Low-Rank Subspaces for Unsupervised Entity Linking

Akhil Arora
EPFL
akhil.arora@epfl.ch

Alberto García-Durán*
Atinary Technologies
agaduran@gmail.com

Robert West
EPFL
robert.west@epfl.ch

Abstract

Entity linking is an important problem with many applications. Most previous solutions were designed for settings where annotated training data is available, which is, however, not the case in numerous domains. We propose a light-weight and scalable entity linking method, EIGENTHEMES, that relies solely on the availability of entity names and a referent knowledge base. EIGENTHEMES exploits the fact that the entities that are truly mentioned in a document (the "gold entities") tend to form a semantically dense subset of the set of all candidate entities in the document. Geometrically speaking, when representing entities as vectors via some given embedding, the gold entities tend to lie in a low-rank subspace of the full embedding space. EIGENTHEMES identifies this subspace using the singular value decomposition and scores candidate entities according to their proximity to the subspace. On the empirical front, we introduce multiple strong baselines that compare favorably to (and sometimes even outperform) the existing state of the art. Extensive experiments on benchmark datasets from a variety of real-world domains showcase the effectiveness of our approach.

1 Introduction

Entity linking (EL) is the task of grounding mentions to a reference knowledge base (also referred to as knowledge graph). With a plethora of applications, including but not limited to information extraction (Hoffart et al., 2011) and automatic knowledge base construction (Gao et al., 2018), EL is one of the most actively researched topics in natural language processing. Despite the recent proliferation of EL methods, recent works (Tsai and Roth, 2016; Upadhyay et al., 2018) have pointed out that the performance of existing techniques largely relies on the existence of large corpora of annotated data.

Unsupervised entity linking. In contrast, truly unsupervised entity linkers should be able to operate in the *absolute absence of annotated data*, with access to only a reference knowledge graph (KG) and a list of entity names, or aliases. Furthermore, neither the candidate generator nor the disambiguation technique (the two key modules of any typical entity linker) can make use of annotated data. Therefore, "unsupervised" disambiguation techniques that leverage labeled data to generate candidate entities (Pan et al., 2015) are not applicable in our setting. Only very recently have researchers (Le and Titov, 2019; Logeswaran et al., 2019) started to focus on EL systems that can operate in the absolute absence of annotated data. The motivation is well founded: there are some domains that link to their own specialized KGs, for which labeled data is not readily available, expensive to obtain, or scarce in the best case. This can be exemplified with domains such as law, science, or medicine. Moreover, companies may have their proprietary KGs where some entities are only meaningful with respect to the company. In all these cases, available labeled data cannot generalize to the corresponding specialized KGs.

Challenges. The majority of the existing methods (Milne and Witten, 2008; Ganea and Hofmann, 2017; Yamada et al., 2017; Gupta et al., 2017; Le and Titov, 2018) in the EL literature are ill-suited for unsupervised EL as they rely heavily on annotated data. These methods typically leverage annotated data to (1) generate candidate entities, (2) use features (*e.g.* the prior probability $P(e|m)$ of an entity e given a mention m) derived from these annotations, (3) learn aligned word and entity embeddings enabling computation of similarity between an entity and the mention context, and (4) use a set of mentions and their corresponding gold entities to train supervised models. Thus, it is non-trivial and unclear how to adapt these methods to work in the absence of annotated data.

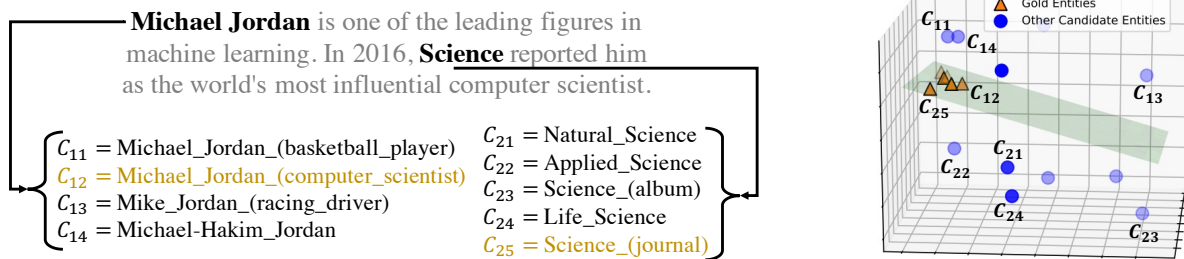*Research done while at EPFL.

