loss-augmented decoding and a non-overlapping inference algorithm.

4.1 Modeling Surface Features

We include the 37 features used by Yang and Chang (2015) as our surface feature set. These features are extracted from various sources, including a named entity recognizer, an entity type recognizer, and some statistics of the Wikipedia pages.

We exploit a multilayer perceptron (MLP) to transform the surface features to a real-valued score. The output of the MLP is formalized as follows,

$$g_1(\mathbf{x}, y_t, t; \Theta_1) = \boldsymbol{\beta}^\top \mathbf{h} + b$$
$$\mathbf{h} = \tanh(\mathbf{W}\phi(\mathbf{x}, y_t, t) + \mathbf{b}), \quad (3)$$

where $\phi(\mathbf{x}, y_t, t)$ is the feature function, $\mathbf{W}$ is an $M \times D$ matrix, the weights $\mathbf{b}$ are bias terms, and $\mathbf{h}$ is the output of the hidden layer of the MLP. $\boldsymbol{\beta}$ is an M dimensional vector of weights for the output score, and b is the bias term. The parameters of the MLP are $\Theta_1 = \{\mathbf{W}, \mathbf{b}, \boldsymbol{\beta}, b\}$. Yang and Chang (2015) argue that non-linearity is the key for obtaining good results on the task, as linear models are not expressive enough to capture the high-order relationships between the dense features. They propose a tree-based non-linear model for the task. The MLP forms simple non-linear mappings between the input features and the output score, whose parameters will be jointly learnt with other components in NTEL.

4.2 Modeling User, Mention, and Entity

To leverage the social network structure, we first train low-dimensional embeddings for the authors using the social relations. The mention and entity representations are given by word embeddings learnt with a large Twitter corpus and pre-trained Freebase entity embeddings respectively. We will denote the user, word, entity embedding matrices as:

$$\mathbf{E}^{(u)} = \{\mathbf{v}_u^{(u)}\} \quad \mathbf{E}^{(w)} = \{\mathbf{v}_w^{(w)}\} \quad \mathbf{E}^{(e)} = \{\mathbf{v}_e^{(e)}\},$$

where $\mathbf{E}^{(u)}, \mathbf{E}^{(w)}, \mathbf{E}^{(e)}$ are $V^{(u)} \times D^{(u)}, V^{(w)} \times D^{(w)}, V^{(e)} \times D^{(e)}$ matrices, and $\mathbf{v}_u^{(u)}$, $\mathbf{v}_w^{(w)}$, $\mathbf{v}_e^{(e)}$ are $D^{(u)}$, $D^{(w)}$, $D^{(e)}$ dimensional embedding vectors respectively. $V^{(u)}, V^{(w)}, V^{(e)}$ are the vocabulary sizes for users, words, and entities. Finally, we present a composition model for learning semantic interactions between user, mention, and entity.

User embeddings We obtain low-dimensional Twitter author embeddings $\mathbf{E}^{(u)}$ using *LINE* — the recently proposed model for embedding information networks (Tang et al., 2015). Specifically, we train *LINE* with the second-order proximity, which assumes that Twitter users sharing many neighbors are

close to each other in the embedding space. According to the original paper, the second-order proximity yields slightly better performances than the first-order proximity, which assumes connecting users are close to each other, on a variety of downstream tasks.

Mention embeddings The representation of a mention is the average of embeddings of words it contains. As each mention is typically one to three words, the simple representations often perform surprisingly well (Socher et al., 2013). We adopt the structured skip-gram model (Ling et al., 2015) to learn the word embeddings $\mathbf{E}^{(w)}$ on a Twitter corpus with 52 million tweets (Owoputi et al., 2013). The mention vector of the t-th mention candidate can be written as:

$$\mathbf{v}_t^{(m)} = \frac{1}{|\mathbf{x}_t^{(w)}|} \sum_{w \in \mathbf{x}_t^{(w)}} \mathbf{v}_w^{(w)}, \quad (4)$$

where $\mathbf{x}_t^{(w)}$ is the set of words in the mention.

