Deep Joint Entity Disambiguation with Local Neural Attention

Octavian-Eugen Ganea and **Thomas Hofmann**
Department of Computer Science
ETH Zurich
{octavian.ganea,thomas.hofmann}@inf.ethz.ch

Abstract

We propose a novel deep learning model for joint document-level entity disambiguation, which leverages learned neural representations. Key components are entity embeddings, a neural attention mechanism over local context windows, and a differentiable joint inference stage for disambiguation. Our approach thereby combines benefits of deep learning with more traditional approaches such as graphical models and probabilistic mention-entity maps. Extensive experiments show that we are able to obtain competitive or state-of-the-art accuracy at moderate computational costs.

1 Introduction

Entity disambiguation (ED) is an important stage in text understanding which automatically resolves references to entities in a given knowledge base (KB). This task is challenging due to the inherent ambiguity between surface form mentions such as names and the entities they refer to. This many-to-many ambiguity can often be captured partially by name-entity co-occurrence counts extracted from entity-linked corpora.

ED research has largely focused on two types of contextual information for disambiguation: *local* information based on words that occur in a context window around an entity mention, and, *global* information, exploiting document-level coherence of the referenced entities. Many state-of-the-art methods aim to combine the benefits of both, which is also the philosophy we follow in this paper. What is specific to our approach is that we use embeddings of entities as a common representation to assess local as well as global evidence.

In recent years, many text and language understanding tasks have been advanced by neural network architectures. However, despite recent work, competitive ED systems still largely employ manually designed features. Such features often rely on domain knowledge and may fail to capture all relevant statistical dependencies and interactions. The explicit goal of our work is to use deep learning in order to learn basic features and their combinations from scratch. To the best of our knowledge, our approach is the first to carry out this program with full rigor.

2 Contributions and Related Work

There is a vast prior research on entity disambiguation, highlighted by (Ji, 2016). We will focus here on a discussion of our main contributions in relation to prior work.

Entity Embeddings. We have developed a simple, yet effective method to embed entities and words in a common vector space. This follows the popular line of work on word embeddings, e.g. (Mikolov et al., 2013; Pennington et al., 2014), which was recently extended to entities and ED by (Yamada et al., 2016; Fang et al., 2016; Zwicklbauer et al., 2016; Huang et al., 2015). In contrast to the above methods that require data about entity-entity co-occurrences which often suffers from sparsity, we rather bootstrap entity embeddings from their canonical entity pages and local context of their hyperlink annotations. This allows for more efficient training and alleviates the need to compile co-linking statistics. These vector representations are a key component to avoid hand-engineered features, multiple disambiguation steps, or the need for additional *ad hoc* heuristics when solving the ED task.

Context Attention. We present a novel attention mechanism for local ED. Inspired by mem-

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 2602–2612
Copenhagen, Denmark, September 7–11, 2017.

ory networks of (Sukhbaatar et al., 2015) and insights of (Lazic et al., 2015), our model deploys attention to select words that are informative for the disambiguation decision. A learned combination of the resulting context-based entity scores and a mention–entity prior yields the final local scores. Our local model achieves better accuracy than the local probabilistic model of (Ganea et al., 2016), as well as the feature-engineered local model of (Globerson et al., 2016). As an added benefit, our model has a smaller memory footprint and it's very fast for both training and testing.

There have been other deep learning approaches to define local context models for ED. For instance (Francis-Landau et al., 2016; He et al., 2013) use convolutional neural networks (CNNs) and stacked denoising auto-encoders, respectively, to learn representations of textual documents and canonical entity pages. Entities for each mention are locally scored based on cosine similarity with the respective document embedding. In a similar local setting, (Sun et al., 2015) embed mentions, their immediate contexts and their candidate entities using word embeddings and CNNs. However, their entity representations are restrictively built from entity titles and entity categories only. Unfortunately, the above models are rather 'black-box' (as opposed to ours which reveals the attention focus) and were never extended to perform joint document disambiguation.

Collective Disambiguation. Last, a novel deep learning architecture for global ED is proposed. Mentions in a document are resolved jointly, using a conditional random field (Lafferty et al., 2001) with parametrized potentials. We suggest to learn the latter by casting loopy belief propagation (LBP) (Murphy et al., 1999) as a rolled-out deep network. This is inspired by similar approaches in computer vision, e.g. (Domke, 2013), and allows us to backpropagate through the (truncated) message passing, thereby optimizing the CRF potentials to work well in conjunction with the inference scheme. Our model is thus trained end-to-end with the exception of the pre-trained word and entity embeddings. Previous work has investigated different approximation techniques, including: random graph walks (Guo and Barbosa, 2016), personalized PageRank (Pershina et al., 2015), inter-mention voting (Ferragina and Scaiella, 2010), graph pruning (Hoffart et al., 2011), integer linear programming (Cheng and Roth, 2013), or ranking SVMs (Ratinov et al., 2011). Mostly connected to our approach is (Ganea et al., 2016) where LBP is used for inference (but not learning) in a probabilistic graphical model and (Globerson et al., 2016) where a single round of message passing with attention is performed. To our knowledge, we are one of the first to investigate differentiable message passing for NLP problems.

3 Learning Entity Embeddings

In a first step, we propose to train entity vectors that can be used for the ED task (and potentially for other tasks). These embeddings compress the semantic meaning of entities and drastically reduce the need for manually designed features or co-occurrence statistics.