Figure 1: (Left) Excerpt from a Wikipedia article about *Michael I. Jordan* with two mentions (**in bold**) and corresponding candidate entities generated using surface name matching. (Right) Embedding space for all the candidate entities within a document. The gold entities (triangles) form a cluster in the proximity of the subspace (green plane) identified by EIGENTHEMES, while other candidates (circles) are distant from the subspace.

By the same token, one may not expect a large validation set of annotated data wherein to perform extensive hyperparameter tuning. Thus, novel approaches to EL without annotated data should either possess a small number of hyperparameters or be robust to hyperparameter tuning.

EIGENTHEMES. We propose EIGENTHEMES, a scalable approach for performing collective disambiguation in the absence of annotated data. Given a document with a number of mentions, each mention with a set of candidate entities that it can be linked to, and vector representations for all these candidate entities, EIGENTHEMES builds a matrix of the vector representations of all candidate entities within the document, and uses singular value decomposition to learn a subspace spanned by a number k of components referred to as "eigenthemes", where k is the only hyper-parameter specific to EIGENTHEMES. Note that each principal component of the learned subspace captures the topical relatedness among entities across a different "latent" facet, and thus, by keeping several principal components, EIGENTHEMES is equipped to deal with multi-topic text.

By design, EIGENTHEMES is suitable in settings where the gold entities within a document are topically related, in particular much more so than other subsets of candidate entities—a realistic assumption made by almost all existing works in the EL literature (Cucerzan, 2007; Yamada et al., 2016; Tsai and Roth, 2016), and for which we also provide a *data-driven verification* in § 6. By virtue of this assumption, one may infer that the mentions *Michael Jordan* and *Science* in the example from Fig. 1 link to entities from the realm of *science*. Moreover, as illustrated for toy data in Fig. 1 and later verified for real data (*cf.* § 6), gold entities tend to form a semantically dense subset of all candidate entities in the document and are therefore expected to lie in the subspace or in its proximity.

EIGENTHEMES defines a similarity function that will output high values for entities whose embeddings lie in the proximity of the subspace. Eventually, these scores are used to perform collective disambiguation.

Contributions.

- We propose EIGENTHEMES, a light-weight and fully unsupervised approach possessing the capability to incorporate external signals as weights for learning improved subspaces (§ 4).
- We propose multiple strong baselines that compare favorably to (and sometimes even outperform) the current state of the art τMIL-ND (Le and Titov, 2019), and showcase the superiority of EIGENTHEMES over the considered methods on four benchmark datasets (§ 5).
- For the first time, we provide a data-driven justification for the popular and long-standing assumption regarding the relatedness among gold entities mentioned within a document being higher than any other subset of candidate entities (§ 6).

2   Related Work

Unsupervised EL. Recent works (Le and Titov, 2019; Logeswaran et al., 2019) have spurred interest in EL with no or limited annotated data. Le and Titov (2019) propose τMIL-ND, which uses distant supervision to compute a compatibility score between each of the candidate entities and the context of the mention. Entities are decomposed as sets of entity properties (types) and their representations are learned simultaneously with all other model parameters. The model is trained using a weak signal for scoring all entities in the (positive) set of candidates of a mention higher than all entities in a (negative) set of randomly sampled entities. Candidate entities are ranked based on the computed local compatibility scores. While the setting is identical to ours, the approaches are

radically different. A downside of τMIL-ND (Le and Titov, 2019) is that it requires validation of a large number (15) of hyperparameters, which becomes problematic if the performance relies on an extensive hyper-parameter tuning on a per-dataset basis. Logeswaran et al. (2019) train an entity linking model using an annotated dataset and apply it to another dataset where annotations are not available. Entities are only identified by (long) textual descriptions, and the model relies strictly on language understanding to resolve the new entities. Distant supervision for EL has also been explored by Fan et al. (2015), however, their approach relies on features extracted from entity descriptions available from Wikipedia articles. We do not compare to Logeswaran et al. (2019) and Fan et al. (2015), as, unlike ours, their approach cannot be generalized to set-ups where such entity descriptions are not available.

Applications of low-rank subspaces. While low-rank subspaces have been employed in diverse application domains, *e.g.*, face recognition (Turk and Pentland, 1991), link prediction (West et al., 2009), distributional semantics (Levy and Goldberg, 2014; Mu et al., 2017), sense disambiguation (Mu et al., 2016), *etc.*, to the best of our knowledge, they have never been used for performing EL. We also emphasize our non-obvious use of low-rank subspaces: (1) we compute one decomposition per document, whereas most aforementioned applications compute a single corpus-wide decomposition, and (2) we incorporate weights.