Entity embeddings We use the pre-trained Freebase entity embeddings released by Google to represent entity candidates, which we refer as $\mathbf{E}^{(e)}$.[4] The embeddings are trained with the skip-gram model (Mikolov et al., 2013) on 100 billion words from various news articles. The entity embeddings can also be learnt from Wikipedia hyperlinks or Freebase entity relations, which we leave as future work.

Compositions of user, mention, and entity The distributed representations of users, mentions, and entities offer additional information that is useful for improving entity disambiguation capability. In particular, we explore the information by making two assumptions: socially connected users are interested in similar entities (entity homophily), and semantically related mentions are likely to be linked to similar entities.

We utilize a simple composition model that takes the form of the summation of two bilinear scoring functions, each of which explicitly leverages one of the assumptions. Given the author representation $\mathbf{v}_u^{(u)}$, the mention representation $\mathbf{v}_t^{(m)}$, and the entity representation $\mathbf{v}_{y_t}^{(e)}$, the output of the model can be written as:

$$\begin{aligned} g_2(\mathbf{x}, y_t, u, t; \Theta_2) =& \mathbf{v}_u^{(u)\top} \mathbf{W}^{(u,e)} \mathbf{v}_{y_t}^{(e)} \\ &+ \mathbf{v}_t^{(m)\top} \mathbf{W}^{(m,e)} \mathbf{v}_{y_t}^{(e)}, \end{aligned} \quad (5)$$

where $\mathbf{W}^{(u,e)}$ and $\mathbf{W}^{(m,e)}$ are $D^{(u)} \times D^{(e)}$ and $D^{(w)} \times D^{(e)}$ bilinear transformation matrices. Similar bilinear formulation has been used in the literature of knowledge base completion and inference (Socher et al., 2013; Yang et al., 2014). The parameters of the composition model are $\Theta_2 = \{\mathbf{W}^{(u,e)}, \mathbf{W}^{(m,e)}, \mathbf{E}^{(u)}, \mathbf{E}^{(w)}, \mathbf{E}^{(e)}\}$.

4.3 Non-overlapping Inference

The non-overlapping constraint for entity assignments requires inference method that is different from the standard Viterbi algorithm for a linear chain. We now present a variant of the Viterbi algorithm for the non-overlapping structure. Given the overall scoring function $g(\mathbf{x}, y_t, u, t)$ for the t-th mention candidate choosing an entity y_t, we sort the mention candidates by their end indices and define the Viterbi recursion by

$$\hat{y}_t = \underset{y_t \in \mathcal{Y}_{\mathbf{x}_t}, y_t \neq \textbf{Nil}}{\arg\max}\ g(\mathbf{x}, y_t, u, t) \quad (6)$$

$$a(1) = \max(g(\mathbf{x}, \textbf{Nil}, u, 1), g(\mathbf{x}, \hat{y}_1, u, 1)) \quad (7)$$

$$a(t) = \max\left(\psi_t(\textbf{Nil}), \psi_t(\hat{y}_t)\right) \quad (8)$$

$$\psi_t(\textbf{Nil}) = g(\mathbf{x}, \textbf{Nil}, u, t) + a(t-1) \quad (9)$$

$$\begin{aligned} \psi_t(\hat{y}_t) =& g(\mathbf{x}, \hat{y}_t, u, t) + \sum_{prev(t) < t' < t} g(\mathbf{x}, \textbf{Nil}, u, t') \\ &+ a(prev(t)) \end{aligned} \quad (10)$$

where $\mathcal{Y}_{\mathbf{x}_t}$ is set of entity candidates for the t-th mention candidate, and $prev(t)$ is a function that points out the previous non-overlapping mention candidate for the t-th mention candidate. We exclude any second-order features between entities. Therefore, for each mention candidate, we only need to decide whether it can take the highest scored entity candidate $\hat{y}_t$ or the special **Nil** entity based on whether it is overlapped with other mention candidates.