Entity embeddings are bootstrapped from word embeddings and are trained independently for each entity. A few arguments motivate this decision: (i) there is no need for entity co-occurrence statistics that suffer from sparsity issues and/or large memory footprints; (ii) vectors of entities in a subset domain of interest can be trained separately, obtaining potentially significant speed-ups and memory savings that would otherwise be prohibitive for large entity KBs;[1] (iii) entities can be easily added in an incremental manner, which is important in practice; (iv) the approach extends well into the tail of rare entities with few linked occurrences; (v) empirically, we achieve better quality compared to methods that use entity co-occurrence statistics.

Our model embeds words and entities in the same low-dimensional vector space in order to exploit geometric similarity between them. We start with a pre-trained word embedding map $\mathbf{x} : \mathcal{W} \rightarrow \mathbb{R}^d$ that is known to encode semantic meaning of words $w \in \mathcal{W}$; specifically we use word2vec pre-trained vectors (Mikolov et al., 2013). We extend this map to entities $\mathcal{E}$, i.e. $\mathbf{x} : \mathcal{E} \rightarrow \mathbb{R}^d$, as described below.

We assume a generative model in which words that co-occur with an entity e are sampled from a conditional distribution $p(w|e)$ when they are generated. Empirically, we collect word-entity co-occurrence counts $\#(w, e)$ from two sources: (i) the canonical KB description page of the entity (e.g. entity's Wikipedia page in our case), and (ii) the windows of fixed size surrounding mentions of the entity in an annotated corpus (e.g. Wikipedia

[1] Notably useful with (limited memory) GPU hardware.

hyperlinks in our case). These counts define a practical approximation of the above word-entity conditional distribution, i.e. $\hat{p}(w|e) \propto \#(w,e)$. We call this the "positive" distribution of words related to the entity. Next, let $q(w)$ be a generic word probability distribution which we use for sampling "negative" words unrelated to a specific entity. As in (Mikolov et al., 2013), we choose a smoothed unigram distribution $q(w) = \hat{p}(w)^\alpha$ for some $\alpha \in (0,1)$. The desired outcome is that vectors of positive words are closer (in terms of dot product) to the embedding of entity e compared to vectors of random words. Let $w^+ \sim \hat{p}(w|e)$ and $w^- \sim q(w)$. Then, we use a max-margin objective to infer the optimal embedding for entity e:

$$\begin{aligned} J(\mathbf{z}; e) &:= \mathbb{E}_{w^+|e}\, \mathbb{E}_{w^-} \left[h\left(\mathbf{z}; w^+, w^-\right)\right] \\ h(\mathbf{z}; w, v) &:= \left[\gamma - \langle \mathbf{z}, \mathbf{x}_w - \mathbf{x}_v \rangle\right]_+ \\ \mathbf{x}_e &:= \underset{\mathbf{z}:\|\mathbf{z}\|=1}{\arg\min}\, J(\mathbf{z}; e) \end{aligned} \tag{1}$$

where $\gamma > 0$ is a margin parameter and $[\cdot]_+$ is the ReLU function. The above loss is optimized using stochastic gradient descent with projection over sampled pairs (w^+, w^-). Note that the entity vector is directly optimized on the unit sphere which is important in order to obtain qualitative embeddings.

We empirically assess the quality of our entity embeddings on entity similarity and ED tasks as detailed in Section 7 and Appendix A. The technique described in this section can also be applied, in principle, for computing embeddings of general text documents, but a comparison with such methods is left as future work.

4 Local Model with Neural Attention

We now explain our local ED approach that uses word and entity embeddings to steer a neural attention mechanism. We build on the insight that only a few context words are informative for resolving an ambiguous mention, something that has been exploited before in (Lazic et al., 2015). Focusing only on those words helps reducing noise and improves disambiguation. (Yamada et al., 2016) observe the same problem and adopt the restrictive strategy of removing all non-nouns. Here, we assume that a context word may be relevant, if it is strongly related to at least one of the entity candidates of a given mention.

Context Scores.

Let us assume that we have computed a mention–entity prior $\hat{p}(e|m)$ (procedure detailed in Section 6). In addition, for each mention m, a pruned candidate set $\Gamma(m)$ of at most S entities has been identified. Our model, depicted in Figure 1, computes a score for each $e \in \Gamma(m)$ based on the K-word local context $c = \{w_1, \ldots, w_K\}$ surrounding m, as well as on the prior. It is a composition of differentiable functions, thus it is smooth from input to output, allowing us to easily compute gradients and backpropagate through it.

Each word $w \in c$ and entity $e \in \Gamma(m)$ is mapped to its embedding via the pre-trained map $\mathbf{x}$ (cf. Section 3). We then compute an unnormalized support score for each word in the context as follows:

$$u(w) = \max_{e \in \Gamma(m)} \mathbf{x}_e^\top \mathbf{A} \mathbf{x}_w \tag{2}$$

where $\mathbf{A}$ is a parameterized diagonal matrix. The weight is high if the word is strongly related to at least one candidate entity. We often observe that uninformative words (e.g. similar to stop words) receive non-negligible scores which add undesired noise to our local context model. As a consequence, we (hard) prune to the top $R \leq K$ words with the highest scores[2] and apply a softmax function on these weights. Define the reduced context:

$$\bar{c} = \{w \in c | u(w) \in \text{topR}(\mathbf{u})\} \tag{3}$$

Then, the final attention weights are explicitly

$$\beta(w) = \begin{cases} \frac{\exp[u(w)]}{\sum_{v \in \bar{c}} \exp[u(v)]}\,. & \text{if } w \in \bar{c} \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Finally, we define a β-weighted context-based entity-mention score via

$$\Psi(e, c) = \sum_{w \in \bar{c}} \beta(w)\, \mathbf{x}_e^\top \mathbf{B}\, \mathbf{x}_w \tag{5}$$

where $\mathbf{B}$ is another trainable diagonal matrix. We will later use the same architecture for the *unary* scores of our global ED model.