3 Problem and Notation

We first formalize the task of interest. Let D be a single document from a collection $\mathcal{D}$ of documents. Similar to most previous works in the EL literature, with few exceptions (Sil and Yates, 2013; Kolitsas et al., 2018), we assume that mention spans (usually obtained by a named entity recognizer) are provided. Also, let $\mathcal{M}_D = \{m_1, m_2, \ldots, m_N\}$ be the set of N mentions contained in D, and $\mathcal{E}$ be the set of entities in the reference KG $\mathcal{G}$. A low-dimensional representation (embedding) can be learned for each entity by applying node representation learning techniques such as DEEPWALK (Perozzi et al., 2014) to the graph $\mathcal{G}$. The entity embeddings learned by these techniques are known to be meaningful with respect to the relatedness of the entities they represent (Almasian et al., 2019). The EL task consists in finding, for each mention

$m \in \mathcal{M}_D$, the entity $e \in \mathcal{E}$ to which it refers. A key component of any EL system is the candidate generation system. Let $\mathcal{C}_T$ denote such a system, which, for a given mention m, retrieves at most T entities—typically referred to as candidate entities. In the simplest case, the candidate generation system may retrieve all possible candidates (then $T = \infty$) (Yamada et al., 2016) for a given mention, but it is common to most of the works to retrieve a small subset of the most likely entities for that mention based on a certain sorting criterion. There is no consensus in the literature regarding a suitable value of T, and previous works have used varying values such as 7 (Ganea and Hofmann, 2017), 20 (Upadhyay et al., 2018) or 30 (Kolitsas et al., 2018). For each mention $m \in \mathcal{M}_D$, the set of candidate entities is given by $\mathcal{C}_T(m) = \{e_m^1, e_m^2, \ldots, e_m^T\}$, where e_m^i indicates the ith candidate entity for the mention m. For the example introduced in § 1, the candidate entities for the mentions *Michael Jordan* and *Science* are portrayed in Fig. 1.

A second key component of any entity linker is the disambiguation module. This module selects an entity contained in the referent KG $\mathcal{G}$ for each mention of a document. The disambiguation process may be performed independently for each mention, or collectively for all mentions in a document. In the latter case, the fundamental assumption of all collective disambiguation algorithms is that the set of gold entities are topically related.

Setting. Contrary to most of the existing work, where both the candidate generation and the disambiguation modules are dependent on annotated data, we focus on the setting where we only know the referent KG $\mathcal{G}$ and the mentions to be disambiguated, but have no annotated data that could generalize to the entities contained in $\mathcal{G}$.

4 Entity Linking with EIGENTHEMES

EIGENTHEMES learns a low-rank subspace from the ensemble of embeddings corresponding to all candidate entities for all mentions within a document (§ 4.1). We will see in § 4.2 that the learning of the subspace may also be guided by different weights associated with the embeddings. The learned subspace is represented by what we refer to as "eigenthemes". The eigenthemes are components that are learned so as to decompose the ensemble of the entity embeddings as a linear combination of these components.

The vector–subspace similarity function (§ 4.3) takes the learned low-rank subspace and the vector representation of a candidate entity, and computes a similarity score as a weighted sum of similarities between the entity vector representation and each of the eigenthemes of the subspace. The similarity between the entity representation and an eigentheme indicates how much of the signal from the former can be projected into the latter, and it is weighted with a value that relates to the strength of the eigentheme in the entity embedding ensemble of the document. As a result, only those entities that lie in the learned subspace or in its proximity will have a high score. It is important to note that while the vector–subspace similarity function is applied to each mention's candidates independently, the learned subspace encompasses information from all mentions in a document. Therefore, the disambiguation performed by EIGENTHEMES is collective, as the subspace given by the eigenthemes is learned at a global level in the document.

Additional details (pseudocode, architectural overview, *etc.*) about EIGENTHEMES are presented in Appendix B.

4.1 Subspace Learning

Let $\mathbf{E}$ be the d-dimensional embeddings for the entities $\mathcal{E}$ in the reference knowledge graph $\mathcal{G}$. Given a document D, we build the candidate space CS by taking the union of the candidate entities given by $\mathcal{C}_T(m)$ for all the mentions $m \in \mathcal{M}_D$ in the document. The $n_D \times d$ document embedding matrix $\mathbf{E}_D$ results from mapping all the n_D entities in CS to their corresponding embeddings in $\mathbf{E}$ and stacking them as rows.