[4] Available at https://code.google.com/archive/p/word2vec/

4.4 Loss-augmented Training

The parameters need to be learnt during training are $\Theta = [\Theta_1, \{\mathbf{W}^{(u,e)}, \mathbf{W}^{(m,e)}\}]$.[5] We train NTEL by minimizing the following loss function for each training tweet:

$$L(\Theta) = \max_{\mathbf{y} \in \mathcal{Y}_\mathbf{x}} (\Delta(\mathbf{y}, \mathbf{y}^*) + s(\mathbf{x}, \mathbf{y}, u)) - s(\mathbf{x}, \mathbf{y}^*, u), \tag{11}$$

where $\mathbf{y}^*$ is the gold structure, $\mathcal{Y}_\mathbf{x}$ represents the set of valid output structures for $\mathbf{x}$, and $\Delta(\mathbf{y}, \mathbf{y}^*)$ is the weighted hamming distance between the gold structure $\mathbf{y}^*$ and the valid structure $\mathbf{y}$. The hamming loss is decomposable on the mention candidates, which enables efficient inferences. We set the hamming loss weight to 0.2 after a preliminary search. Note that the number of parameters in our composition model is large. Thus, we include an L2 regularizer on these parameters, which is omitted from Equation 11 for brevity. The evaluation of the loss function corresponds to the loss-augmented inference problem:

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y} \in \mathcal{Y}_\mathbf{x}} (\Delta(\mathbf{y}, \mathbf{y}^*) + s(\mathbf{x}, \mathbf{y}, u)), \tag{12}$$

which can be solved by the above non-overlapping inference algorithm. We employ vanilla SGD algorithm to optimize all the parameters. The numbers of training epochs are determined by early stopping (at most 1000 epochs). Training takes 6-8 hours on 4 threads.

5 Experiments

In this section, we evaluate NTEL on the NEEL and TACL datasets as described in § 2, focusing on investigating whether social information can improve the task. We also compare NTEL with the previous state-of-the-art system.

5.1 Social Network Expansion

We utilize Twitter follower, mention, and retweet social networks to train user embeddings. We were able to identify 2,312 authors for the tweets of the two datasets in March 2016. We then used the Twitter API to crawl their friend links and timelines, from which we can induce the networks. We find the numbers of social connections (bidirectional links) between these users are relatively small. In order to learn better user embeddings, we expand the set of author nodes by including nodes that will do the most to densify the author networks. For the follower network, we add additional individuals who are followed by at least twenty authors in the original set. For the mention or retweet networks, we add all users who have mentioned or retweeted by at least ten authors in the original set. The statistics of the resulting networks are presented in Table 3.

Network	# Author	# Relation
FOLLOWER+	8,772	286,800
MENTION+	6,119	57,045
RETWEET+	7,404	59,313

Table 3: Statistics of author social networks used for training user embeddings.

5.2 Experimental Settings

Following Yang and Chang (2015), we train all the models with the NEEL-train dataset and evaluate different systems on the NEEL-test and TACL datasets. In addition, 800 tweets from the NEEL-train dataset are sampled as our development set to perform parameter tuning. Note that Yang and Chang (2015) also attempt to optimize F1 scores by balancing precision and recall scores on the development set; we do not fine tune our F1 in this way, so that we can apply a single trained system across different test sets.

Metrics We follow prior work (Guo et al., 2013a; Yang and Chang, 2015) and perform the standard evaluation for an end-to-end entity linking system, computing precision, recall, and F1 score according to the entity references and the system outputs. An output entity is considered as correct if it matches the gold entity and the mention boundary overlaps with the gold mention boundary. More details about the metrics are described by Carmel et al. (2014).

