Entity Linking via Joint Encoding of Types, Descriptions, and Context

Nitish Gupta[*]
University of Pennsylvania
Philadelphia, PA
nitishg@seas.upenn.edu

Sameer Singh
University of California
Irvine, CA
sameer@uci.edu

Dan Roth[*]
University of Pennsylvania
Philadelphia, PA
danroth@seas.upenn.edu

Abstract

For accurate entity linking, we need to capture various information aspects of an entity, such as its description in a KB, contexts in which it is mentioned, and structured knowledge. Additionally, a linking system should work on texts from different domains without requiring domain-specific training data or hand-engineered features.

In this work we present a neural, modular entity linking system that learns a unified dense representation for each entity using multiple sources of information, such as its description, contexts around its mentions, and its fine-grained types. We show that the resulting entity linking system is effective at combining these sources, and performs competitively, sometimes out-performing current state-of-the-art systems across datasets, without requiring any domain-specific training data or hand-engineered features. We also show that our model can effectively "embed" entities that are new to the KB, and is able to link its mentions accurately.

1 Introduction

Entity linking, the task of identifying the real-world entity a mention in text refers to, provides the ability to ground text to existing knowledge bases, and thus supports multiple natural language understanding, and knowledge acquisition tasks.

A key challenge for successful entity linking is the need to capture semantic and background information at various levels of granularity. For example, to resolve the mention "India" in "**India** plays a match in England today" to the correct entity, `India_cricket_team`, one needs to use mention-level context to identify that the sentence refers to a sports team (using *plays* and *match*), use document-level context to identify the sport, and information about the entity to realize that `India_cricket_team` is a sports team and the string "India" may refer to it. The problem has been studied extensively by employing a variety of machine learning, and inference methods, including a pipeline of deterministic modules (Ling et al., 2015), simple classifiers (Cucerzan, 2007; Ratinov et al., 2011), graphical models (Durrett and Klein, 2014), classifiers augmented with ILP inference (Cheng and Roth, 2013), and more recently, neural approaches (He et al., 2013; Sun et al., 2015; Francis-Landau et al., 2016).

We present a neural approach to linking[1] that learns a dense unified representation of entities by encoding the semantic and background information from multiple sources – encyclopedic entity descriptions, entity-type information, and the contexts the entity occurs in – thus capturing different aspects of the "meaning" of an entity. Hence, we overcome the shortcomings of several existing models that do not capture all these aspects. For example, methods, such as Vinculum (Ling et al., 2015), do not make use of the local context of the mention ("plays" and "match") while others, such as Berkeley-CNN (Francis-Landau et al., 2016), do not take entity-types into account. Our proposed model uses compositional training to ensure that the learned entity representation captures the various information sources available to it, making it quite modular. Specifically, we introduce encoders for the different sources of information about the entity, and encourage the entity embedding to be similar to all of the encoded representations.

A key requirement for information extraction systems is their ability to work across texts from

[*]Work performed while these authors were at UIUC.

[1] The source code and the datasets are available at https://nitishgupta.github.io/neural-el

various domains. Some methods (Francis-Landau et al., 2016; Nguyen et al., 2016; Hoffart et al., 2011) train parameters on domain-specific linked data, thus hampering their ability to generalize to new domains. By only making use of indirect supervision that is available in Wikipedia/Freebase, we refrain from using domain specific training data, and produce a domain-independent linking system. Our comprehensive evaluation on recent entity linking benchmarks reveals that the resulting entity linker compares favorably to state-of-the-art systems across datasets, even those that have hand-engineered features or use dataset-specific training. We hence show that our model not only leverages all the available information for each entity effectively, but is also robust to missing information, such as entities without links/description in Wikipedia or with incomplete entity types.

In the real-world, new entities are regularly added to the knowledge bases, thus, it is important for any entity linking system to be extendable to such entities, especially the ones that do not have any existing linked mentions. By the virtue of our model's modular nature, it can easily incorporate new entities not present during training. Specifically, we show that our model can perform accurate linking for new entities, without having to re-train the existing entity representations, only using their description and types.

2 Related Work

Existing approaches for entity linking differ in several ways, including the machine learning models, the types of training data, and the kinds of information used about the entities.

Many existing approaches use links and information from Wikipedia as the only source of supervision to build the entity linking system. These approaches use sparse entity and mention-context representations, such as, based on the Wikipedia categories (Cucerzan, 2007), weighted bag of words in the entity description and mention context (Kulkarni et al., 2009; Ratinov et al., 2011), hand crafted features based on partial string matches, punctuations in entity name (McNamee et al., 2009), etc. Heuristics (Mihalcea and Csomai, 2007) or linear classifiers (Bunescu and Pasca, 2006; Cucerzan, 2007; Ratinov et al., 2011; McNamee et al., 2009) are used over these features to rank entity candidates for linking. Recently, neural models have been proposed as a way to support better generalization over the sparse features; e.g., using feed-forward networks on bag-of-words of the entity context (He et al., 2013), or using entity-class information from KB (Sun et al., 2015).