To learn the subspace $\mathcal{S}_D$ of $\mathbf{E}_D$, we use the Singular Value Decomposition (SVD), one of the most fundamental techniques for matrix factorization. The SVD of $\mathbf{E}_D$ decomposes each entity embedding in the matrix as a linear combination of left and right singular vectors, as well as singular values. The truncated SVD obtains a rank-k ($k < \min(n_D, d)$) approximation $\tilde{\mathbf{E}}_D$ to $\mathbf{E}_D$ by just keeping the k largest singular values and their associated left and right singular vectors. The approximated rank-k matrix is

$$\tilde{\mathbf{E}}_D = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^\top. \quad (1)$$

where $\mathbf{U}_k$ and $\mathbf{V}_k$ are the $n_D \times k$ and $d \times k$ orthonormal matrices that preserve the k left and right singular vectors respectively, and $\mathbf{\Sigma}_k$ is a $k \times k$ diagonal matrix of the k largest singular values.

Each entity embedding is approximated as a linear combination—with coefficients given by the rows of $\mathbf{U}_k \mathbf{\Sigma}_k$—of the right singular vectors, *i.e.* the columns of $\mathbf{V}_k$. The columns of $\mathbf{V}_k$ are the eigenthemes that form the subspace (hyperplane) to be used to perform collective disambiguation. The aforementioned coefficients are determined by $\mathbf{U}_k$, whose rows are entity-specific, and $\mathbf{\Sigma}_k$, which relates to the global strength of each eigentheme in $\mathbf{E}_D$. As discussed previously, the proposed similarity function to score candidate entities leverages both eigenthemes $\mathbf{V}_k$ and their strength $\mathbf{\Sigma}_k$.

Avoiding norm induced bias. Eckart and Young (Eckart and Young, 1936) proved that $\tilde{\mathbf{E}}_D$ is the solution to the following optimization problem.

$$\arg\min_{\mathbf{X}} ||\mathbf{E}_D - \mathbf{X}||_F^2 \quad \text{subject to rank}(\mathbf{X}) = k.$$

That is, $\tilde{\mathbf{E}}_D$ is the rank-k approximation to $\mathbf{E}_D$ that minimizes the Frobenius norm between both matrices. Thus, the truncated SVD is affected by the norm of the embeddings used to construct $\mathbf{E}_D$. It is for this reason that, prior to the learning of the subspace, each of the entity embeddings of the $\mathbf{E}_D$ matrix is normalized to have unit L_2 norm. Otherwise, the learning of the subspace would be driven by entities whose embeddings' norm is larger. Having normalized all the embeddings and chosen the number k of components to be small, only the data points of the dense region of the embedding space will lie in the learned subspace or in its proximity.

4.2 Incorporating Weights

Let $\mathcal{W}$ be a weighting scheme[1] for assigning weights to all the candidate entities in a document based on an external signal, which may capture a prior likelihood of an entity to be the gold entity. The weights provided by $\mathcal{W}$ can serve to guide the learning of the subspace. The weights associated with entities have an impact in the optimization problem solved by the SVD. Consequently, the low-rank decomposition prioritizes accurate approximation of the entity embeddings contained in the document embedding matrix based on both the density of the entities in the embedding space and their associated weights.

We follow a simple approach to incorporate weights in the subspace learning by scaling each row of the document embedding matrix $\mathbf{E}_D$ with the weights given by an $n_D \times n_D$ diagonal matrix $\mathbf{W}_D$. Therefore, the subspace $\mathcal{S}_D$ is learned

---

[1]Weighting schemes are described in Appendix D.4.

from the rank-k approximation of $\mathbf{W}_D \mathbf{E}_D$. The values in $\mathbf{W}_D$ are obtained by applying the weighting scheme $\mathcal{W}$ to each of the n_D candidate entities. The weighting scheme enriched subspace is then learned by performing an eigendecomposition of the weighted sums of squares and cross product (SSCP) matrix, which is formally stated as follows:

$$(\mathbf{W}_D \mathbf{E}_D)^\top (\mathbf{W}_D \mathbf{E}_D) = \mathbf{E}_D^\top \mathbf{W}_D^2 \mathbf{E}_D. \quad (2)$$

The benefits of this extension are: (1) each candidate entity is enriched with some evidence about being a gold entity, and (2) the number of candidates per mention T can be increased without having a considerable negative impact on the learned subspace, as unlikely candidate entities will be penalized by the weighting scheme. Empirically, weighted EIGENTHEMES obtains better performance and is more robust to the parameter T than its vanilla unweighted version (detailed analysis in Appendix D.4).