Competitive systems Our first baseline system, NTEL-nonstruct, ignores the structure information and makes the entity assignment decision for each mention candidate individually. For NTEL, we start with a baseline system using the surface features, and then incorporate the two bilinear functions

[5] We fixed the pre-trained embedding matrices during loss-augmented training.

(user-entity and mention-entity) described in Equation 5 incrementally. Our main evaluation uses the RETWEET+ network, since the retweet network had the greatest entity homophily; an additional evaluation compares across network types.

Parameter tuning We tune all the hyper-parameters on the development set, and then re-train the models on the full training data with the best parameters. We choose the number of hidden units for the MLP from $\{20, 30, 40, 50\}$, and the regularization penalty for our composition model from $\{0.001, 0.005, 0.01, 0.05, 0.1\}$. The sizes of user embeddings and word embeddings are selected from $\{50, 100\}$ and $\{200, 400, 600\}$ respectively. The pre-trained Freebase entity embedding size is 1000. The learning rate for the SGD algorithm is set as 0.01. During training, we check the performance on the development set regularly to perform early stopping.

5.3 Results

Table 4 summarizes the empirical findings for our approach and S-MART (Yang and Chang, 2015) on the tweet entity linking task. For the systems with user-entity bilinear function, we report results obtained from embeddings trained on RETWEET+ in Table 4, and other results are available in Table 5. The best hyper-parameters are: the number of hidden units for the MLP is 40, the L2 regularization penalty for the composition parameters is 0.005, and the user embedding size is 100. For the word embedding size, we find 600 offers marginal improvements over 400 but requires longer training time. Thus, we choose 400 as the size of word embeddings.

As presented in Table 4, NTEL-nonstruct performs 2.7% F1 worse than the NTEL baseline on the two test sets, which indicates the non-overlapping inference improves system performance on the task. With structured inference but without embeddings, NTEL performs roughly the same as S-MART, showing that a feedforward neural network offers similar expressivity to the regression trees employed by Yang and Chang (2015).

Performance improves substantially with the incorporation of low-dimensional author, mention, and entity representations. As shown in Table 4, by learning the interactions between mention and entity representations, NTEL with mention-entity bilinear function outperforms the NTEL baseline system by 1.8% F1 on average. Specifically, the bilinear function results in considerable performance gains in recalls, with small compromise in precisions on the datasets.

Social information helps to increase about 1% F1 on top of both the NTEL baseline system and the NTEL system with mention-entity bilinear composition. In contrast to the mention-entity composition model, which mainly focuses on improving the baseline system on recall scores, the user-entity composition model increases around 2.5% recalls, without much sacrifice in precisions.

Our best system achieves the state-of-the-art results on the NEEL-test dataset and the TACL dataset, outperforming S-MART by 0.9% and 5.4% F1 scores respectively. To establish the statistical significance of the results, we obtain 100 bootstrap samples for each test set, and compute the F1 score on each sample for each algorithm. Two-tail paired t-test is then applied to determine if the F1 scores of two algorithms are significantly different. NTEL significantly outperforms S-MART on the NEEL-test dataset and the TACL dataset under $p < 0.01$ level, with t-statistics equal to 11.5 and 33.6 respectively.

As shown in Table 5, MENTION+ and RETWEET+ perform slightly better than FOLLOWER+. Puniyani et al. (2010) show that the mention network has stronger linguistic properties than the follower network, as it gives better correlations on each author's distribution over latent topics as induced by latent Dirichlet allocation (Blei et al., 2003). Our results suggest that the properties hold with respect to the authors' interests on real world entities.