Local Score Combination.

We integrate these context scores with the context-independent scores encoded in $\hat{p}(e|m)$.

[2] We implement this in a differentiable way by setting the lowest K-R attention weights in $\mathbf{u}$ to $-\infty$ and applying a vanila softmax on top of them. We used the layers Threshold and TemporalDynamicKMaxPooling from Torch nn package, which allow subgradient computation.

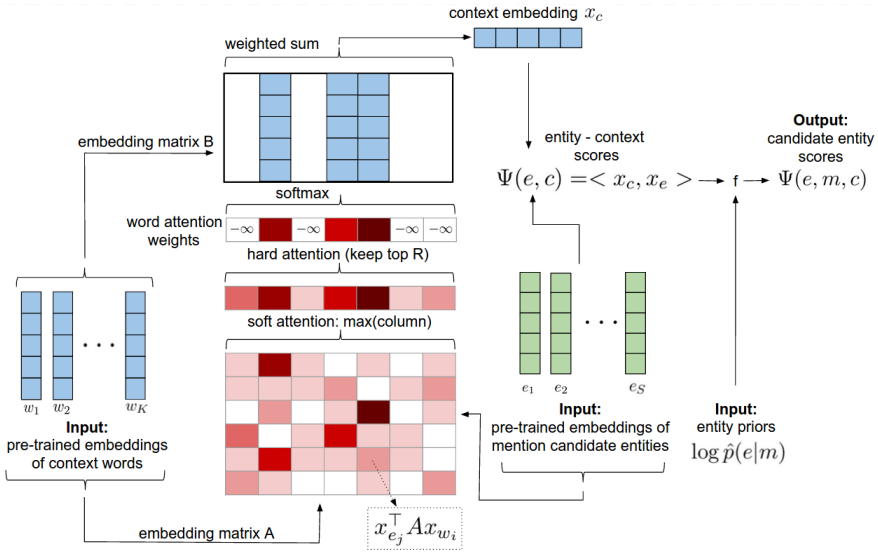

Figure 1: Local model with neural attention. Inputs: context word vectors, candidate entity priors and embeddings. Outputs: entity scores. All parts are differentiable and trainable with backpropagation.

Our final (unnormalized) local model is a combination of both $\Psi(e,c)$ and $\log \hat{p}(e|m)$:

$$\Psi(e,m,c) = f(\Psi(e,c), \log \hat{p}(e|m)) \quad (6)$$

We find a flexible choice for f to be important and superior to a naïve weighted average combination model. We therefore use a neural network with two fully connected layers of 100 hidden units and ReLU non-linearities, which we regularize as suggested in (Denton et al., 2015) by constraining the sum of squares of all weights in the linear layer. We use standard projected SGD for training. The same network is also used in Section 5.

Prediction is done independently for each mention m_i and context c_i by maximizing the $\Psi(e, m_i, c_i)$ score.

Learning the Local Model.

Entity and word embeddings are pre-trained as discussed in Section 3. Thus, the only learnable parameters are the diagonal matrices $\mathbf{A}$ and $\mathbf{B}$, plus the parameters of f. Having few parameters helps to avoid overfitting and to be able to train with little annotated data. We assume that a set of known mention-entity pairs $\{(m, e^*)\}$ with their respective context windows have been extracted from a corpus. For model fitting, we then utilize a max-margin loss that ranks ground truth entities higher than other candidate entities. This leads us to the objective:

$$\theta^* = \arg\min_\theta \sum_{D\in\mathcal{D}} \sum_{m\in D} \sum_{e\in\Gamma(m)} g(e,m), \quad (7)$$

$$g(e,m) := [\gamma - \Psi(e^*,m,c) + \Psi(e,m,c)]_+$$

where $\gamma > 0$ is a margin parameter and $\mathcal{D}$ is a training set of entity annotated documents. We aim to find a Ψ (i.e. parameterized by θ) such that the score of the correct entity e^* referenced by m is at least a margin γ higher than that of any other candidate entity e. Whenever this is not the case, the margin violation becomes the experienced loss.

5 Document-Level Deep Model

Next, we address global ED assuming document coherence among entities. We therefore introduce the notion of a document as consisting of a set of mentions $\mathbf{m} = m_1, \ldots, m_n$, along with their context windows $\mathbf{c} = c_1, \ldots c_n$. Our goal is to define a joint probability distribution over $\Gamma(m_1) \times \ldots \times \Gamma(m_n) \ni \mathbf{e}$. Each such $\mathbf{e}$ selects one candidate entity for each mention in the document. Obviously, the state space of $\mathbf{e}$ grows exponentially in the number of mentions n.

CRF Model.