Some models ignore the entity's description on Wikipedia, but rather, only rely on the context from links to learn entity representations (Lazic et al., 2015), or use a pipeline of existing annotators to filter entity candidates (Ling et al., 2015). Our model is similar to these approaches by only using information from Wikipedia; however, we do not use hand-crafted features, and use multiple sources of information such as local and document-level entity context, KB descriptions, and entity types, to learn explicit entity representation.

Few recent entity linking approaches (Hoffart et al., 2011; Durrett and Klein, 2014; Nguyen et al., 2016; Francis-Landau et al., 2016) use manually-annotated domain specific training data to learn the linking system. AIDA (Hoffart et al., 2011), for example, evaluate their system on test set from CoNLL-YAGO dataset but also train on the training data from the same dataset. Berkeley-CNN (Francis-Landau et al., 2016), that uses CNNs operating over different granularity of entity and mention contexts, also follows this training regime and trains separate models for each dataset. Such approaches can be prohibitive in many applications as it encourages the model to over-fit to the peculiarities of different datasets and domains.

Other forms of information, apart from descriptions, and context from linked data, are also utilized for linking. Many approaches perform joint inference over the linking decisions in a document (Milne and Witten, 2008; Ratinov et al., 2011; Hoffart et al., 2011; Globerson et al., 2016), identify mentions that do not link to any existing entity (`NIL`) (Bunescu and Pasca, 2006; Ratinov et al., 2011), and cluster `NIL`-mentions (Wick et al., 2013; Lazic et al., 2015) to discover new entities. Few approaches jointly model entity linking, and other related NLP tasks to improve linking, such as, coreference resolution (Hajishirzi et al., 2013), relational inference (Cheng and Roth, 2013), and joint coreference with typing (Durrett and Klein, 2014). In our model, we use fine-grained type information of the entity as an auxiliary distant supervision to improve mention-context representation but do not use intermediate typing decisions for linking.

Many approaches that learn entity embeddings for other applications have also been proposed,

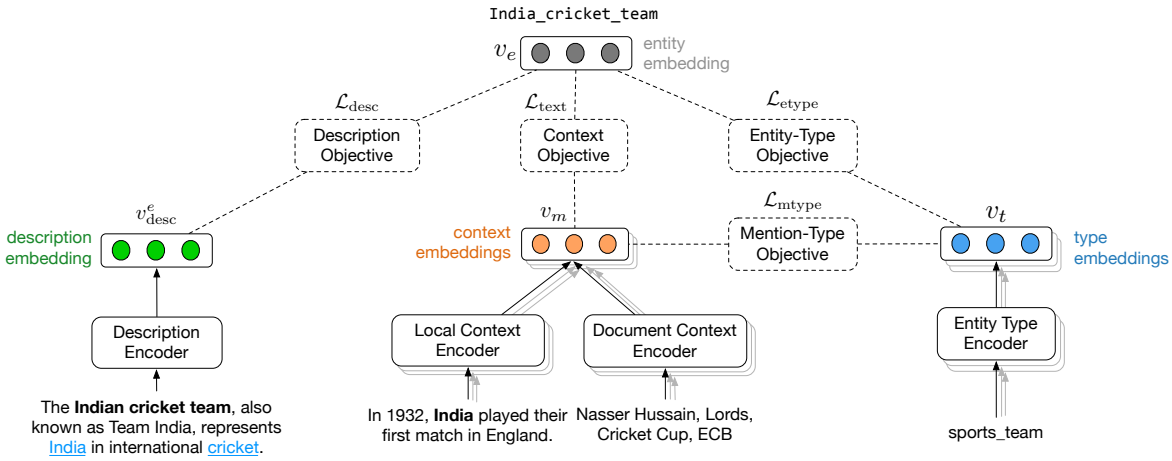

Figure 1: **Overview of the Model** (§ 3): Each entity has a Wikipedia description, linked mentions in Wikipedia (only one shown), and fine-grained types from Freebase (only one shown). We encode local and document-level mention contexts (§ 3.1), entity-description (§ 3.2), and fine-grained entity-types (§ 3.3 & § 3.4). Joint optimization (§ 3.5) over these provides the unified entity representations $\{v_e\}$.

such as, from the structured KB for KB completion (Bordes et al., 2011, 2013; Yang et al., 2014; Lin et al., 2015), or from both structured KBs, and text for relation extraction (Toutanova et al., 2016; Verga et al., 2016a). However, since it is not trivial for these models to incorporate new entities to the KB, few recent approaches alleviate this issue by representing entities as a composition of words in their names (Socher et al., 2013), relations they participate in (Verga et al., 2016b), or their types (Das et al., 2017), but do not use multiple sources of information jointly. In our work, we use structured knowledge (types) as well as unstructured knowledge (description and context) to learn entity embeddings for entity linking, and show that it extends to new entities.

