ELDEN: Improved Entity Linking using Densified Knowledge Graphs

Priya Radhakrishnan[+], Partha Talukdar[*] and Vasudeva Varma[+]

[+]IIIT Hyderabad, India
[*]Indian Institute of Science, Bangalore, India
priya.r@research.iiit.ac.in, ppt@iisc.ac.in, vv@iiit.ac.in

Abstract

Entity Linking (EL) systems aim to automatically map mentions of an entity in text to the corresponding entity in a Knowledge Graph (KG). Degree of connectivity of an entity in the KG directly affects an EL system's ability to correctly link mentions in text to the entity in KG. This causes many EL systems to perform well for entities well connected to other entities in KG, bringing into focus the role of KG density in EL. In this paper, we propose Entity Linking using Densified Knowledge Graphs (ELDEN). ELDEN is an EL system which first densifies the KG with co-occurrence statistics from a large text corpus, and then uses the densified KG to train entity embeddings. Entity similarity measured using these trained entity embeddings result in improved EL. ELDEN outperforms state-of-the-art EL system on benchmark datasets. Due to such densification, ELDEN performs well for sparsely connected entities in the KG too. ELDEN's approach is simple, yet effective. We have made ELDEN's code and data publicly available.

1 Introduction

Entity Linking (EL) is the task of mapping *mentions* of an *entity* in text to the corresponding entity in Knowledge Graph (KG) (Hoffart et al., 2011; Dong et al., 2014; Chisholm and Hachey, 2015). EL systems primarily exploit two types of information: (1) similarity of the mention to the candidate entity string, and (2) coherence between the candidate entity and other entities mentioned in the vicinity of the mention in text. Coherence essentially measures how well the candidate entity is connected, either directly or indirectly, with other KG entities mentioned in the vicinity (Milne and Witten, 2008; Globerson et al., 2016). In the state-
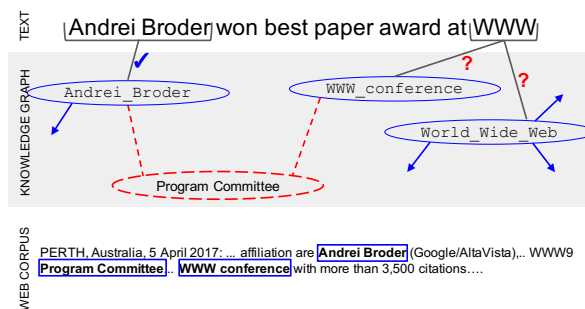

Figure 1: Improving entity disambiguation performance using KG densification: edges to WWW_conference, a sparsely-connected entity in the KG, is increased by adding edges from *pseudo entity* Program Committee whose mention co-occurs with it in web corpus. ELDEN, the system proposed in this paper, uses such densified KG to successfully link ambiguous mention *WWW* to the correct entity WWW_conference, instead of the more popular entity World Wide Web.

of-the-art EL system by (Yamada et al., 2016), coherence is measured as distance between embeddings of entities. This system performs well on entities which are densely-connected in KG, but not so well on sparsely-connected entities in the KG.

We demonstrate this problem using the example sentence in Figure 1. This sentence has two mentions: *Andrei Broder* and *WWW*. The figure also shows mention-entity linkages, i.e., mentions and their candidate entities in KG. Using a conventional EL system, the first mention *Andrei Broder*[1] can be easily linked to Andrei_Broder using string similarity between the mention and candidate entity strings. String similarity works well in this case as this mention is unambiguous in the given setting. However, the second mention

---

[1]We use *italics* to denote textual mentions and typewriter to indicate an entity in KG.

WWW has two candidates, `World_Wide_Web` and `WWW_conference`, and hence is ambiguous. In such cases, coherence measure between the candidate entity and other unambiguously linked entity(ies) is used for disambiguation.

State-of-the-art EL systems measure coherence as similarity between embeddings of entities. The entity embeddings are trained based on the number of common edges in KG[2]. In our example, common edges are edges `World_Wide_Web` shares with `Andrei_Broder` and edges `WWW_conference` shares with `Andrei_Broder`. But `WWW_conference` has less number of edges (it is a sparsely-connected entity) compared to `World_Wide_Web`. This leads to poor performance[3] whereby *WWW* is erroneously linked to `World_Wide_Web` instead of linking to `WWW_conference`.

In this paper, we propose ELDEN, an EL system which increases nodes and edges of the KG by using information available on the web about entities and *pseudo entities*. Pseudo Entities are words and phrases that frequently occur in Wikipedia, and co-occur with mentions of KG entities in the web corpus. Thus ELDEN uses a web corpus to find pseudo entities and refines the co-occurrences with Pointwise Mutual Information (PMI) (Church and Hanks, 1989) measure. ELDEN then adds edges to the entity from pseudo entities. In Figure 1, pseudo entity *Program Committee* co-occurs with mentions of `Andrei_Broder` and `WWW_conference` in web corpus and has a positive PMI value with both. So ELDEN adds edges from *Program Committee* to `Andrei_Broder` and `WWW_conference`, densifying neighborhood of the entities. Coherence, now measured as similarity between entity embeddings where embeddings are trained on densified KG, leads to improved EL performance.

Density (number of KG edges) of candidate entity affects EL performance. In our analysis of density and number of entities having that density in the Wikipedia KG, we find that entities with 500 edges or less make up more than 90%. Thus, creating an EL system that performs well on densely as well as sparsely-connected entities is a challenging, yet unavoidable problem.

We make the following contributions:

- ELDEN presents a simple yet effective graph densification method which may be applied to improve EL involving any KG.