Our model is a fully-connected pairwise conditional random field, defined on the log scale as

$$g(\mathbf{e},\mathbf{m},\mathbf{c}) = \sum_{i=1}^{n} \Psi_i(e_i) + \sum_{i<j} \Phi(e_i,e_j) \quad (8)$$

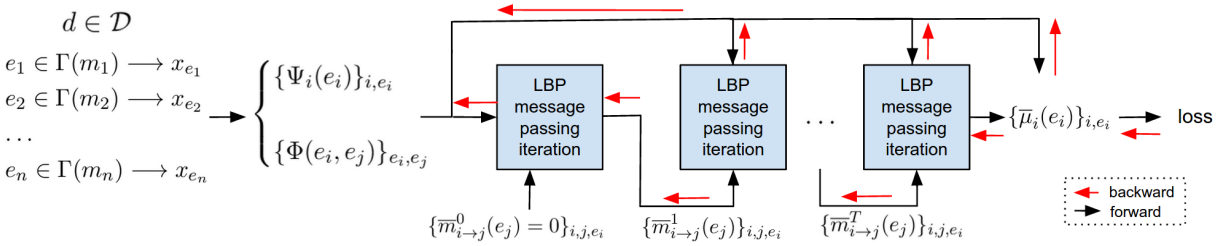

Figure 2: Global model: unrolled LBP deep network that is end-to-end differentiable and trainable.

The unary factors are the local scores $\Psi_i(e_i) = \Psi(e_i, c_i)$ described in Eq. (5). The pairwise factors are bilinear forms of the entity embeddings

$$\Phi(e, e') = \frac{2}{n-1}\, \mathbf{x}_e^\top \mathbf{C}\, \mathbf{x}_{e'}\,, \qquad (9)$$

where $\mathbf{C}$ is a diagonal matrix. Similar to (Ganea et al., 2016), the above normalization helps balancing the unary and pairwise terms across documents with different numbers of mentions.

The function value $g(\mathbf{e}, \mathbf{m}, \mathbf{c})$ is supposedly high for semantically related sets of entities that also have local support. The goal of a global ED prediction method is to perform maximum-a-posteriori on this CRF to find the set of entities $\mathbf{e}$ that maximize $g(\mathbf{e}, \mathbf{m}, \mathbf{c})$.

Differentiable Inference.

Training and prediction in binary CRF models as the one above is NP-hard. Therefore, in learning one usually maximizes a likelihood approximation and during operations (i.e. in prediction) one may use an approximate inference procedure, often based on message-passing. Among many challenges of these approaches, it is worth pointing out that weaknesses of the approximate inference procedure are generally not captured during learning. Inspired by (Domke, 2011, 2013), we use *truncated fitting* of loopy belief propagation (LBP) to a fixed number of message passing iterations. Our model directly optimizes the marginal likelihoods, using the same networks for learning and prediction. As noted by (Domke, 2013), this method is robust to model mis-specification, avoids inherent difficulties of partition functions and is faster compared to double-loop likelihood training (where, for each stochastic update, inference is run until convergence is achieved).

Our architecture is shown in Figure 2. A neural network with T layers encodes T message passing iterations of synchronous max-product LBP[3] which is designed to find the most likely (MAP) entity assignments that maximize $g(\mathbf{e}, \mathbf{m}, \mathbf{c})$. We also use message damping, which is known to speed-up and stabilize convergence of message passing. Formally, in iteration t, mention m_i votes for entity candidate $e \in \Gamma(m_j)$ of mention m_j using the normalized log-message $\overline{m}^t_{i\to j}(e)$ computed as:

$$m^{t+1}_{i\to j}(e) = \max_{e' \in \Gamma(m_i)} \{\Psi_i(e') + \Phi(e, e') + \sum_{k \neq j} \overline{m}^t_{k\to i}(e')\}\,. \qquad (10)$$

Herein the first part just reflects the CRF potentials, whereas the second part is defined as

$$\overline{m}^t_{i\to j}(e) = \log[\delta \cdot \mathrm{softmax}(m^t_{i\to j}(e)) + (1-\delta) \cdot \exp(\overline{m}^{t-1}_{i\to j}(e))] \qquad (11)$$

where $\delta \in (0, 1]$ is a damping factor. Note that, without loss of generality, we simplify the LBP procedure by dropping the factor nodes. The messages at first iteration (layer) are set to zero.

After T iterations (network layers), the beliefs (marginals) are computed as:

$$\mu_i(e) = \Psi_i(e) + \sum_{k \neq i} \overline{m}^T_{k\to i}(e) \qquad (12)$$

$$\overline{\mu}_i(e) = \frac{\exp[\mu_i(e)]}{\sum_{e' \in \Gamma(m_i)} \exp[\mu_i(e')]} \qquad (13)$$

Similar to the local case, we obtain accuracy improvement when combining the mention-entity prior $\hat{p}(e|m)$ with marginal $\mu_i(e)$ using the same non-linear combination function f from Equation 6 as follows:

$$\rho_i(e) := f(\overline{\mu}_i(e), \log \hat{p}(e|m_i)) \qquad (14)$$

The learned function f for global ED is non-trivial (see Figure 3), showing that the influence of the prior tends to weaken for larger $\mu(e)$,

[3] Sum-product and mean-field performed worse in our experiments.