5.4 Error Analysis & Discussion

We examine the outputs of different systems, focusing on investigating what errors are corrected by the two bilinear functions. The results reveal that the mention-entity composition improves the system ability to tackle mentions that are abbreviations such as 'WSJ' (*The Wall Street Journal*) and 'SJSU' (*San Jose State University*), which leads to higher recall scores. The mention-entity model also helps to eliminate errors that incorrectly link non-entities to popular entities. For example, the NTEL baseline

System	user -entity	mention -entity	NEEL-test			TACL			Avg. F1
			P	R	F1	P	R	F1	
Our approach									
NTEL-nonstruct			80.0	68.0	73.5	64.7	62.3	63.5	68.5
NTEL			**82.8**	69.3	75.4	68.0	66.0	67.0	71.2
NTEL	✓		82.3	71.8	76.7	66.9	68.7	67.8	72.2
NTEL		✓	80.2	**75.8**	77.9	66.9	**69.3**	68.1	73.0
NTEL	✓	✓	81.9	75.6	**78.6**	**69.0**	69.0	**69.0**	**73.8**
Best published results									
S-MART			80.2	75.4	77.7	60.1	67.7	63.6	70.7

Table 4: Evaluation results on the NEEL-test and TACL datasets for different systems. The best results are in **bold**.

Network	NEEL-test			TACL		
	P	R	F1	P	R	F1
FOLLOWER+	82.2	75.1	78.5	67.8	68.7	68.2
MENTION+	**82.5**	**76.0**	**79.1**	67.5	**69.3**	68.4
RETWEET+	81.9	75.6	78.6	**69.0**	69.0	**69.0**

Table 5: Comparison of different social networks with our full model. The best results are in **bold**.

system links 'sec' in the tweet 'I'm a be in Miami for sec to hit da radio!' to *Southeastern Conference*, which is corrected by the mention-entity composition model. The word semantic information encoded in the mention representations alleviates the biased entity information given by the surface features.

The user-entity composition model is good at handling highly ambiguous mentions. For example, our full model successfully disambiguates entities for mentions such as 'Sox' (*Boston Red Sox* vs. *Chicago White Sox*), 'Sanders' (*Bernie Sanders* vs. *Barry Sanders*), and 'Memphis' (*Memphis Grizzlies* vs. *Memphis, Tennessee*), which are mistakenly linked to the other entities or **Nil** by the mention-entity model. Another example is that the social network information helps the system correctly link 'Kim' to *Lil' Kim* instead of *Kim Kardashian*, despite that the latter entity's wikipedia page is considerably more popular.

6 Related Work

Tweet entity linking Previous work on entity linking mainly focuses on well-written documents (Bunescu and Pasca, 2006; Cucerzan, 2007; Milne and Witten, 2008), where entity disambiguation is usually performed by maximizing the global topical coherence between entities. However, these approaches often yield unsatisfactory performance on Twitter messages, due to the short and noisy nature of the tweets. To tackle this problem, collective tweet entity linking methods that leverage enriched context and metadata information have been proposed (Huang et al., 2014). Guo et al. (2013b) search for textually similar tweets for a target tweet, and encourage these Twitter messages to contain similar entities through label propagation. Shen et al. (2013) employ Twitter user account information to improve entity linking, based on the intuition that all tweets posted by the same user share an underlying topic distribution. Fang and Chang (2014) demonstrate that spatial and temporal signals are critical for the task, and they advance the performance by associating entity prior distributions with different timestamps and locations. Our work overcomes the difficulty by leveraging social relations — socially connected individuals are assumed to share similar interests on entities. As the Twitter post information is often sparse for some users, our assumption enables the utilization of more relevant information that helps to improve the task.

NLP with social relations Most previous work on incorporating social relations for NLP problems focuses on Twitter sentiment analysis, where the existence of social relations between users is considered as a clue that the sentiment polarities of messages from the users should be similar. Speriosu et al. (2011) construct a heterogeneous network with tweets, users, and n-grams as nodes, and the sentiment label distributions associated with the nodes

are refined by performing label propagation over social relations. Tan et al. (2011) and Hu et al. (2013) leverage social relations for sentiment analysis by exploiting a factor graph model and the graph Laplacian technique respectively, so that the tweets belonging to social connected users share similar label distributions. We work on entity linking in Twitter messages, where the label space is much larger than that of sentiment classification. The social relations can be more relevant in our problem, as it is challenging to obtain the entity prior distribution for each individual.