3 Jointly Embedding Entity Information

Knowledge bases contain different kinds of information about entities such as textual description, linked mentions (in Wikipedia), and types (in Freebase). For accurate linking, it is often necessary to combine information from these various sources. Here, we describe our model that encodes information about the set of entities $\mathcal{E}$ using dense unified representation for linking ($v_e \in \mathbb{R}^d, \forall e \in \mathcal{E}$). In particular, we use existing mentions in Wikipedia to encode the context (§ 3.1), textual descriptions from Wikipedia to encode background information (§ 3.2), and fine-grained types from Freebase as structured topical knowledge (§ 3.3). Figure 1 provides an overview of our model.

3.1 Encoding the Mention Context, C

Consider the example mention in Figure 1 that contains two mentions, "India" and "England". In order to disambiguate "India" to the correct entity, a linking system would need to utilize both the local context (*played* and *match*), and the document context (to identify the sport). However, the model needs to represent context such that the semantics are preserved, e.g. "England" should not be linked to a sports team even though it shares the context with "India". In this section, we describe how we encode these two types of context, using a LSTM-based encoder to capture the lexical and syntactical local-context of a mention (v_m^{local}), and a feed-forward network to encode the document-level topical knowledge (v_m^{doc}), and combine them in a single representation for each mention (v_m).

Local-Context Encoder Given a mention m in the sentence $s = w_1, \ldots, m, \ldots, w_N$, we use LSTM encoders on the left $(w_1, \ldots, m)$ and right $(m, \ldots, w_N)$ contexts of the mention separately, and then combine it to form the local context representation of the mention (Fig. 2). More precisely, we formulate an LSTM as $h_i, s_i = \text{LSTM}(u_i, h_{i-1}, s_{i-1})$, $u_i \in \mathbb{R}^{d_w}$ is the input embedding of the i-th token in the sequence, and $h_{i-1}, s_{i-1} \in \mathbb{R}^l$ is the previous output and the cell state of the LSTM, respectively. The left-LSTM is applied to the sequence $(w_1, \ldots, m)$ with the last output $\overrightarrow{h^l_m}$, while a different right-LSTM is ap-

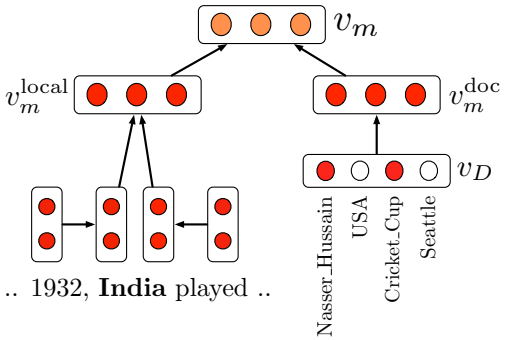

Figure 2: Overview of the mention context encoder

plied to the sequence $(w_N, \ldots, m)$[2] to produce $\overleftarrow{h}_m^r$. We concatenate these output $[\overrightarrow{h}_m^l, \overleftarrow{h}_m^r]$ and pass it through a single layer feed-forward network[3] to produce the local context representation of the mention (v_m^{local}), where $v_m^{\text{local}} \in \mathbb{R}^{D_m}$. Note that this encoder will produce different representations for different mentions in the same sentence.

Document-Context Encoder To represent the document context of a mention m, we use a bag-of-mention surfaces representation, $v_D \in \{0,1\}^{|V_G|}$, of the document, similar to Lazic et al. (2015). The vocabulary V_G consists of all mention surfaces seen in our training data, e.g. *USA*, *Nasser_Hussain*, *Pearl_Jam* etc. Such a representation helps capture the topical and entity coherence information in the document by utilizing co-occurrence between entity surface forms. This sparse vector v_D of bag-of-mention surfaces is compressed to a low-dimensional representation $v_m^{\text{doc}} \in \mathbb{R}^{D_m}$ using a single layer feed-forward network.

Mention-Context Encoder We combine the local (v_m^{local}) and document (v_m^{doc}) level context vectors by concatenating them, and passing them through a single-layer feed-forward network to obtain the mention context embedding $v_m \in \mathbb{R}^d$. In order to learn the entity representation v_e such that it encodes all of its mentions' contexts, we introduce an objective that encourages the context representation v_m to be similar to v_e (where mention m is a link to entity e), and dissimilar to other candidates[4]. Precisely, we maximize the probability of predicting the correct entity from the mention-context vector as $P_{\text{text}}(e|m) = \frac{\exp(v_m \cdot v_e)}{\sum_{c_k \in C_m} \exp(v_m \cdot v_{c_k})}$, where C_m is the set of candidate entities. Given all the mentions in Wikipedia, we jointly optimize the entity representations, and the context encoders by maximizing the following log-likelihood:

$$\mathcal{L}_{\text{text}} = \frac{1}{M} \sum_{i=1}^{M} \log P_{\text{text}}(e_{m^{(i)}} | m^{(i)}) \qquad (1)$$

where $m^{(i)}$ is the i^{th} mention in the linked data, and $e_{m^{(i)}}$ is the entity the mention refers to.