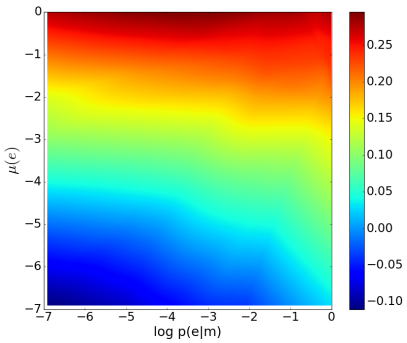

Figure 3: Non-linear scoring function of the belief and mention prior learned with a neural network. Achieves a 1.7% improvement on AIDA-B dataset compared to a weighted average scheme.

whereas it has a dominating influence whenever the document-level evidence is weak. We also experimented with the prior integrated directly inside the unary factors $\Psi_i(e_i)$, but results were worse because, in some cases, the global entity interaction is not able to recover from strong incorrect priors (e.g. country names have a strong prior towards the respective countries as opposed to national sports teams).

Parameters of our global model are the diagonal matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ and the weights of the f network. As before, we find a margin based objective to be the most effective and we suggest to fit parameters by minimizing a ranking loss[4] defined as:

$$L(\theta) = \sum_{D \in \mathcal{D}} \sum_{m_i \in D} \sum_{e \in \Gamma(m_i)} h(m_i, e) \quad (15)$$

$$h(m_i, e) = [\gamma - \rho_i(e_i^*) + \rho_i(e)]_+ \quad (16)$$

Computing this objective is trivial by running T times the steps described by Eqs. (10), (11), followed in the end by the step in Eq. (13). Each step is differentiable and the gradient of the model parameters can be computed on the resulting marginals and back-propagated over messages using chain rule.

At test time, marginals $\rho_i(e)$ are computed jointly per document using this network, but prediction is done independently for each mention m_i by maximizing its respective marginal score.

6 Candidate Selection

We use a mention-entity prior $\hat{p}(e|m)$ both as a feature and for entity candidate selection. It is computed by averaging probabilities from two indexes build from mention entity hyperlink count statistics from Wikipedia and a large Web corpus (Spitkovsky and Chang, 2012). Moreover, we add the YAGO dictionary of (Hoffart et al., 2011), where each candidate receives a uniform prior.

Candidate selection, i.e. construction of $\Gamma(e)$, is done for each input mention as follows: first, the top 30 candidates are selected based on the prior $\hat{p}(e|m)$. Then, in order to optimize for memory and run time (LBP has complexity quadratic in S), we keep only 7 of these entities based on the following heuristic: (i) the top 4 entities based on $\hat{p}(e|m)$ are selected, (ii) the top 3 entities based on the local context-entity similarity measured using the function from Eq. 5 are selected.[5]. We refrain from annotating mentions without any candidate entity, implying that precision and recall can be different in our case.

In a few cases, generic mentions of persons (e.g. "Peter") are coreferences of more specific mentions (e.g. "Peter Such") from the same document. We employ a simple heuristic to address this issue: for each mention m, if there exist mentions of persons that contain m as a continuous subse-

Method \ Metric	NDCG@1	NDCG@5	NDCG@10	MAP
WikiLinkMeasure (WLM)	0.54	0.52	0.55	0.48
(Yamada et al., 2016) d = 500	0.59	0.56	0.59	0.52
our (canonical pages) d = 300	0.624	0.589	0.615	0.549
our (canonical&hyperlinks) d = 300	**0.632**	**0.609**	**0.641**	**0.578**

Table 1: Entity relatedness results on the test set of (Ceccarelli et al., 2013). WLM is a well-known similarity method of (Milne and Witten, 2008).

Dataset	Number mentions	Number docs	Mentions per doc	Gold recall
AIDA-train	18448	946	19.5	-
AIDA-A (valid)	4791	216	22.1	96.9%
AIDA-B (test)	4485	231	19.4	98.2%
MSNBC	656	20	32.8	98.5%
AQUAINT	727	50	14.5	94.2%
ACE2004	257	36	7.1	90.6%
WNED-CWEB	11154	320	34.8	91.1%
WNED-WIKI	6821	320	21.3	92%

Table 2: Statistics of ED datasets. *Gold recall* is the percentage of mentions for which the entity candidate set contains the ground truth entity. We only train on mentions with at least one candidate.

[4]Optimizing a marginal log-likelihood loss function performed worse.

[5]We have used a simpler context vector here computed by simply averaging all its constituent word vectors.

Methods	AIDA-B
Local models	
prior $\hat{p}(e\|m)$	71.9
(Lazic et al., 2015)	86.4
(Globerson et al., 2016)	87.9
(Yamada et al., 2016)	87.2
our (local, K=100, R=50)	**88.8**
Global models	
(Huang et al., 2015)	86.6
(Ganea et al., 2016)	87.6
(Chisholm and Hachey, 2015)	88.7
(Guo and Barbosa, 2016)	89.0
(Globerson et al., 2016)	91.0
(Yamada et al., 2016)	91.5
our (global)	**92.22 ± 0.14**

Table 3: In-KB accuracy for AIDA-B test set. All baselines use KB+YAGO mention-entity index. For our method we show 95% confidence intervals obtained over 5 runs.

quence of words, then we consider the merged set of the candidate sets of these specific mentions as the candidate set for the mention m. We decide that a mention refers to a person if its most probable candidate by $\hat{p}(e|m)$ is a person.