7 Conclusion

We present a neural based structured learning architecture for tweet entity linking, leveraging the tendency of socially linked individuals to share similar interests on named entities — the phenomenon of *entity homophily*. By modeling the compositions of vector representations of author, entity, and mention, our approach is able to exploit the social network as a source of contextual information. This vector-compositional model is combined with nonlinear feature combinations of surface features, via a feedforward neural network. To avoid predicting overlapping entity mentions, we employ a structured prediction algorithm, and train the system with loss-augmented decoding.

Social networks arise in other settings besides microblogs, such as webpages and academic research articles; exploiting these networks is a possible direction for future work. We would also like to investigate other metadata attributes that are relevant to the task, such as spatial and temporal signals.

Acknowledgments This research was supported by the National Science Foundation under awards IIS-1111142 and RI-1452443, by the National Institutes of Health under award number R01-GM112697-01, and by the Air Force Office of Scientific Research.

References

Sitaram Asur and Bernardo A Huberman. 2010. Predicting the future with social media. In *Web Intelligence and Intelligent Agent Technology (WI-IAT)*, pages 492–499.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.

R. C Bunescu and M. Pasca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL)*.

Amparo E Cano, Giuseppe Rizzo, Andrea Varga, Matthew Rowe, Milan Stankovic, and Aba-Sah Dadzie. 2014. Making sense of microposts (# microposts2014) named entity extraction & linking challenge. *Making Sense of Microposts (# Microposts2014)*.

David Carmel, Ming-Wei Chang, Evgeniy Gabrilovich, Bo-June Paul Hsu, and Kuansan Wang. 2014. Erd'14: entity recognition and disambiguation challenge. In *ACM SIGIR Forum*, pages 63–77.

Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on wikipedia data. In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*.

Yuan Fang and Ming-Wei Chang. 2014. Entity linking on microblogs with spatial and temporal signals. *Transactions of the Association for Computational Linguistics (ACL)*.

Stephen Guo, Ming-Wei Chang, and Emre Kiciman. 2013a. To link or not to link? a study on end-to-end tweet entity linking. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, Atlanta, GA.

Yuhang Guo, Bing Qin, Ting Liu, and Sheng Li. 2013b. Microblog entity linking by leveraging extra posts. In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*, Seattle, WA.

Dirk Hovy. 2015. Demographic factors improve classification performance. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 752–762, Beijing, China.

Xia Hu, Lei Tang, Jiliang Tang, and Huan Liu. 2013. Exploiting social relations for sentiment analysis in microblogging. In *Proceedings of the sixth ACM international conference on Web search and data mining (WSDM)*, pages 537–546.

Hongzhao Huang, Yunbo Cao, Xiaojiang Huang, Heng Ji, and Chin-Yew Lin. 2014. Collective tweet wikification based on semi-supervised graph regularization. In *Proceedings of the Association for Computational Linguistics (ACL)*, Baltimore, MD.

Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. 2010. What is Twitter, a social network or a news media? In *Proceedings of the Conference on World-Wide Web (WWW)*, pages 591–600, New York. ACM.

Jiwei Li, Alan Ritter, and Dan Jurafsky. 2015. Learning multi-faceted representations of individuals from

heterogeneous evidence using neural networks. *arXiv preprint arXiv:1510.05198*.

Wang Ling, Chris Dyer, Alan Black, and Isabel Trancoso. 2015. Two/too simple adaptations of word2vec for syntax problems. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, Denver, CO.

Michael Mathioudakis and Nick Koudas. 2010. Twittermonitor: trend detection over the twitter stream. In *Proceedings of the ACM SIGMOD International Conference on Management of data (SIGMOD)*, pages 1155–1158.