3.2 Encoding Entity Description, D

The textual description about entities in Wikipedia can provide a useful source of background information about the entity, and thus has been used in many existing linking systems. Given the description as a sequence of words, we first embed each word to a d_w-dimensional vector resulting in a sequence of vectors $w_1, \ldots, w_n$. To encode this description as a fixed size vector, we use a Convolution Neural Network (CNN), similar to Francis-Landau et al. (2016), with global average pooling, to obtain $v_{\text{desc}}^e \in \mathbb{R}^d$.

In order for the entity representation v_e to encode its description, we use a similar objective as in the previous section § 3.1, i.e. we maximize the probability $P_{\text{desc}}(e|v_{\text{desc}}^e)$, and learn the parameters by maximizing the log-likelihood $\mathcal{L}_{\text{desc}}$, defined similarly as (1).

3.3 Encoding Fine-Grained Types, E

Fine-grained types provide a source of structured information that is quite readily available, often more easily than the description or linked data (e.g. Freebase contains tens of millions of entities with types but Wikipedia only contains descriptions for a few million). These types have been shown to be quite useful for linking (Ling et al., 2015), since an accurate prediction of types from the mention, and its match with the entity types can often resolve many challenging ambiguities.

Here, we focus on being able to represent the different types at the entity level, leaving mention-level type information to the next section. Each entity has multiple types $T_e \subset \mathcal{T}$ from the type set $\mathcal{T}$ introduced by Ling and Weld (2012). We compute the probability $P(t|e)$ of type t being relevant to entity e as $\sigma(v_t \cdot v_e)$, where σ is the sigmoid function, $v_e \in \mathbb{R}^d$ is the entity representation, and $v_t \in \mathbb{R}^d$ is the embedding of type t in $\mathcal{T}$. We maximize the log-likelihood of the type information to

[2] We reverse the token sequence in the right context so that right-LSTM starts at the last token and ends at the mention.

[3] We use rectified linear unit (ReLU) as the non-linear activation throughout this paper.

[4] Details on candidate generation in Sec 4

jointly learn entity and type representations:

$$\mathcal{L}_{\text{etype}} = \frac{1}{|\mathcal{E}|} \sum_{e \in \mathcal{E}} \log \prod_{t \in T_e} P(t|e) \prod_{t' \notin T_e} (1 - P(t'|e))$$

3.4 Type-Aware Context Representation, T

Apart from being able to represent the types of the entities, it is also important for our linker to be able to represent the type information at the mention level. In the example in Fig. 1, although the mention "India" is prominently used to refer to the *country*, it is evident from the sentence that it refers to a *Sports Team*. The context-encoder captures this information in an unstructured manner, thus it will be useful for the encoder to directly utilize this supervision. This is a similar setup as Ling et al. (2015) and Shimaoka et al. (2017) that use noisy distant supervision to train a fine-grained type predictor for mentions.

In order for the context encoders, and type embeddings to directly inform each other, we introduce an objective $\mathcal{L}_{\text{mtype}}$ between every v_m and v_t if type t belongs to T_e for the entity e that m refers to. This objective is similar to $\mathcal{L}_{\text{etype}}$ from § 3.3.

3.5 Learning Unified Entity Representations

In the sections above we described different encoder models to capture entity-context information (local- and document-level), entity-description from a KB, and fine-grained types in a single entity representation vector. To learn the entity representations, and parameters of the encoders, we jointly maximize the total objective:

$$\{v_e\}, \Theta = \underset{\{v_e\}, \Theta}{\text{argmax}}\, \mathcal{L}_{\text{text}} + \mathcal{L}_{\text{desc}} + \mathcal{L}_{\text{etype}} + \mathcal{L}_{\text{mtype}}$$

where $\{v_e\}$ is the set of entity representations, and Θ is set of parameters for the different encoders. One advantage of having such a joint, modular objective is that it is robust to missing information, i.e. entities with missing mentions, types, or descriptions will still obtain accurate representations learned using other sources of information.

4 Entity Linking

Given a document, and mentions marked in it for disambiguation, we perform a two-step procedure to link them to an entity. First, we find a set of candidate entities, and their *prior* scores using a pre-computed dictionary. We then use our mention-context encoder to estimate the semantic similarity of each mention with the vector representations of each entity candidate, and combine the results from the two sources for making linking decisions.