4.3 Similarity Function

The subspace $\mathcal{S}_D$, determined by the eigenthemes $\mathbf{V}_k$ and their strength $\mathbf{\Sigma}_k$, learned from all the mentions in a document is at the core of our similarity function, and allows to disambiguate every mention in the document. For each mention, we project the embeddings of the candidate entities into the subspace $\mathcal{S}_D$ and select the candidate with the largest norm. Candidate entities close to the subspace will score highly, while those orthogonal to the subspace will obtain a low score.

Formally, for a mention m, the score of the ith candidate entity e_m^i is computed as follows:

$$\text{score}(e_m^i) = ||\mathbf{e}_m^i \mathbf{V}_k \mathbf{\Sigma}_k||_2, \quad (3)$$

where $\mathbf{e}_m^i$ is the embedding of the entity e_m^i. We observed that re-scaling the projection of the entity embedding into the eigenthemes with the corresponding strengths Σ_k leads to better performance.

5 Experiments

All the resources (code, datasets, *etc.*) required to reproduce the experiments in this paper are available at https://github.com/epfl-dlab/eigenthemes.

5.1 Datasets

We present results on real-world benchmark datasets (Table 1) that are popular in the entity linking literature. The considered datasets constitute a judicious mix of scale and domain types. For details about the datasets, please see Appendix C.

Table 1: Datasets and their statistics. Mentions where the gold-entity was not contained in the candidate-set are marked 'Not-found'. The 'easy' and 'hard' categorization of mentions is discussed in § 5.1.

Dataset	#Mentions			#Documents	Oracle Recall
	#Easy	#Hard	#Not-found		
CoNLL-Val	2645	1542	590	216	87.7%
CoNLL-Test	2555	1417	506	231	88.7%
WNED-Wiki	2731	3025	938	318	85.9%
WNED-Clueweb	4667	4711	1662	320	84.9%
Wikilinks-Random	26373	17397	10729	2336	80.3%

Candidate generation. We employ a simple and efficient approach to generate candidates, which is similar in design to the candidate generator used by Le and Titov (2019). Given a mention m, the entities that contain all the words from m are considered candidate entities. For example, MICHAEL JORDAN (BASKETBALL PLAYER) and MICHAEL JORDAN (COMPUTER SCIENTIST) are candidates for the mention MICHAEL JORDAN, while MICHAEL JACKSON is not. Since the degree of an entity roughly captures its popularity, the candidate entities are sorted based on the degree of their corresponding vertices in the undirected version of the Wikidata KG. Although simple, the effectiveness and practicality of the candidate generator is corroborated by the high oracle recall (percentage of mentions where the true entity is present in the set of candidates) obtained across all the datasets presented in Table 1. It is important to note that the methodology of EIGENTHEMES is orthogonal to the candidate generator. While the latter could be improved (by modifying the string matching heuristic (Sil et al., 2012; Charton et al., 2014), or using word embeddings to match words in entity names and words in the mention, *etc.*), which will only improve the performance of any technique, it is beyond the scope of this work.

Preprocessing. We consider Wikidata as our referent KB. The gold entity annotations for mentions available as Wikipedia page ids in all the aforementioned datasets were appropriately mapped to their corresponding Wikidata identifiers. To ensure that our empirical analyses and the corresponding conclusions are representative of various real-world scenarios requiring entity linking, we, similar to Tsai and Roth (2016) and Guo and Barbosa (2018), introduce the *'easy'* and *'hard'* categorization for mentions in all the datasets. A mention is termed as 'easy' if the first candidate entity (in the list of sorted candidates returned by the candidate generator) is the 'true' entity, and 'hard' otherwise.

Table 2: EL quality measured using precision@1 and MRR on the CoNLL-Test dataset. The best performance is shown in **bold**. For τMIL-ND and WτMIL-ND, we perform 5 independent runs and report the mean and standard deviation. All other techniques are deterministic. Note that the parameter T was fixed to 20 candidates per mention, and thus, '#Hard' mentions reduces from 1417 to 1136 while '#Not-found' increases from 506 to 787.