Miller McPherson, Lynn Smith-Lovin, and James M Cook. 2001. Birds of a feather: Homophily in social networks. *Annual review of sociology*, pages 415–444.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Neural Information Processing Systems (NIPS)*, pages 3111–3119, Lake Tahoe.

D. Milne and I. H. Witten. 2008. Learning to link with Wikipedia. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM)*.

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A. Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 380–390, Atlanta, GA.

Kriti Puniyani, Jacob Eisenstein, Shay Cohen, and Eric P. Xing. 2010. Social links from latent topics in microblogs. In *Proceedings of NAACL Workshop on Social Media*, Los Angeles.

Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. 2011. Named entity recognition in tweets: an experimental study. In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*.

Wei Shen, Jianyong Wang, Ping Luo, and Min Wang. 2013. Linking named entities in tweets with knowledge base via user interest modeling. In *Proceedings of Knowledge Discovery and Data Mining (KDD)*.

Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Y. Ng. 2013. Reasoning With Neural Tensor Networks For Knowledge Base Completion. In *Neural Information Processing Systems (NIPS)*, Lake Tahoe.

Michael Speriosu, Nikita Sudan, Sid Upadhyay, and Jason Baldridge. 2011. Twitter polarity classification with label propagation over lexical links and the follower graph. In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*, pages 53–63.

Chenhao Tan, Lillian Lee, Jie Tang, Long Jiang, Ming Zhou, and Ping Li. 2011. User-level sentiment analysis incorporating social networks. In *Proceedings of Knowledge Discovery and Data Mining (KDD)*.

Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *Proceedings of the Conference on World-Wide Web (WWW)*.

Shaomei Wu, Jake M Hofman, Winter A Mason, and Duncan J Watts. 2011. Who says what to whom on twitter. In *Proceedings of the Conference on World-Wide Web (WWW)*, pages 705–714.

Yi Yang and Ming-Wei Chang. 2015. S-mart: Novel tree-based structured learning algorithms applied to tweet entity linking. In *Proceedings of the Association for Computational Linguistics (ACL)*, Beijing, China.

Yi Yang and Jacob Eisenstein. 2015. Putting things in context: Community-specific embedding projections for sentiment analysis. *arXiv preprint arXiv:1511.06052*.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*.

Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. 2015. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *Proceedings of the Association for Computational Linguistics (ACL)*, Beijing, China.

A Appendix: Additional Results

System	user -entity	mention -entity	NEEL-test			TACL			Avg. F1
			P	R	F1	P	R	F1	
Our approach									
NTEL-nonstruct			83.0	71.8	77.0	80.9	69.0	74.5	75.8
NTEL			84.4	73.9	78.8	82.0	71.3	76.3	77.6
NTEL	✓		83.8	76.7	80.1	81.8	**73.3**	77.3	78.7
NTEL		✓	84.1	78.3	81.1	83.0	71.7	76.9	79.0
NTEL	✓	✓	**84.8**	**79.3**	**82.0**	**83.5**	72.7	**77.7**	**79.9**
Best published results									
S-MART			83.2	79.2	81.1	76.8	73.0	74.9	78.0

Table 6: Evaluation results on the NEEL-test and TACL datasets for different systems. Twitter messages that contain no ground truth entities are excluded for both training and testing. The best results are in **bold**.

In the first version of (Yang and Chang, 2015), the Twitter messages that contain no ground truth entities are excluded in the experiments. For completeness, we now present the evaluation results of NTEL in this setting, which are shown in Table 6. The RETWEET+ network is adopted to train author embeddings. The best hyper-parameters are the same as those described in § 5, except for the L2 regularization penalty for the composition parameters, which is set as 0.01 here.

The results are generally better than those presented in Table 4. As shown, NTEL benefits from the distributed representations of authors, mentions, and entities, which improve the average F1 score by 2.3 points. NTEL also gives the best results on the datasets, outperforming S-MART by about 2% F1 on average.