A typical KB contains millions of entities, which makes it prohibitively expensive to compute a similarity score between each mention and all entities in the KB. Prior work has shown that, for a given mention, aggressively pruning the set of possible entities to a small subset hurts performance only negligibly, while making the linker extremely efficient. For each mention m, we generate a set of candidate entities $C_m = \{c_j\} \subset \mathcal{E}$ using CrossWikis (Spitkovsky and Chang, 2012), a dictionary computed from a Google crawl of the web that stores the frequency with which a mention links to a particular entity. To generate C_m we choose the top-30 entities for each mention string, and normalize this frequency across the chosen candidates to compute $P_{\text{prior}}(e|m)$. In the literature, such a dictionary is often built from the anchor links in Wikipedia (Ratinov et al., 2011; Hoffart et al., 2011) but Ling et al. (2015) show using CrossWikis gives improved prior scores and candidate recall.

For each mention m, we use our learned mention-context encoder from § 3.1 to encode the mention's context as v_m, and estimate the distribution over the candidates using $P_{\text{text}}(e|m)$. We treat these two pieces of evidence; pre-computed prior probability, and the context-based probability, as independent, disjunctive sources of signal, and thus combine them to compute $P(e|m)$ as:

$$\begin{aligned} P(e|m) &= P_{\text{prior}}(e|m) + P_{\text{text}}(e|m) \\ &\quad - (P_{\text{prior}}(e|m) * P_{\text{text}}(e|m)) \qquad (2) \\ \hat{e}_m &= \underset{e \in C_m}{\text{argmax}}\; P(e|m) \qquad (3) \end{aligned}$$

where $\hat{e}_m$ is the predicted entity that the mention m should be disambiguated to.

5 Evaluation Setup

Here we provide a detailed description of how we train our models, benchmark datasets, linking systems we compare to, and the evaluation metrics.

Training Data Our primary source of information about the entities is Wikipedia (dump dated 2016/09/20). We use existing links in Wikipedia, with the anchors as mentions, and links as the true entity, as input to the context encoder (see § 3.1). As the description of each entity (§ 3.2), we use the first 100 tokens of the entity's Wikipedia page

(same as Francis-Landau et al. (2016)). To obtain entity types (see § 3.3), we extract the types for each entity from Freebase and map them to the 112 fine-grained types introduced by Ling and Weld (2012). For context and description encoders, we use pre-trained 300-dimensional case-sensitive word embeddings by Pennington et al. (2014) as the first layer that is not updated during training.

Hyper-parameters We perform coarse-grained tuning of the hyper-parameters using a fraction of the training data. The vectors for the entities, types, contexts, and descriptions are of size $d = 200$. The size of the local context encoder LSTM hidden layer l, local context output, and the document-context encoder output D_m is set to $100 (= l = D_m)$. The document context vocabulary contains $|V_G| = 1.5$ million strings. We use dropout (Srivastava et al., 2014) with a probability of 0.4. Additionally, we use word-dropout where we replace a random subset of tokens (mention-strings) in the local (document) context with "unk" (rate of 0.4 and 0.6 for local and document context respectively). We use Adam (Kingma and Ba, 2014) for optimization, with learning rate 0.005 and mini-batches of size 1000.

Existing Approaches We compare our approach to the following five entity-linking models: (1) **Plato** (Lazic et al., 2015), an unsupervised generative model that uses indirect-supervision from Wikipedia and an additional corpus of 50 million unlabeled webpages, (2) **Wikifier** (Ratinov et al., 2011), an unsupervised linker that uses hand-crafted features to rank candidates, (3) **Vinculum** (Ling et al., 2015), a modular, unsupervised pipeline system, (4) **AIDA** (Hoffart et al., 2011), a supervised linker trained on CoNLL data and uses hand-crafted features, and (5) **BerkCNN** (Francis-Landau et al., 2016), a recent neural supervised approach that has variants that use hand-crafted features.

Evaluation Setup We evaluate our approach on the following four datasets: CoNLL-YAGO (Hoffart et al., 2011), ACE 2004 (NIST, 2004; Ratinov et al., 2011), ACE 2005 (NIST, 2005; Bentivogli et al., 2010), and Wikipedia (Ratinov et al., 2011). For each of these datasets, we use the standard test/development splits, but do not use any information from the training splits. End-to-end entity linking systems such as Vinculum and Wikifier perform an NER-style F1 evaluation where a prediction is only considered correct if the system mention boundaries match the gold annotation, and the predicted link is correct (we compare against these by extracting mentions with Stanford-NER). On the other hand, systems like Plato, AIDA, and Berkeley-CNN assume mentions are provided, and evaluate using the linking accuracy for gold-mentions. Further, the approaches we compare here (including ours) do not predict NIL entities for the datasets evaluated on.