- By using pseudo entities and unambiguous mentions of entity in a corpus, we demonstrate how non-entity-linked corpus can be used to improve EL performance.

- We have made ELDEN's code and data publicly available[4].

2 Related Work

Entity linking: Most EL systems use coherence among entities (Cheng and Roth, 2013) to link mentions. We studied coherence measures and datasets used in six recent[5] EL systems (He et al., 2013; Huang et al., 2015; Sun et al., 2015; Yamada et al., 2016; Globerson et al., 2016; Barrena et al., 2016). We see that the two popular datasets used for evaluating EL (Chisholm and Hachey, 2015) on documents are CoNLL (Hoffart et al., 2011) and TAC2010 (Ji et al., 2010), here after TAC. The popular coherence measures used are (1) WLM, (2) Entity Embedding Similarity and (3) Jaccard Similarity (Chisholm and Hachey, 2015; Guo et al., 2013). WLM is widely acknowledged as most popular (Hoffart et al., 2012) with almost all the six approaches analyzed above using WLM or its variants. Entity embedding similarity (Yamada et al., 2016) is reported to give highest EL[6] performance and is the baseline of ELDEN.

Enhancing entity disambiguation: Among methods proposed in literature to enhance entity disambiguation utilizing KG (Bhattacharya and Getoor) uses additional relational information between database references; (Han and Zhao, 2010) uses semantic relatedness between entities in other KGs; and (Shen et al., 2018) uses paths consisting of defined relations between entities in the KG (IMDB and DBLP). All these methods utilize structured information, while our method shows how unstructured data (web corpus about the entity to be linked) can be effectively used for entity disambiguation.

Entity Embeddings: ELDEN presents a method to enhance embedding of entities and words in a

---

[2]Wikipedia Link based Measure (WLM) (Milne and Witten, 2008) used in Yamada et al.'s system is based on number of common edges in KG.

[3]This paper focuses on mention disambiguation. We assume mention and candidate entities are detected already.

[4]https://github.com/
priyaradhakrishnan0/ELDEN

[5] (Shen et al., 2015) presents a survey of EL systems.

[6]Named Entity Disambiguation (NED) and EL are synonymous terms in research (Hoffart et al., 2011)

common vector space. Word embedding methods like word2vec (Mikolov et al., 2013) and Glove (Pennington et al., 2014) have been extended to entities in EL (Yamada et al., 2016; Fang et al., 2016; Zwicklbauer et al., 2016; Huang et al., 2015). These methods use data about entity-entity co-occurrences to improve the entity embeddings. In ELDEN, we improve it with web corpus co-occurrence statistics. Ganea and Hofmann (2017) present a very interesting neural model for jointly learning entity embedding along with mentions and contexts.

KG densification with pseudo entities: KG densification using external corpus has been studied by Kotnis *et al.* (2015) and Hegde and Talukdar (2015). Densifying edge graph is also studied as 'link prediction' in literature (Martínez et al., 2016). Kotnis *et al.* augment paths between KG nodes using 'bridging entities' which are noun phrases mined from an external corpus. ELDEN has a similar approach as it proposes densifying the KG edges of entities by adding edges from pseudo entities. However, densification is used for relation inference in the former methods whereas it is used for entity coherence measurement in ELDEN.

Word co-occurrence measures: Chaudhari *et al.* (2011) survey several co-occurrence measures for word association including PMI, Jaccard (Dice, 1945) and Co-occurrence Significance Ratio (CSR). Damani (2013) proves that considering corpus level significant co-occurrences, PMI is better than others. Budiu *et al.* (2007) compare Latent Semantic Analysis (LSA), PMI and Generalized Latent Semantic Analysis (GLSA) and conclude that for large corpora like web corpus, PMI works best on word similarity tests. Hence, we chose PMI to refine co-occurring mentions of entities in web corpus.

3 Definitions and Problem Formulation

In this section, we present a few definitions and formulate the EL problem.

Knowledge Graph (KG): A Knowledge Graph is defined as $G = (E, F)$ with entities E as nodes and F as edges. In allegiance to EL literature and baselines (Milne and Witten, 2008; Globerson et al., 2016), we use the Wikipedia hyperlink graph as the KG in this paper, where nodes correspond to Wikipedia articles and edges are incoming links from one Wikipedia article to another. ELDEN ultimately uses a densified version of this

Wikipedia KG, as described in Section 4.

Sparsely connected entities: Following Hoffart *et al.* (2012), we define an entity to be a sparsely connected entity, if the number of edges incident on the entity node in the KG is less than threshold η[7]. Otherwise, the entity is called a densely-connected entity.

Entity Linking (EL): *Given a set of mentions $M_D = \{m_1, ..., m_n\}$ in a document D, and a knowledge graph $G = (E, F)$, the problem of entity linking is to find the assignment $\Lambda : M_D \to E_D$, where E_D is the set of entities linked to mentions in document D such that $E_D \subseteq E$.*

For mention $m_i \in M_D$, let the set of possible entities it can link to (candidate entities) be $\mathbb{C}_i$. Then, the solution to the EL problem is an assignment Λ where,

$$\Lambda(m_i) = \arg\max_{e \in \mathbb{C}_i} \left[\phi(m_i, e) + \beta \cdot \psi(e, E_D) \right]$$

(1)

Here, $\phi(m_i, e) \in [0, 1]$ measures the contextual compatibility of mention m_i and entity e. $\phi(m_i, e)$ is obtained by combining prior probability and context similarity. $\psi(e, E_D)$ measures the coherence of e with entities in E_D. β is a variable controlling inclusion of ψ in the assignment Λ.