7 Experiments

7.1 ED Datasets

We validate our ED models on some of the most popular available datasets used by our predecessors[6]. We provide statistics in Table 2.

- AIDA-CoNLL dataset (Hoffart et al., 2011) is one of the biggest manually annotated ED datasets. It contains training (AIDA-train), validation (AIDA-A) and test (AIDA-B) sets.
- MSNBC (MSB), AQUAINT (AQ) and ACE2004 (ACE) datasets cleaned and updated by (Guo and Barbosa, 2016)[7]
- WNED-WIKI (WW) and WNED-CWEB (CWEB): are larger, but automatically extracted, thus less reliable. Are built from the ClueWeb and Wikipedia corpora by (Guo and Barbosa, 2016; Gabrilovich et al., 2013).

7.2 Training Details and (Hyper)Parameters

We explain training details of our approach. All models are implemented in the Torch framework. **Entity Vectors Training & Relatedness Evaluation.** For entity embeddings only, we use

[6]TAC-KBP datasets used by (Yamada et al., 2016; Globerson et al., 2016; Sun et al., 2015) are no longer available.

[7]Available at: bit.ly/2gnSBLg

Global methods	MSB	AQ	ACE	CWEB	WW
prior $\hat{p}(e\|m)$	89.3	83.2	84.4	69.8	64.2
(Fang et al., 2016)	81.2	88.8	85.3	-	-
(Ganea et al., 2016)	91	89.2	**88.7**	-	-
(Milne and Witten, 2008)	78	85	81	64.1	81.7
(Hoffart et al., 2011)	79	56	80	58.6	63
(Ratinov et al., 2011)	75	83	82	56.2	67.2
(Cheng and Roth, 2013)	90	**90**	86	67.5	73.4
(Guo and Barbosa, 2016)	92	87	88	77	**84.5**
our (global)	**93.7** ± **0.1**	88.5 ± 0.4	**88.5** ± **0.3**	**77.9** ± **0.1**	77.5 ± 0.1

Table 4: Micro F1 results for other datasets.

Wikipedia (Feb 2014) corpus for training. Entity vectors are initialized randomly from a 0-mean normal distribution with standard deviation 1. We first train each entity vector on the entity's Wikipedia canonical description page (title words included) for 400 iterations. Subsequently, Wikipedia hyperlinks of the respective entities are used for learning until validation score (described below) stops improving. In each iteration, 20 positive words, each with 5 negative words, are sampled and used for optimization as explained in Section 3. We use Adagrad (Duchi et al., 2011) with a learning rate of 0.3. We choose embedding size $d = 300$, pre-trained (fixed) Word2Vec word vectors[8], $\alpha = 0.6$, $\gamma = 0.1$ and window size of 20 for the hyperlinks. We remove stop words before training. Since our method allows to train the embedding of each entity independently of other entities, we decide for efficiency reasons (and without loss of generality) to learn only the vectors of all entities appearing as mention candidates in all the test datasets described in Sec. 7.1, a total of 270000 entities. Training of those takes 20 hours on a single TitanX GPU with 12GB of memory.

We test and validate our entity embeddings on the entity relatedness dataset of (Ceccarelli et al., 2013). It contains 3319 and 3673 queries for the test and validation sets. Each query consist of one target entity and up to 100 candidate entities with gold standard binary labels indicating if the two entities are related. The associated task requires ranking of related candidate entities higher than the others. Following previous work, we use different evaluation metrics: normalized discounted cumulative gain (NDCG) and mean average precision (MAP). The validation score used during learning is then the sum of the four metrics showed in Table 1. We perform candidate ranking based on cosine similarity of entity pairs.

[8]By Word2Vec authors: http://bit.ly/1R9Wsqr

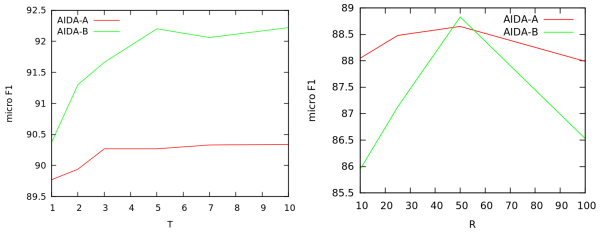

Table 5: Effects of two of the hyper-parameters. Left: A low T (e.g.5) is already sufficient for accurate approximate marginals. Right: Hard attention improves accuracy of a local model with K=100.

Freq gold entity	Number mentions	Solved correctly
0	5	80.0 %
1-10	0	-
11-20	4	100.0%
21-50	50	90.0%
> 50	4345	94.2%

$\hat{p}(e\|m)$ **gold entity**	Number mentions	Solved correctly
≤ 0.01	36	89.19%
0.01 - 0.03	249	88.76%
0.03 - 0.1	306	82.03%
0.1 - 0.3	381	86.61%
> 0.3	3431	96.53%

Table 6: ED accuracy on AIDA-B for our best system splitted by Wikipedia hyperlink frequency and mention prior of the ground truth entity, in cases where the gold entity appears in the candidate set.