	CoNLL		ACE05	Wiki
	Test	Dev		
Plato (Sup)	79.7	-	-	-
Plato (Semi-Sup)	86.4	-	-	-
*AIDA**	*81.8*	-	-	-
*BerkCNN:Sparse**	*74.9*	-	*83.6*	*81.5*
*BerkCNN:CNN**	*81.2*	*86.91*	*84.5*	*75.7*
*BerkCNN:Full**	*85.5*	-	*89.9*	*82.2*
Priors	68.5	70.9	81.1	78.1
Model C	81.4	83.4	83.7	86.1
Model CD	81.0	83.2	85.8	86.1
Model CT	82.3	83.9	86.5	88.2
Model CDT	82.5	85.6	86.8	88.0
Model CDTE	82.9	84.9	85.6	89.0

Table 1: **Entity Linking Performance:** Accuracy of existing systems, and variations of our model on gold mentions. The model using context information is labeled C, entity-description as D, context-typing as T, and entity-type encoding as E. Existing models marked in *Italics** train domain-specific linkers for each dataset. Our system performs competitively to these systems, and outperforms Plato (Sup) that uses the same indirect supervision.

6 Results

In this section we present various experiments to evaluate the performance of our proposed entity-linking system. Specifically, we focus on the following questions: (1) how effective is our model in combining different information on standard linking benchmarks, without requiring domain specific information (§ 6.1), (2) is our model able to accommodate unseen entities by using their types, or description, without re-training the entity representations (§ 6.2), and (3) how does the model perform on fine-grained mention typing, a task it is not directly trained for, compared to approaches designed for the task (§ 6.3). Further, Sec 6.4 presents examples to show the effect of encoding different kinds of information in a unified entity representation.

	F1	Accuracy
AIDA	77.8	-
Wikifier	85.1	-
Vinculum	88.5	-
Model C	88.9	93.1
Model CDT	89.8	93.9
Model CDTE	90.7	94.3

Table 2: **Results for ACE-2004**: F1 is calculated for predicted mentions, and accuracy on gold-mentions. Results for Wikifier and AIDA are from (Ling et al., 2015). All systems use the same mention extraction protocol showing the difference in F1 is due to linking performance.

6.1 Entity Linking

In Table 1 we present linking accuracy for our models that vary in the information they use. We see that the model that only encodes the context-information, Model C ($\mathcal{L} = \mathcal{L}_{\text{text}}$) consistently performs better than picking the entity with the highest prior probability from CrossWikis, indicating that the model is able to utilize the context across datasets. On incorporating the description with context (Model CD) we see improvement in the performance on ACE-2005, but slight decrease in CoNLL, suggesting the entity descriptions are not extremely useful for the latter (it contains rare entities, many short and incomplete sentences, and specific entities as annotations for metonymic mentions, as also observed by Ling et al. (2015)). On introducing the entity type-aware loss in Model CT to the context-only model, we see significantly improved results for all datasets, demonstrating that explicitly modeling fine-grained types helps learning a better context encoder and, in turn, type-aware entity representations. Combining descriptions with this model (Model CDT) shows further gains in accuracy indicating that our model is able to exploit complementary information from the two sources. Finally, on introducing explicit entity-type encoding, Model CDTE performs the best on two of the four datasets. As we will see in § 6.2, encoding entity-type information also allows our models to easily generalize to new entities.

On comparison to existing systems we see that all our variants outperform Plato's indirectly-supervised model trained on Wikipedia, which is the same information our Model C and CD use. Their semi-supervised model, that is additionally trained on 50 million web-pages, performs much better. In comparison to AIDA and Berkeley-CNN, that train separate models on respective datasets, we perform better than AIDA and Berkeley-CNN's *sparse* and neural model. On combining features from CNN to the *sparse* model, the Berkeley-CNN models for each dataset outperform our model, but are unlikely to generalize across the datasets[5].

In Table 2 we present results for our models on ACE-2004. Our model outperforms the Wikifier and Vinculum systems that only use information from Wikipedia, and AIDA, by a significant margin, indicating its possible over-fitting to the CoNLL domain. Hence, it shows our model's ability to perform accurate linking across different datasets without using domain-specific information.

Method	Accuracy
Random Guessing	16.7
Random Embeddings	34.0
Entity Description	65.1
Fine-Grained Types	73.7
Description + Types	79.5

Table 3: **Cold-Start Entities:** Linking new entities by using different information to learn their embeddings. Our model is able to jointly utilize description and type information better.

6.2 Cold-Start Entities

In realistic situations, new entities are regularly added to the knowledge base with little or no linked data for them. Hence, it is important for any information extraction system that learns entity representations to be easily extendable for such entities without needing to be re-trained. In this section, we consider the use of our approach to this setting.

In particular, for each such new entity, we need to determine their embedding using only their description and/or type information. For a new entity for which only the description is available, we directly set its embedding to be the output of the entity-description encoder without any need for learning. If only fine-grained types are available, we learn the new entity-embedding by optimizing the objective $\mathcal{L}_{\text{etype}}$. In case both description and types are available, we jointly maximize the similarity of the entity embedding with the output of the entity-description, and the type encoders (i.e. optimize $\mathcal{L}_{\text{desc}}$ and $\mathcal{L}_{\text{etype}}$). Note that we only learn the embeddings of each new entity, keeping all other

[5] Ling et al. (2015) show that AIDA is unable to perform well on datasets it has not been trained on.