Problem Formulation : (Yamada et al., 2016) is a recently proposed state-of-the-art EL system. We consider it as a representative EL system and use it as the main baseline for the experiments in this paper. In this section, we briefly describe Yamada *et al.* (2016)'s two-step approach that solves the EL problem presented above.

Step 1: A mention $m_i \in M_D$ is defined to be *unambiguous* if $\exists e \in \mathbb{C}_i$ such that $\phi(m_i, e) \geq \gamma$. Let $M_D^{(u)} \subseteq M_D$ be the set of such unambiguous mentions in document D. For all unambiguous mentions $m \in M_D^{(u)}$, Yamada *et al.* freeze the assignment by solving Equation 1 after setting $\beta = 0$. In other words, $\psi(e, E_D)$ is not used while assigning entities to unambiguous mentions. Assigning entities first to unambiguous mentions has also been found to be helpful in prior research (Milne and Witten, 2008; Guo and Barbosa, 2014). Let A_D be the set of entities linked to in this step. In Figure 1, mention *Andrei Broder* is unambiguous[8].

---

[7]Like (Hoffart et al., 2012) we set $\eta = 500$ for the experiments in this paper.

[8]Between mention *Andrei Broder* and entity `Andrei_Broder`, string similarity and prior probability are 1.0. In the experiments we use a γ value of 0.95.

Notation	Definition
v_e	entity embedding of entity e with dimension $1 * d$
E	set of all titles in Wikipedia
S	set of all pseudo entities considered in ELDEN
E_+	set of all entities considered in ELDEN
$\mathbf{V}$	word vectors of E_+ where $\mathbf{V} \in \mathbb{R}^{k*d}$
ψ_{ELDEN}	Embedding similarity measured using $\mathbf{V}$ trained on G_{dense}
ψ_{Yamada}	Embedding similarity measured using $\mathbf{V}$ trained on input KG G

Table 1: Notation used in KG densification and learning entity embeddings (Please see Sec 4 for more details).

Step 2: In this step, Yamada *et al.* links all ambiguous mentions by solving Equation 2.

$$\Lambda(m_i) =$$
$$\arg\max_{e \in \mathbb{C}_i} \left(\phi(m_i, e) + \frac{1}{|A_D|} \sum_{e_j \in A_D} v_e \cdot v_{e_j} \right)$$
$$\forall m_i \in M_D \setminus M_D^{(u)} \quad (2)$$

where $v_e, v_{e_j} \in \mathbb{R}^d$ are d-dimensional embeddings of entities e and e_j respectively. Please note that the equation above is a reformulation of Equation 1 with $\beta = 1$ and $\psi(e, E_D) = \frac{1}{|A_D|} \sum_{e_j \in A_D} v_e \cdot v_{e_j}$, where A_D is derived from E_D as described in Step 1. As coherence $\psi(e, E_D)$ is applied only in disambiguation of ambiguous mentions, we apply densification to only selective nodes of KG.

Embeddings of entities are generated using word2vec model and trained using WLM (Details in Section 5). Given a graph $G = (E, F)$, the WLM coherence measure $\psi_{wlm}(e_i, e_j)$ between two entities e_i and e_j is defined in Equation 3 where C_e is the set of entities with edge to entity e.

$$\psi_{\text{wlm}}(e_i, e_j) = 1 -$$
$$\frac{log(max(|C_{e_i}|, |C_{e_j}|)) - log(|C_{e_i} \cap C_{e_j})|)}{log(|E|) - log(min(|C_{e_i}|, |C_{e_j}|))}$$
$$(3)$$

4 Our Approach: ELDEN

In this section, we present ELDEN, our proposed approach. ELDEN extends (Yamada et al., 2016), with one important difference: rather than working with the input KG directly, ELDEN works with the densified KG, created by selective densification of the KG with co-occurrence statistics extracted from a large corpus. Even though this

is a simple change, this results in improved EL performance. The method performs well even for sparsely connected entities.

Overview : Overview of the ELDEN system is shown in Figure 2. ELDEN starts off with densification of the input KG, using statistics from web corpus. Embeddings of entities are then learned utilizing the densified KG in the next step. Embedding similarity estimated using the learned entity embeddings is used in calculating coherence measure in subsequent EL. Notation used is summarized in Table 1.

*(i)***KG Densification** Figure 2 depicts densification of KG in 'Input KG' and 'Densified KG'. It shows two Wikipedia titles `Andrei_Broder` and `WWW_conference` from our running example (Figure 1). There are no edges common between `Andrei_Broder` and `WWW_conference`. In a web corpus, mentions of `Andrei_Broder` and `WWW_conference` co-occur with *Program committee* and it has a positive PMI value with both the entities. So ELDEN adds an edge from *Program committee* to both the entities. Here *Program committee* is a pseudo entity. Thus, ELDEN densifies the KG by adding edges from pseudo entities when the mentions of Wikipedia entity and pseudo entity co-occur in a web corpus and the pseudo entity has a positive PMI value with given entity.

Taking a closer look, KG densification process starts from 'input KG' which is Wikipedia hyperlink graph $G = (E, F)$, where the nodes are Wikipedia titles (E) and edges are hyperlinks (F). ELDEN processes Wikipedia text corpus and identifies phrases (unigrams and bi-grams) that occur frequently, i.e. more than 10 times in it. We denote these phrases as *pseudo entities* (S) and add them as nodes to the KG. Let $E_+ = E \cup S$ be the resulting set of nodes.