Category	Method	Precision@1			MRR		
		Overall[#]	Easy	Hard	Overall[#]	Easy	Hard
Existent	NAMEMATCH (Riedel et al., 2010)	0.412	0.645	0.174	0.415	0.645	0.184
Existent	τMIL-ND (SoTA) (Le and Titov, 2019)	0.451 ± 0.019	0.700 ± 0.032	0.187 ± 0.006	0.539 ± 0.017	0.777 ± 0.029	0.353 ± 0.005
Proposed	LOCAL CTXT	0.296	0.420	0.223	0.401	0.537	0.374
Proposed	GLOBAL CTXT	0.303	0.403	0.289	0.423	0.542	0.448
Proposed	DEGREE	0.571	**1.0**[†]	0.0	0.649	**1.0**[†]	0.312
Proposed	AVG	0.488	0.658	0.445	0.593	0.756	0.636
Proposed	WτMIL-ND	0.499 ± 0.022	0.778 ± 0.037	0.217 ± 0.008	0.592 ± 0.018	0.853 ± 0.030	0.415 ± 0.007
Proposed	EIGEN	**0.617**[†]	0.858	**0.500**[†]	**0.690**[†]	0.910	**0.674**[†]
-	Ceiling	0.824	1.0	1.0	0.824	1.0	1.0

[†] Indicates statistical significance ($p < 0.01$) between the best and the second-best method using the Student's paired t-test.
[#] Overall is computed by considering all mentions (including Not-found in addition to Easy and Hard).

5.2 Methods Benchmarked

Existing methods. To the best of our knowledge, surface name matching (NAMEMATCH) and τMIL-ND are the only methods tried in previous work.

• **NAMEMATCH (Riedel et al., 2010):** For each mention m, NAMEMATCH retrieves all Wikidata entities whose names match exactly with the mention string. In the event of multiple matching entities, we choose the entity with the highest KG degree as the prediction for mention m.

• τ**MIL-ND (Le and Titov, 2019):** The current state of the art (SoTA), which is trained using the New York Times dataset provided by Le and Titov.

Newly introduced baselines. We introduce five creative and simple, yet effective baselines.

• **LOCAL CTXT:** computes a representation of the local context of a mention as the average of its surrounding words' embeddings. For each mention m, candidate entities are ranked based on their semantic similarity with the context, and the one with the highest similarity is chosen as the prediction for m. Entity representations are computed from their textual descriptions (details in Appendix D.3). We set the context window size to 5^2.

• **GLOBAL CTXT:** follows the exact same procedure as LOCAL CTXT, with the only difference being the size of the context window. Following convention in the literature (Yamada et al., 2016), we use all the nouns in a document to obtain the global context representation.

• **DEGREE:** a natural baseline (courtesy of our candidate generator) for performing EL without annotated data. For each mention m, entity degree is used to obtain a ranking of the candidate entities,

and the one with the highest degree is chosen as the prediction for m.

• **Average (AVG):** constructs a representation of $\mathbf{E}_D$ as a d-dimensional vector by computing the average of the rows of $\mathbf{E}_D$. Each candidate entity is scored by computing the cosine similarity between its embedding and the AVG based representation.

• **WτMIL-ND:** extends the current SoTA τMIL-ND by incorporating the weights used by EIGEN into its compatibility scoring function.

EIGENTHEMES (EIGEN). Our proposed solution.

Unless stated otherwise, weights are employed for AVG and EIGEN. Note that we opt against purely supervised baselines as running them correctly in the "true" unsupervised setting is inconceivable (*cf.* challenges in § 1).

5.3 Setup

Embeddings. For word embeddings, we use the publicly available (Archive, 2013) 300-dimensional vector representation of words obtained by training WORD2VEC (Mikolov et al., 2013) on the Google News dataset. The 128-dimensional entity embeddings were obtained by training DEEPWALK (Perozzi et al., 2014) on the Wikidata knowledge graph.

Weighting scheme. The candidate entities of a mention are weighted using their reciprocal rank, where the ranking is induced by the candidate generator (Le and Titov, 2019).

Parameters. The maximum number of candidates per mention T is fixed to 20. Note that restricting T also results in a reduction in the oracle recall. We fix the number k of components for constructing the subspace representation using EIGEN to 10.

---

^{2}Other window sizes were evaluated, however, the chosen value 5 resulted in the best downstream EL performance.
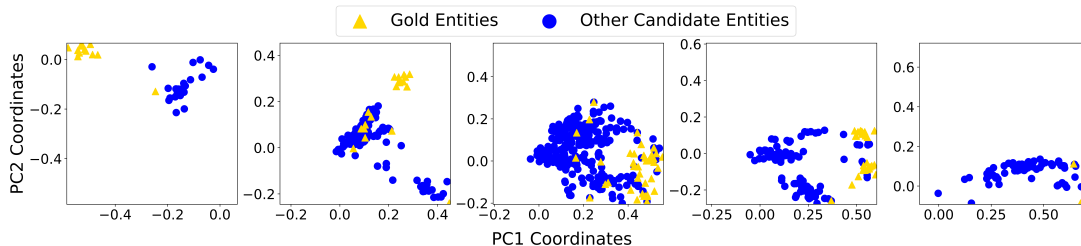
Figure 2: Scatter plots for 5 documents sampled randomly from the CoNLL-Test dataset. For each document, we project all the candidate entities on the first two components (PC1 and PC2) of the subspace learned by EIGEN.