Local and Global Model Training. Our local and global ED models are trained on AIDA-train (multiple epochs), validated on AIDA-A and tested on AIDA-B and other datasets mentioned in Section 7.1. We use Adam (Kingma and Ba, 2014) with learning rate of 1e-4 until validation accuracy exceeds 90%, afterwards setting it to 1e-5. Variable size mini-batches consisting of all mentions in a document are used during training. We remove stop words. Hyper-parameters of the best validated global model are: $\gamma = 0.01, K = 100, R = 25, S = 7, \delta = 0.5, T = 10$. For the local model, $R = 50$ was best. Validation accuracy is computed after each 5 epochs. To regularize, we use early stopping, i.e. we stop learning if the validation accuracy does not increase after 500 epochs. Training on a single GPU takes, on average, 2ms per mention, or 16 hours for 1250 epochs over AIDA-train.

By using diagonal matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$, we keep the number of parameters very low (approx. 1.2K parameters). This is necessary to avoid overfitting when learning from a very small training set. We also experimented with diagonal plus low-rank matrices, but encountered quality degradation.

7.3 Entity Similarity Results

Results for the entity similarity task are shown in Table 1. Our method outperforms the well established Wikipedia link measure and the method of (Yamada et al., 2016) using less information (only word - entity statistics). We note that the best result on this dataset was reported in the unpublished work of (Huang et al., 2015). Their entity embeddings are trained on many more sources of information (e.g. KG links, relations, entity types). However, our focus was to prove that lightweight trained embeddings useful for the ED task can also perform decently for the entity similarity task. We emphasize that our global ED model outperforms Huang's ED model (Table 3), likely due to the power of our local and joint neural network architectures. For example, our attention mechanism clearly benefits from explicitly embedding words and entities in the same space.

7.4 ED Baselines & Results

We compare with systems that report state-of-the-art results on the datasets. Some baseline scores from Table 4 are taken from (Guo and Barbosa, 2016). The best results for the AIDA datasets are reported by (Yamada et al., 2016) and (Globerson et al., 2016). We do not compare against (Pershina et al., 2015) since, as noted also by (Globerson et al., 2016), their mention index artificially includes the gold entity (guaranteed gold recall), which is not a realistic setting.

For a fair comparison with prior work, we use in-KB accuracy and micro F1 (averaged per mention) metrics to evaluate our approach. Results are shown in Tables 3 and 4. We run our system 5 times, each time we pick the best model on the validation set, and report results on the test set for these models. We obtain state of the art accuracy on AIDA which is the largest and hardest (by the accuracy of the $\hat{p}(e|m)$ baseline) manually created ED dataset . We are also competitive on the other datasets. It should be noted that all the other methods use, at least partially, engineered features. The merit of our proposed method is to show that, with the exception of the $\hat{p}(e|m)$ feature, a neural network is able to learn the best features for ED without requiring expert input.

To gain further insight, we analyzed the accuracy on the AIDA-B dataset for situations where gold entities have low frequency or mention prior. Table 6 shows that our method performs well in these harder cases.

Mention	Gold entity	$\hat{p}(e\|m)$ of gold entity	Attended contextual words
Scotland	Scotland national rugby union team	0.034	England Rugby team squad Murrayfield Twickenham national play Cup Saturday World game George following Italy week Friday selection dropped row month
Wolverhampton	Wolverhampton Wanderers F.C.	0.103	matches League Oxford Hull league Charlton Oldham Cambridge Sunderland Blackburn Sheffield Southampton Huddersfield Leeds Middlesbrough Reading Coventry Darlington Bradford Birmingham Enfield Barnsley
Montreal	Montreal Canadiens	0.021	League team Hockey Toronto Ottawa games Anaheim Edmonton Rangers Philadelphia Caps Buffalo Pittsburgh Chicago Louis National home Friday York Dallas Washington Ice
Santander	Santander Group	0.192	Carlos Telmex Mexico Mexican group firm market week Ponce debt shares buying Televisa earlier pesos share stepped Friday analysts ended
World Cup	FIS Alpine Ski World Cup	0.063	Alpine ski national slalom World Skiing Whistler downhill Cup events race consecutive weekend Mountain Canadian racing

Table 7: Examples of context words selected by our local attention mechanism. Distinct words are sorted decreasingly by attention weights and only words with non-zero weights are shown.

7.5 Hyperparameter Studies

In Table 5, we analyze the effect of two hyperparameters. First, we see that hard attention (i.e. $R < K$) helps reducing the noise from uninformative context words (as opposed to keeping all words when $R = K$).

Second, we see that a small number of LBP iterations (hard-coded in our network) is enough to obtain good accuracy. This speeds up training and testing compared to traditional methods that run LBP until convergence. An explanation is that a truncated version of LBP can perform well enough if used at both training and test time.

7.6 Qualitative Analysis of Local Model

In Table 7 we show some examples of context words attended by our local model for correctly solved hard cases (where the mention prior of the correct entity is low). One can notice that words relevant for at least one entity candidate are chosen by our model in most of the cases.

7.7 Error Analysis

We analyse some of the errors made by our model on the AIDA-B dataset. We mostly observe three situations: i) annotation errors, ii) gold entities that do not appear in mentions' candidate sets, or iii) gold entities with very low $p(e|m)$ prior whose mentions have an incorrect entity candidate with high prior. For example, the mention "Italians" refers in some specific context to the entity "Italy national football team" rather than the entity representing the country. The contextual information is not strong enough in this case to avoid an incorrect prediction. On the other hand, there are situations where the context can be misleading, e.g. a document heavily discussing about cricket will favor resolving the mention "Australia" to the entity "Australia national cricket team" instead of the gold entity "Australia" (naming a location of cricket games in the given context).