ELDEN then adds edges connecting entities in E_+ to entities in E. This is done by processing a web text corpus looking for mentions of entities in E_+, and linking the mentions to entities in KG $G' = (E_+, F)$. ELDEN uses Equation 1 with $\beta = 0$ for this entity linking, i.e. only mention-entity similarity $\phi(m, e)$ is used during this linking[9]. Based on this entity linked corpus, a co-occurrence matrix M of size $|E_+| \times |E_+|$ is constructed. Each cell $M_{i,j}$ is set to the PMI between

---
[9]Since prior probabilities of pseudo entities are not available, only mention-entity similarity component of $\phi(m, s)$ is used while linking a mention m to a pseudo entity $s \in S$.
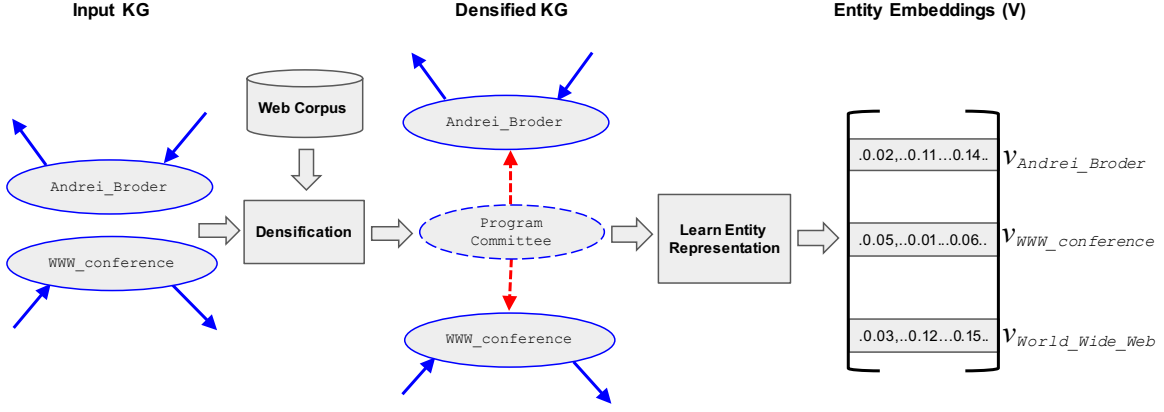
Embedding similarity $= \boldsymbol{\psi}_{\text{ELDEN}} = \text{Cosine}\left(v_{Entity1}, v_{Entity2}\right)$

$\text{Cosine}\left(v_{Andrei\_Broder}, v_{WWW\_conference}\right) > \text{Cosine}\left(v_{Andrei\_Broder}, v_{World\_Wide\_Web}\right)$

Figure 2: ELDEN consists of KG densification, training entity embeddings on densified KG and building EL system that uses similarity between the trained embeddings as coherence measure. ELDEN's difference from baseline method is that while baseline method uses input KG for training v_e, ELDEN uses densified KG for training v_e. Hence, the improved performance of ELDEN is solely attributed to densification.

Objective function component	Feature Group	Feature	Explanation
Contextual compatibility $\phi(m_i, e_i)$	Base	Entity Prior	Fraction of edges that links to the entity
		Prior Probability	Of all possible pages a mention can link to, the probability that it links to a given page
		Max Prior Probability	Maximum value of prior probabilities for a mention
		Max Prior Probability in document	Maximum value of prior probabilities for a document
		Num of candidates	Candidate count of the mention
	String Similarity	Exact match	Whether mention is an exact match of candidate
		Partial match	Whether mention is part of candidate
Coherence $\psi(e_i, e_j)$	Coherence measures	ψ_{wlm}	WLM (Please see Equation 3)
		ψ_{Yamada}	Similarity between entity embeddings that are learned using ψ_{wlm}
		ψ_{dense}	ψ_{wlm} calculated on densified KB
		ψ_{ELDEN}	Similarity between entity embeddings that are trained using ψ_{dense}
		$\psi_{\text{ELDEN++}}$	Combination of ψ_{wlm}, ψ_{Yamada}, ψ_{dense}, and ψ_{ELDEN}. ELDEN uses this coherence feature.

Table 2: Features used by various EL systems. Context compatibility $\phi(m_i, e_i)$ and coherence $\psi(e_i, e_j)$ of Equation 1 are parameterized using features as shown above. Context compatibility features are used by all systems. Coherence features used by Yamada16 are ψ_{wlm} and ψ_{Yamada}. ELDEN uses $\psi_{\text{ELDEN++}}$ as coherence feature.

the entities e and e'.

$$M_{e,e'} = \text{PMI}(e, e') = \log \frac{f(e, e') \times N}{f(e) \times f(e')}$$

where $f(e)$ is the frequency of entity e in web corpus, $f(e, e')$ is the sentence-constrained pair frequency of the entity pair (e, e') in web corpus, and $N = \sum_{e,e' \in E_+} f(e, e')$. Please note that PMI, and there by M, are symmetric. The expanded set of edges, F_+, is now defined as

$$F_+ = F \cup \{(e, e'), (e', e) \mid e' \in E_+, e \in E, M_{e,e'} > 0\}$$

In other words, we augment the set of initial edges F with additional edges connecting entities in E_+ with entities in E such that PMI between the entities is positive.

ELDEN now constructs the KG $G_{\text{dense}} = (E_+, F_+)$, which is a densified version of the input KG $G = (E, F)$. ELDEN uses this densified KG G_{dense} for subsequent processing and entity linking.

*(ii)*Learning Embeddings of Densified KG Entities ELDEN derives entity embeddings using the same setup, corpus and Word2vec skip-gram with negative sampling model as in Yamada *et al.*, However, instead of training embeddings over the input KG, ELDEN trains embeddings of entities in the densified KG G_{dense}. Let $\mathbf{V}$ be the word2vec matrix containing embeddings of entities in E_+ where $\mathbf{V} \in \mathbb{R}^{k*d}$. v_e is the embedding of entity e in E_+ with dimension $1 * d$.