Models	Acc.	Macro F1	Micro F1
FIGER	47.4	69.2	65.5
SSIR-LSTM	55.6	75.1	71.7
SSIR-Full	59.6	78.9	75.3
Our Model	57.7	72.8	72.1

Table 4: **Typing Prediction:** Performance on the FIGER (GOLD) dataset. Our performance is competitive with FIGER (Ling and Weld, 2012) and neural-LSTM model of Shimaoka et al. (2017). Their SSIR-Full model that uses a biLSTM layer, an attention layer, combined with hand-crafted features is state-of-art for this task.

parameters of our model (Model CDTE) fixed.

To evaluate this setting of new entities, we randomly select 1000 rare entities from Wikipedia that are not used during training. Among all mentions of these entities in Wikipedia, we only keep the mentions for which our candidate generation generates more than one candidate, resulting in 3791 mentions. On average, each mention had 6 candidate entities, and further, as priors are not available in this setting, we only rely on the context probability for linking, making this a challenging task.

We present the results of using different types of information about the entity for this data in Table 3. It is surprising that randomly initialized embeddings for these new entities perform better than random guessing, suggesting our model is sometimes able to eliminate the wrong candidates purely based on their learned embedding, i.e. an entity with a random embedding has a higher likelihood of being the correct entity. More importantly, we see that our model variants that utilize the available entity information are able to link much more accurately (47-60% error reduction). Further, using both description and types results in the best embeddings for these new entities ($\sim 80\%$ accuracy).

6.3 Fine-Grained Typing

Since entity embeddings are trained to be both, context and type-aware, we evaluate whether they can be used to predict fine-grained types for mentions from context (using v_m and v_t). Compared to existing systems trained specifically for this task, embeddings from our approach (Model CDTE) performs competitively (see Table 4). In particular, our model performs better than the neural-LSTM model of Shimaoka et al. (2017), suggesting that our multi-task linking, and typing loss facilitates effective encoding of mention contexts.

12th Asian Nations Cup finals are hosted by **Lebanon** until this October 29. **Model CD:** `Lebanon_football_team` **Model CT:** `Lebanon` *(correct)* **Model CDTE:** `Lebanon` *(correct)*
Yugoslav midfielder **Petrovic** scored twice as PSV Eindhoven romped to a 6-0 win. **Model CD:** `Zeljko_Petrovic` *(correct)* **Model CT:** `Vladimir_Petrovic` **Model CDTE:** `Zeljko_Petrovic` *(correct)*
Ince was clambering over a wall at the Republican stadium during an under-21 clash. **Model CD:** `Ince` **Model CT:** `Tom_Ince` **Model CDTE:** `Paul_Ince` *(correct)*

Table 5: **Example predictions by our models:** Model CT (Ex.1) and CD (Ex.2) predict correctly when correct type prediction or background knowledge is sufficient, respectively. Only Model CDTE (Ex.3) predicts correctly when combination of context, types, and background knowledge is required.

6.4 Example Predictions

In Table 5 we show the prediction from different variants of our model for a few example mentions. In the first example, detecting the type of the mention is crucial, and thus we see both Model CT and CDTE are able to predict accurately. On the other hand, predicting the type of the mention is not especially useful in Example 2, and background factual knowledge from the entity description is needed (which models CD and CDTE are able to encode). Example 3 shows a challenging example where the appropriate combination of context, type prediction, and background knowledge is needed, that our Model CDTE is able to combine.

7 Conclusion

Motivated by the need to provide accurate entity-linking systems that are able to incorporate multiple sources of information, and do not require domain-specific datasets or hand-crafted features, we presented a novel neural approach to linking. We proposed a compositional training objective to learn unified entity embeddings that encode the variety of information available for each entity: its unstructured textual description, local and document contexts for its mentions, and sets of fine-grained types attached to it. The joint formulation allows the model to fruitfully combine the various sources of information, providing accurate linking on multiple datasets, generalization to new entities with

missing linked data, and the use of entity embeddings for related tasks such as type prediction.

There are a number of avenues for future work. Further research will include encoding more structured knowledge about the entities, such as their relations to other entities, to make their representations semantically richer. We will investigate how we can use unstructured resources, such as the corpus of unlabeled webpages used by Plato, and noisy supervision from the Wikilinks corpus (Singh et al., 2012) in order to further improve the model. We will also evaluate our approach on substantially varied domains, such as discussion forums, and social media posts.

Acknowledgments

We would like to thank Mike Lewis, Xiao Ling, Ananya, Shyam Upadhyay, and the anonymous EMNLP reviewers for their valuable feedback. This work is supported in part by the US Defense Advanced Research Projects Agency (DARPA) under contract FA8750-13-2-0008, and in part by an Adobe Research faculty award. The views expressed are those of the authors and do not reflect the official policy or position of the Dept. of Defense, the U.S. Government, or Adobe.