Evaluation metrics. We use (1) precision@1, and (2) mean reciprocal rank (MRR) to evaluate the quality of the methods benchmarked in this study. Following convention in the literature, we compute *Micro* aggregates of the metrics over all mentions.

Additional details about the experimental setup (effect of different embedding methods, weighting schemes, the number of candidates T, *etc.*), and hyperparameter tuning (effect of k) are present in Appendix D and E, respectively.

5.4 Results: CoNLL

We assess the efficacy of EIGEN by comparing its quality measured using precision@1 and MRR with the considered methods on the CoNLL-Test dataset. The results are presented in Table 2. Note that Ceiling corresponds to the oracle recall of the candidate generator and provides an upper bound on precision@1 and MRR.

Overall performance. It is evident that EIGEN achieves the best overall quality and significantly outperforms all the considered methods. The key highlights are as follows: (1) EIGEN obtains an improvement of 15 percentage points over the existing SoTA τMIL-ND, (2) AVG, a hard-to-beat baseline (Arora et al., 2017; Wu et al., 2019) and a natural competitor of EIGEN, is around 12 percentage points inferior to EIGEN, which substantiates our intuition (*cf.* § 1 and 4) about the superiority of subspaces in capturing global topical relatedness among entities in a document, and (3) DEGREE, a simple baseline that we introduce in this work, substantially outperforms all the other considered methods—even the existing SoTA, τMIL-ND—and is second only to EIGEN, which remains about 5 percentage points and statistically significantly better (p-value < 0.01) than DEGREE.

Performance on 'easy' and 'hard' mentions. The key finding is that the closest competitors—WτMIL-ND and DEGREE—of EIGEN lack robustness to the variation in mention-types, which is substantiated by the huge disparity of EL perfor-

mance measured using precision@1 (with similar trends observed for MRR) for both WτMIL-ND (78% for easy vs. 22% for hard mentions) and DEGREE (100% for easy vs. 0% for hard mentions). This result highlights the key limitation of WτMIL-ND and DEGREE that they cannot address challenging scenarios of EL. A detailed analysis about the potential effects of this limitation is performed in § 6, while a discussion on the potential reasons behind the existence of this disparity in performance is presented in Appendix D.5.

6 Analysis

In this section, we perform a post-mortem analysis on a plethora of aspects impacting the downstream EL performance of the considered methods measured using precision@1. Results for MRR show similar trends and are therefore omitted.

Do learned subspaces capture the relatedness among gold entities? Fig. 2 shows that gold entities tend to form tight clusters when projected on the first two components of the subspace learned by EIGEN. This result provides a data-driven justification regarding the existence of relatedness among gold entities mentioned within a document—a key assumption used in the design of almost every method in the EL literature. Moreover, it also showcases the ability of the subspaces learned by EIGEN to capture such relatedness.

Do gold entities lie closer to the learned subspaces when compared to other candidate entities? The disambiguation module of EIGEN relies on the gold entities of a document being closer to the learned subspace than other candidate entities. While the strong EL performance of EIGEN portrayed in Table 2 provides substantive evidence in favor of the aforementioned property, we conduct a more direct assessment, which is described as follows. For every mention in a document, we project all candidate entities onto the subspace learned by EIGEN and compare the score (Eq. 3) of the gold

entity to the average score (Eq. 3) of the non-gold entities, finding that the gold entity's score G is *statistically significantly higher* (81% on average) than the score N of the average non-gold entity: $\frac{G-N}{N} = 0.81$, with bootstrapped 95% confidence interval $[0.78, 0.84]$. This finding convincingly substantiates the fact that incorrect candidate entities *rarely* exhibit stronger topical relatedness than gold entities. In the event that incorrect candidates also possess strong relatedness, EIGEN (by design) can differentiate between the true and spurious relatedness by leveraging weighting schemes.