In word2vec model, entities in context are used

to predict the target entity. ELDEN maximizes the objective function (Goldberg and Levy, 2014) of word2vec skip-gram model with negative sampling, $L = \sum_{(t,c) \in \mathbb{P}} L_{t,c}$ where

$$L_{t,c} = log\ \theta(v_c \cdot v_t) + \sum_{n \in \mathbb{N}_{(t,c)}} log\ \theta(-v_n \cdot v_t)$$

Here v_t and v_c are the entity embeddings of target entity t and context entity c. $\mathbb{P}$ is the set of target-context entity pairs considered by the model. $\mathbb{N}_{(t,c)}$ is a set of randomly sampled entities used as negative samples with pair (t,c). This objective is maximized with respect to variables v_t's and v_c's, where $\theta(x) = \frac{1}{1+e^{-x}}$.

$\mathbb{P}$ and $\mathbb{N}$ are derived using G_{dense}. t and c are entities in E_+ such that c shares a common edge with t. v_n is randomly sampled from $\mathbf{V}$, for entities that do not share a common edge with t. Entity embedding similarity measured using $\mathbf{V}$ trained this way on G_{dense} is ψ_{ELDEN}. Embedding similarity is measured as cosine distance between v_es. Embeddings of S are trained using positive and negative word contexts derived using context length.

***(iii)* Bringing it All Together: ELDEN**

ELDEN is a supervised EL system which uses two sets of features: (1) contextual compatibility $\phi(m,e)$; and (2) coherence $\psi(e_i, e_j)$. These features are summarized in Table 2. Similarity between entity embeddings is measured as cosine similarity between v_es.

5 Experiments

In this section, we evaluate the following:

- Is ELDEN's corpus co-occurrence statistics-based densification helpful in disambiguating entities better? (Sec. 6.1)

- Where does ELDEN's selective densification of KG nodes link entities better? (Sec. 6.3)

Setup : ELDEN is implemented using Random Forest ensemble [10] (Breiman, 1998). Parameter values were set using CoNLL development set. Feature limit of 3 with number of estimators as 100 yielded best performance.

Knowledge Graph: Wikipedia Following prior EL literature, we use Wikipedia hypergraph as our KG (Milne and Witten, 2008; Globerson et al., 2016). This KG is enhanced with pseudo entities as explained in Section 4. We process

---

Specifics	Value
Number of titles after cleaning ($|E|$)	4.6 M
Number of pseudo entities ($|S|$)	1.3 M
Total number of entities ($|E_+|$)	5.9 M
Number of epochs	3
Learning rate	0.25 linearly reducing to 0.01
Number of negative samples ($|\mathbb{N}|$)	5
Context window size	3
Dimensions of embedding (d)	100

Table 3: Parameters used in Wikipedia processing and training KG.

the Wikipedia corpus following the same procedure as in Yamada *et al.* (2016). We cleaned [11] Wikipedia dump dated Nov 2015. We then parsed the Wikipedia article text to identify pseudo entities. More details on KG and parameters used for training embeddings are in Table 3. Training took 4 days on gpu with 2 cores.

Preprocessing: Web corpus and Densified KG For our experiments, we created a web corpus by querying Google[12]. Candidate entities of all mentions in the dataset are queried in Google and top ten search results are considered for unigram and bigram frequencies. This corpus occupied 6.8GB for candidate entities of TAC and CoNLL dataset mentions (54336 entities). Even for sparsely connected entities, an average corpus size of 670 lines or more[13] was collected. We note that though some of the entities mentioned in this dataset are ten or more years old, we are able to collect, on an average more than 670 lines of web content. Thus corpus proves to be a good source of additional links for densification, for both common and rare entities. As Taneva and Weikum (2013) also note, it is not hard to find content about sparsely connected entities on the web.

The web corpus is analyzed for mentions and pseudo entities. Co-occurrence matrix M is created[14] for mention and pseudo entities occurring within window of size 10 for PMI calculation[15]. Edges are added from pseudo entities with positive PMI to mention of given entity. In experiments we add edges from top 10 pseudo entities ordered by

---

PMI values[16].

Evaluation Dataset: In line with prior work on EL, we test the performance of ELDEN on CoNLL and TAC datasets. As this paper focuses on entity disambiguation, we tested ELDEN against datasets and baseline methods for disambiguation. We note that the entity disambiguation evaluation part of other recent datasets like ERD 2014 and TAC 2015 is exactly same as the TAC 2010 evaluation (Ellis et al., 2014) [17].

Training: ELDEN's parameters were tuned using training (development) sets of CoNLL and TAC datasets. CoNLL and TAC datasets consist of documents where mentions are marked and entity to which the mention links to, is specified. We use only mentions that link to a valid Wikipedia title (non NIL entities) and report performance on test set. Some aspects of these datasets relevant to our experiments are provided below.

CoNLL: In CoNLL test set (5267 mentions), we report Precision of topmost candidate entity, aggregated over all mentions (P-micro) and aggregated over all documents (P-macro), i.e., if tp, fp and p are the individual true positives, false positives and precision for each document in a dataset of δ documents, then

$$P_{\text{micro}} = \frac{\sum_{i=1}^{\delta} \text{tp}_i}{\sum_{i=1}^{\delta} \text{tp}_i + \sum_{i=1}^{\delta} \text{fp}_i} \text{ and } P_{\text{macro}} = \frac{\sum_{i=1}^{\delta} \text{p}_i}{\delta}.$$

For CoNLL candidate entities, we use (Pershina et al., 2015) dataset[18].