8 Conclusion

We have proposed a novel deep learning architecture for entity disambiguation that combines entity embeddings, a contextual attention mechanism, an adaptive local score combination, as well as unrolled differentiable message passing for global inference. Compared to many other methods, we do not rely on hand-engineered features, nor on an extensive corpus for entity co-occurrences or relatedness. Our system is fully differentiable, although we chose to pre-train word and entity embeddings. Extensive experiments show the competitiveness of our approach across a wide range of corpora. In the future, we would like to extend this system to perform nil detection, coreference resolution and mention detection.

Our code and data are publicly available: http://github.com/dalab/deep-ed

Acknowledgments

We thank Aurelien Lucchi, Marina Ganea, Jason Lee, Florian Schmidt and Hadi Daneshmand for their comments and suggestions.

This research was supported by the Swiss National Science Foundation (SNSF) grant number 407540_167176 under the project "Conversational Agent for Interactive Access to Information".

References

Diego Ceccarelli, Claudio Lucchese, Salvatore Orlando, Raffaele Perego, and Salvatore Trani. 2013. Learning relatedness measures for entity linking. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 139–148. ACM.

Xiao Cheng and Dan Roth. 2013. Relational inference for wikification. *Urbana*, 51(61801):16–58.

Andrew Chisholm and Ben Hachey. 2015. Entity disambiguation with web links. *Transactions of the Association for Computational Linguistics*, 3:145–156.

Emily Denton, Jason Weston, Manohar Paluri, Lubomir Bourdev, and Rob Fergus. 2015. User conditional hashtag prediction for images. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1731–1740. ACM.

Justin Domke. 2011. Parameter learning with truncated message-passing. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2937–2943. IEEE.

Justin Domke. 2013. Learning graphical model parameters with approximate marginal inference. *IEEE transactions on pattern analysis and machine intelligence*, 35(10):2454–2467.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159.

Wei Fang, Jianwen Zhang, Dilin Wang, Zheng Chen, and Ming Li. 2016. Entity disambiguation by knowledge and text jointly embedding. *CoNLL 2016*, page 260.

Paolo Ferragina and Ugo Scaiella. 2010. Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1625–1628. ACM.

Matthew Francis-Landau, Greg Durrett, and Dan Klein. 2016. Capturing semantic similarity for entity linking with convolutional neural networks. *arXiv preprint arXiv:1604.00734*.

Evgeniy Gabrilovich, Michael Ringgaard, and Amarnag Subramanya. 2013. Facc1: Freebase annotation of clueweb corpora, version 1 (release date 2013-06-26, format version 1, correction level 0). *Note: http://lemurproject.org/clueweb09/FACC1/Cited by*, 5.

Octavian-Eugen Ganea, Marina Ganea, Aurelien Lucchi, Carsten Eickhoff, and Thomas Hofmann. 2016. Probabilistic bag-of-hyperlinks model for entity linking. In *Proceedings of the 25th International Conference on World Wide Web*, pages 927–938. International World Wide Web Conferences Steering Committee.

Amir Globerson, Nevena Lazic, Soumen Chakrabarti, Amarnag Subramanya, Michael Ringgaard, and Fernando Pereira. 2016. Collective entity resolution with multi-focal attention. In *ACL (1)*.

Zhaochen Guo and Denilson Barbosa. 2016. Robust named entity disambiguation with random walks.

Zhengyan He, Shujie Liu, Mu Li, Ming Zhou, Longkai Zhang, and Houfeng Wang. 2013. Learning entity representation for entity disambiguation. In *ACL (2)*, pages 30–34.

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 *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 782–792. Association for Computational Linguistics.

Hongzhao Huang, Larry Heck, and Heng Ji. 2015. Leveraging deep neural networks and knowledge graphs for entity disambiguation. *arXiv preprint arXiv:1504.07678*.

Heng Ji. 2016. Entity discovery and linking reading list.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

John Lafferty, Andrew McCallum, Fernando Pereira, et al. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the eighteenth international conference on machine learning, ICML*, volume 1, pages 282–289.

Nevena Lazic, Amarnag Subramanya, Michael Ringgaard, and Fernando Pereira. 2015. Plato: A selective context model for entity resolution. *Transactions of the Association for Computational Linguistics*, 3:503–515.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

David Milne and Ian H Witten. 2008. Learning to link with wikipedia. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 509–518. ACM.

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

Fifteenth conference on Uncertainty in artificial intelligence, pages 467–475. Morgan Kaufmann Publishers Inc.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–43.

Maria Pershina, Yifan He, and Ralph Grishman. 2015. Personalized page rank for named entity disambiguation.

Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 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-Volume 1*, pages 1375–1384. Association for Computational Linguistics.

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

Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448.

Yaming Sun, Lei Lin, Duyu Tang, Nan Yang, Zhenzhou Ji, and Xiaolong Wang. 2015. Modeling mention, context and entity with neural networks for entity disambiguation. In *IJCAI*, pages 1333–1339.

Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2016. Joint learning of the embedding of words and entities for named entity disambiguation. *CoNLL 2016*, page 250.

Stefan Zwicklbauer, Christin Seifert, and Michael Granitzer. 2016. Robust and collective entity disambiguation through semantic embeddings. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 425–434. ACM.