References

Luisa Bentivogli, Pamela Forner, Claudio Giuliano, Alessandro Marchetti, Emanuele Pianta, and Kateryna Tymoshenko. 2010. Extending english ace 2005 corpus annotation with ground-truth links to wikipedia. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *The Conference on Advances in Neural Information Processing Systems (NIPS)*.

Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. 2011. Learning structured embeddings of knowledge bases. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*.

Razvan Bunescu and Marius Pasca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of the European Chapter of the ACL (EACL)*.

Xiao Cheng and Dan Roth. 2013. Relational inference for wikification. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on Wikipedia data. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 708–716.

Rajarshi Das, Arvind Neelakantan, David Belanger, and Andrew McCallum. 2017. Chains of reasoning over entities, relations, and text using recurrent neural networks. In *Proceedings of the European Chapter of the ACL (EACL)*.

Greg Durrett and Dan Klein. 2014. A joint model for entity analysis: Coreference, typing, and linking. *Transactions of the Association for Computational Linguistics (TACL)*.

Matthew Francis-Landau, Greg Durrett, and Dan Klein. 2016. Capturing semantic similarity for entity linking with convolutional neural networks. In *Proceedings of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*.

Amir Globerson, Nevena Lazic, Soumen Chakrabarti, Amarnag Subramanya, Michael Ringgaard, and Fernando Pereira. 2016. Collective entity resolution with multi-focal attention. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Hannaneh Hajishirzi, Leila Zilles, Daniel S Weld, and Luke S Zettlemoyer. 2013. Joint coreference resolution and named-entity linking with multi-pass sieves. In *Proceedings of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*.

Zhengyan He, Shujie Liu, Mu Li, Ming Zhou, Longkai Zhang, and Houfeng Wang. 2013. Learning entity representation for entity disambiguation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

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 for Natural Language Processing (EMNLP)*.

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

Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan, and Soumen Chakrabarti. 2009. Collective annotation of wikipedia entities in web text. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*.

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 (TACL)*.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*.

Xiao Ling, Sameer Singh, and Daniel S Weld. 2015. Design challenges for entity linking. *Transactions of the Association for Computational Linguistics (TACL)*.

Xiao Ling and Daniel S Weld. 2012. Fine-grained entity recognition. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*.

Paul McNamee, Mark Dredze, Adam Gerber, Nikesh Garera, Tim Finin, James Mayfield, Christine D Piatko, Delip Rao, David Yarowsky, and Markus Dreyer. 2009. Hltcoe approaches to knowledge base population at tac 2009. In *TAC*.

Rada Mihalcea and Andras Csomai. 2007. Wikify!: Linking documents to encyclopedic knowledge. In *Proceedings of ACM Conference on Information and Knowledge Management (CIKM)*.

David Milne and Ian H. Witten. 2008. Learning to link with Wikipedia. In *Proceedings of ACM Conference on Information and Knowledge Management (CIKM)*, pages 509–518.

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 *Proceedings the International Conference on Computational Linguistics (COLING)*.

NIST. 2005. The ACE evaluation plan.

US NIST. 2004. The ace evaluation plan. *US National Institute for Standards and Technology (NIST)*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*.

L. Ratinov, D. Roth, D. Downey, and M. Anderson. 2011. Local and global algorithms for disambiguation to wikipedia. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Sonse Shimaoka, Pontus Stenetorp, Kentaro Inui, and Sebastian Riedel. 2017. Neural architectures for fine-grained entity type classification. In *Proceedings of the European Chapter of the ACL (EACL)*.

Sameer Singh, Amarnag Subramanya, Fernando Pereira, and Andrew McCallum. 2012. Wikilinks: A large-scale cross-document coreference corpus labeled via links to Wikipedia. Technical report, University of Massachusetts, Amherst.

Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *The Conference on Advances in Neural Information Processing Systems (NIPS)*.

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

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*.

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 *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

Kristina Toutanova, Xi Victoria Lin, Wen-tau Yih, Hoifung Poon, and Chris Quirk. 2016. Compositional learning of embeddings for relation paths in knowledge bases and text. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Patrick Verga, David Belanger, Emma Strubell, Benjamin Roth, and Andrew McCallum. 2016a. Multilingual relation extraction using compositional universal schema. In *Proceedings of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*.

Patrick Verga, Arvind Neelakantan, and Andrew McCallum. 2016b. Generalizing to unseen entities and entity pairs with row-less universal schema. In *Proceedings of the European Chapter of the ACL (EACL)*.

Michael Wick, Sameer Singh, Harshal Pandya, and Andrew McCallum. 2013. A joint model for discovering and linking entities. In *Proceedings of the 2013 Workshop on Automated Knowledge Base Construction*.

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*.