TAC: In TAC dataset, we report P-micro of top-ranked candidate entity on 1,020 mentions. P-macro is not applicable to TAC as most documents have only one mention as query mention (or 'mention to be linked'). For TAC candidate entities, we index the Wikipedia word tokens and titles using solr[19]. We index terms in (1) title of the entity, (2) title of another entity redirecting to the entity, and (3) names of anchors that point to the entity, in line with baselines. We are making this TAC candidate set publicly available.

Baseline: Yamada16 Our baseline is the Yamada *et al.* system explained in Section 3. Entity embedding distance measured using v_e trained on the

---

[16]This is a tunable parameter.

[17]These recent datasets consist of other evaluations, e.g., mention detection, multilinguality etc. which is beyond the scope of the paper and hence we didnt focus on them in the paper.

[18]https://github.com/masha-p/PPRforNED

[19]http://lucene.apache.org/solr/

Method	CONLL (P-micro)	CONLL (P-macro)	TAC (P-micro)
(Hoffart et al., 2011)	82.5	81.7	–
(He et al., 2013)	85.6	84.0	81.0
(Ling et al., 2015)	67.5	–	86.8
(Barrena et al., 2016)	88.32	–	–
(Chisholm and Hachey, 2015)	88.7	–	80.7
(Pershina et al., 2015)	91.8	89.9	–
(Globerson et al., 2016)	91.7	–	87.2
(Yamada et al., 2016)	**93.1**	92.6	85.2
ELDEN	93.0	**93.7**	**89.6**

Table 4: Performance comparison with other recent EL approaches. ELDEN matches best results in CoNLL and outperforms the state-of-the-art in TAC dataset. (Please see Section 6.1 for details and $\psi_{\text{ELDEN++}}$ row of Table 5 for ELDEN results.)

input KG G is ψ_{Yamada}.

6 Results

6.1 Does ELDEN's selective densification help in disambiguation in EL?

In Table 4, we compare ELDEN's EL performance with results of other recently proposed state-of-the-art EL methods that use coherence models. We see that ELDEN results matches best results on CoNLL and outperforms state-of-the-art in TAC dataset. In the table, the last four rows uses the Pershina *et al.* (2015) candidate set and hence, we provide a comparison of their disambiguation performance. Improved results of ELDEN over baseline is attributed to the improved disambiguation due to KG densification.

6.2 Why does ELDEN's selective densification work?

We conduct ablation analysis using various feature and feature combinations and present performance of ELDEN and baseline in Table 5. Starting with base features, we add various features to ELDEN incrementally and report their impact on performance. The results when using base feature group alone, and base and string similarity groups together (ϕ) are presented in first and second rows for each dataset. We compare ψ_{ELDEN} to three coherence measures: ψ_{wlm}, ψ_{Yamada} and ψ_{dense}, details of which are provided in Table 2. The performance improvement from each of the four coherence measures are in the next four rows. Performance of ELDEN from using all four coherence features is given in $\psi_{\text{ELDEN++}}$ row.

On CoNLL dataset, ψ_{dense} combined with ϕ,

Dataset	Features	P-micro	P-macro
CoNLL	Base	87.0	88.3
	ϕ	90.0	91.2
	$\phi + \psi_{\text{wlm}}$	91.0*	91.8
	$\phi + \psi_{\text{Yamada}}$	90.0*	91.1
	$\phi + \psi_{\text{dense}}$	92.0*	93.0
	$\phi + \psi_{\text{ELDEN}}$	91.0*	91.2
	$\phi + \psi_{\text{wlm}} + \psi_{\text{Yamada}}$	91.0	91.8
	$\phi + \psi_{\text{dense}} + \psi_{\text{ELDEN}}$	91.0	92.6
	$\phi + \psi_{\text{ELDEN}++}$	**93.0**	**93.7**
TAC	Base	78.2	
	ϕ	80.2	
	$\phi + \psi_{\text{wlm}}$	82.5*	
	$\phi + \psi_{\text{Yamada}}$	83.1*	
	$\phi + \psi_{\text{dense}}$	85.5*	
	$\phi + \psi_{\text{ELDEN}}$	87.3*	
	$\phi + \psi_{\text{wlm}} + \psi_{\text{Yamada}}$	84.4	
	$\phi + \psi_{\text{dense}} + \psi_{\text{ELDEN}}$	88.7	
	$\phi + \psi_{\text{ELDEN}++}$	**89.6**	

Table 5: Ablation analysis involving various coherence measures (see Table 2 for definitions of these measures). Statistically significant improvements over ϕ are marked with an asterisk. ELDEN's coherence measure, $\psi_{\text{ELDEN}++}$, achieves the best overall performance. P-macro is not applicable to TAC as most documents have only one mention marked as query. (Please see Section 6.2).

gave an improvement of 2.0 and 1.9 (P-micro and P-macro) over Yamada16 results. We note that Yamada16 results are from our re-implementation of (Yamada et al., 2016) system [20] and we are able to almost reproduce the baseline results. We also present the results combining baseline's ψ_{Yamada} and ψ_{wlm} versus ELDEN's ψ_{ELDEN} and ψ_{dense} in next two rows. We find the ELDEN's KG densification features perform better than baselines.

On TAC dataset also, combined with ϕ, ψ_{dense} is found to do better than ψ_{wlm} and ψ_{ELDEN} gives a significant P-micro improvement of 4.2 over ψ_{Yamada}. The $\psi_{\text{ELDEN}++}$ P-micro in TAC dataset is statistically significant[21]. In short, we find the KG densification features, ψ_{dense} and ψ_{ELDEN}, as the features causing better performance of ELDEN on both datasets.

6.3 Where does ELDEN's selective densification work better?

While most EL systems give higher precision on CoNLL dataset than TAC dataset, ELDEN performs with high precision on TAC dataset too.

---

[20]We have re-implemented the Yamada *et al* system using hyper-parameters specified in the paper and these are our best-effort results.

[21]We performed two tailed t-test, with 2-tail 95% value of 1.96.

Dataset	% sparsely connected Entities	
	Train	**Test**
TAC	78.8	63.6
CoNLL	48.4	48.2

Table 6: Percentage of sparsely connected entities in evaluation datasets. TAC has higher composition of sparsely connected entities than CoNLL. Hence, ELDEN results are better in TAC over CoNLL (Please see Table 4).

This is explained by analyzing distribution of densely-connected and sparsely connected entities in TAC and CoNLL datasets as presented in Table 6. We see that CoNLL test set has almost half as densely-connected and half as sparsely connected entities, whereas in TAC test set, 63.6% are sparsely connected entities. This higher constitution of sparsely connected entities in TAC, explains ELDEN's better results in TAC relative to CoNLL dataset. As the number of sparsely connected entities is more than the number of densely-connected entities in most KGs (Reinanda et al., 2016), our method is expected to be of significance for most KGs.

6.4 What type of EL errors are best fixed with ELDEN's selective densification ?

We analyzed errors fixed by ELDEN on TAC dataset. We categorize the errors into four classes in line with error classes of Ling *et al.* (2015). We
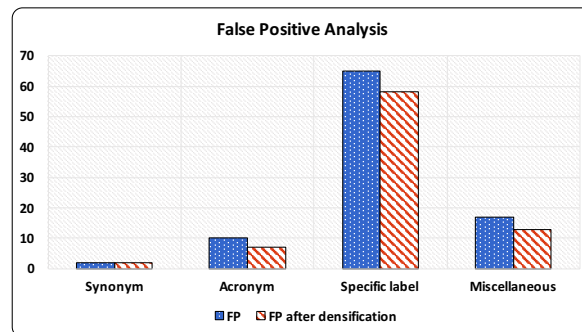

Figure 3: False positives of ELDEN before and after KG densification. Errors reduce with use of ψ_{dense} and ψ_{ELDEN} measures. (Please see Sec 6.4)

manually analyzed 240 wrong predictions of Yamada16 and compared it with that of ELDEN, and the results are presented in Figure 3. We found errors to reduce with use of KG densification features and most of the errors eliminated were in "Specific label" class. Errors in this class called for better modeling of mention's context and link-based similarity (Ling et al., 2015).(More details of this analysis in the supplementary document.)

7 Conclusion

We started this study by analyzing the performance of a state-of-the-art Entity Linking (EL) system and found that its performance was low when linking entities sparsely-connected in the KG. We saw that this can be addressed by densifying the KG with respect to the given entity. We proposed ELDEN, which densifies edge graph of entities using pseudo entities and mentions of entities in a large web corpus. Through our experiments, we find that ELDEN outperforms state-of-the-art baseline on benchmark datasets.

We believe that ELDENs combination of KG densification and entity embeddings is novel. Poor performance of EL systems on sparsely connected entities has been recognized as one of the open challenges by prior research. ELDEN performs well on sparsely connected entities too, as a validation of our method of combining KG densification followed by embedding. Our approach may be applied to any KG as the densification is performed with the help of *unstructured data*, and not any specific KG. We hope the simple graph densification method utilized in ELDEN will be of much interest to the research community.

Pseudo entities can be looked at as entity candidates for KG expansion, as also noted by Farid *et al.* (2016). In future, we plan to enhance ELDEN using EL of pseudo entities to estimate *entity prior* of entities not present in KG. We also plan to explore entity embeddings obtained using other graph densifying methods.

8 Acknowledgments

References

Ander Barrena, Aitor Soroa, and Eneko Agirre. 2016. Alleviating poor context with background knowledge for named entity disambiguation. In *ACL (1)*. The Association for Computer Linguistics.

Indrajit Bhattacharya and Lise Getoor. ???? Online collective entity resolution.

Leo Breiman. 1998. Arcing classifier (with discussion and a rejoinder by the author). *Ann. Statist.* 26(3):801–849. https://doi.org/10.1214/aos/1024691079.

Raluca Budiu, Christiaan Royer, and Peter Pirolli. 2007. Modeling information scent: A comparison of lsa, pmi and glsa similarity measures on common tests and corpora. In *Large Scale Semantic Access to Content (Text, Image, Video, and Sound)*. Paris, France, France, RIAO '07, pages 314–332. http://dl.acm.org/citation.cfm?id=1931390.1931422.

Dipak Chaudhari, Om P. Damani, and Srivatsan Laxman. 2011. Lexical co-occurrence, statistical significance, and word association. *CoRR* abs/1008.5287.

Xiao Cheng and Dan Roth. 2013. Relational inference for wikification. In *EMNLP*. ACL, pages 1787–1796.

Andrew Chisholm and Ben Hachey. 2015. Entity disambiguation with web links. *Transactions of the Association for Computational Linguistics* 3:145–156. https://tacl2013.cs.columbia.edu/ojs/index.php/tacl/article/view/494.

Kenneth Ward Church and Patrick Hanks. 1989. Word association norms, mutual information, and lexicography. ACL '89, pages 76–83. https://doi.org/10.3115/981623.981633.

Om P. Damani. 2013. Improving pointwise mutual information (PMI) by incorporating significant co-occurrence. *CoRR* abs/1307.0596.

Lee R. Dice. 1945. Measures of the amount of ecologic association between species. *Ecology* 26(3):297–302. https://doi.org/10.2307/1932409.

Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. ACM, New York, NY, USA, KDD '14, pages 601–610. https://doi.org/10.1145/2623330.2623623.

Joe Ellis, Jeremy Getman, and Stephanie Strassel. 2014. Overview of linguistic resource for the tac kbp 2014 evaluations: Planning, execution, and results. Linguistic Data Consortium, University of Pennsylvania.

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

Mina Farid, Ihab F. Ilyas, Steven Euijong Whang, and Cong Yu. 2016. Lonlies: Estimating property values for long tail entities. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, USA, SIGIR '16, pages 1125–1128. https://doi.org/10.1145/2911451.2911466.

Octavian-Eugen Ganea and Thomas Hofmann. 2017. Deep joint entity disambiguation with local neural attention. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 2609–2619. http://www.aclweb.org/anthology/D17-1276.

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

Yoav Goldberg and Omer Levy. 2014. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *CoRR* abs/1402.3722. http://arxiv.org/abs/1402.3722.

Stephen Guo, Ming-Wei Chang, and Emre Kiciman. 2013. To link or not to link? a study on end-to-end tweet entity linking. In *NAACL-HLT 2013*. http://research.microsoft.com/apps/pubs/default.aspx?id=183909.

Zhaochen Guo and Denilson Barbosa. 2014. Robust entity linking via random walks. ACM, New York, NY, USA, CIKM '14, pages 499–508. https://doi.org/10.1145/2661829.2661887.

Xianpei Han and Jun Zhao. 2010. Structural semantic relatedness: A knowledge-based method to named entity disambiguation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, ACL '10, pages 50–59. http://dl.acm.org/citation.cfm?id=1858681.1858687.

Zhengyan He, Shujie Liu, Yang Song, Mu Li, Ming Zhou, and Houfeng Wang. 2013. Efficient collective entity linking with stacking. EMNLP. http://research.microsoft.com/apps/pubs/default.aspx?id=202249.

Manjunath Hegde and Partha P. Talukdar. 2015. An entity-centric approach for overcoming knowledge graph sparsity. Association for Computational Linguistics, Lisbon, Portugal, EMNLP '15, pages 530–535. http://aclweb.org/anthology/D15-1061.

Johannes Hoffart, Stephan Seufert, Dat Ba Nguyen, Martin Theobald, and Gerhard Weikum. 2012. Kore: Keyphrase overlap relatedness for entity disambiguation. ACM, New York, NY, USA, CIKM '12, pages 545–554. https://doi.org/10.1145/2396761.2396832.

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. Association for Computational Linguistics, Stroudsburg, PA, USA, EMNLP '11, pages 782–792. http://dl.acm.org/citation.cfm?id=2145432.2145521.

Hongzhao Huang, Larry Heck, and Heng Ji. 2015. Leveraging deep neural networks and knowledge graphs for entity disambiguation. *CoRR* abs/1504.07678. http://arxiv.org/abs/1504.07678.

Heng Ji, Ralph Grishman, Hoa Trang Dang, Kira Griffitt, and Joe Ellis. 2010. Overview of the tac 2010 knowledge base population track. In *In Third Text Analysis Conference (TAC)*.

Bhushan Kotnis, Pradeep Bansal, and Partha P. Talukdar. 2015. Knowledge base inference using bridging entities. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*. pages 2038–2043. http://aclweb.org/anthology/D/D15/D15-1241.pdf.

Xiao Ling, Sameer Singh, and Dan Weld. 2015. Design challenges for entity linking. *Transactions of the Association for Computational Linguistics (TACL)* 3.

Víctor Martínez, Fernando Berzal, and Juan-Carlos Cubero. 2016. A survey of link prediction in complex networks. *ACM Comput. Surv.* 49(4):69:1–69:33. https://doi.org/10.1145/3012704.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS 2013*. pages 3111–3119.

David Milne and Ian H. Witten. 2008. Learning to Link with Wikipedia. ACM, CIKM'08, pages 509–518.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. pages 1532–1543. http://www.aclweb.org/anthology/D14-1162.

Maria Pershina, Yifan He, and Ralph Grishman. 2015. *Personalized page rank for named entity disambiguation*, Association for Computational Linguistics (ACL), pages 238–243. NAACL HLT 2015.

Ridho Reinanda, Edgar Meij, and Maarten de Rijke. 2016. Document filtering for long-tail entities. ACM, CIKM'16.

W. Shen, J. Han, J. Wang, X. Yuan, and Z. Yang. 2018. Shine+: A general framework for domain-specific entity linking with heterogeneous information networks. *IEEE Transactions on Knowledge and Data Engineering* 30(2):353–366. https://doi.org/10.1109/TKDE.2017.2730862.

W. Shen, J. Wang, and J. Han. 2015. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering* 27(2):443–460. https://doi.org/10.1109/TKDE.2014.2327028.

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 Qiang Yang and Michael Wooldridge, editors, *IJCAI*. AAAI Press, pages 1333–1339. http://dblp.uni-trier.de/db/conf/ijcai/ijcai2015.html#SunLTYJW15.

Bilyana Taneva and Gerhard Weikum. 2013. Gem-based entity-knowledge maintenance. ACM, New York, NY, USA, CIKM '13, pages 149–158. https://doi.org/10.1145/2505515.2505715.

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, pages 250–259. http://aclweb.org/anthology/K/K16/K16-1025.pdf.

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*. ACM, New York, NY, USA, SIGIR '16, pages 425–434. https://doi.org/10.1145/2911451.2911535.