Does the distribution of mention-types affect EL performance? Inspired by the *cloze* test (Taylor, 1953) used to assess language learning capability of individuals (Henning et al., 1983), we perform a *mutilation analysis* to assess the impact of the distribution of mention-types on the downstream EL performance. The test is carried out by removing varying number of 'easy' mentions from the CoNLL-Test dataset, thereby increasing the fraction of 'hard' (ambiguous) mentions, and thus, can be deemed analogous to adding noise in the dataset. Specifically, we generate 11 different dataset versions by subsampling (uniformly at random) varying fraction (between 1 and 0 in decrements of 0.1) of easy mentions while retaining all the hard mentions, and measure the 'overall' EL performance in each version. We run this experiment 10 times and report the mean performance. Being very small, the standard deviations were omitted to avoid clutter.

Fig. 3a portrays that the overall EL performance of all the techniques deteriorates with the reduction in the fraction of easy mentions, which is natural. Interestingly, the deterioration observed for the DEGREE baseline is much more profound—precision@1 plummets from 0.57 (fraction-easy = 1) to 0 (fraction-easy = 0)—when compared to all other techniques, and it gets demoted from being the second best technique to the worst. Note that even AVG starts outperforming DEGREE when fraction of easy mentions are less than 0.5. On the contrary, EIGEN consistently outperforms all other techniques in all the subsampled versions.

This experiment establishes two important points. (1) It shows that the performance of DE-GREE is *deceptively high*. More importantly, it unveils a key limitation of DEGREE, *i.e.*, its inability to address challenging scenarios of EL where the fraction of easy mentions is low. (2) It bolsters
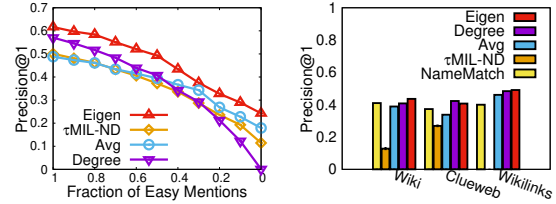
(a) Mutilation Analysis   (b) Out-of-domain Datasets

Figure 3: Analyzing the (a) effect of mention-type distribution, and (b) performance on out-of-domain datasets using precision@1.

the superiority of EIGEN over other techniques by showcasing its ability to reliably and consistently perform well, even in challenging scenarios.

Robustness: Out-of-domain Datasets. It is standard practice in the EL literature (Ganea and Hofmann, 2017; Le and Titov, 2018) to evaluate the robustness of the models trained on the CoNLL-train dataset in out-of-domain datasets without any (re)validation of hyperparameters. In the same vein, the experiments on all the other datasets were performed using the hyperparameters tuned on the CoNLL-Val dataset. Specifically, we assess the robustness of the techniques on the recently introduced WNED-Wiki and WNED-Clueweb datasets, and the Wikilinks-Random dataset of English Wikipedia tables. The results are presented in Fig. 3b. Note that τMIL-ND cannot perform EL in the Wikilinks-Random dataset owing to the absence of sequential textual content, and therefore, the corresponding bar in the plot is non-existent.

It is worth noting that τMIL-ND (the existing SoTA) performs the worst in the out-of-domain scenario, whereas all other techniques exhibit robustness. Hence, this experiment simply unveils a limitation of τMIL-ND, *i.e.*, its lack of robustness, which is a noteworthy finding. Consequently, the experiment also establishes two important properties of EIGEN: (1) robustness to hyperparameter tuning, and (2) adaptability to newer domains.

7 Discussion

Summary of Results. A strong entity linking method for scenarios with no annotated data should stand strong on three pillars: efficacy towards entity disambiguation, scalability to large datasets, and robustness to hyperparameter tuning and mention-type distributions. We performed an extensive comparison of EIGEN with 7 techniques across each of the aforementioned features, and while a quantitative analysis has already been performed, we provide a visual summary in Fig. 4.
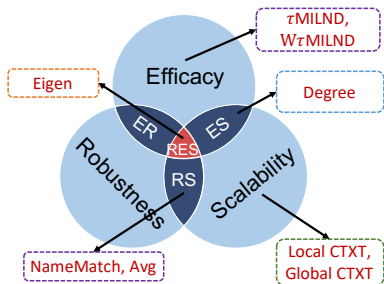
Figure 4: A qualitative summary of the spectrum of unsupervised EL techniques based on their strengths.

EIGEN (classified as "RES") stands strong on all three pillars. (1) It is robust to the presence of noise in the data; provides simplicity of hyperparameter tuning as it possesses just one hyperparameter, *viz.*, the number of components to construct the sub-space; and also portrays robustness to the hyper-parameter tuning step. (2) Its light-weight nature provides the ability to gracefully scale to Web-scale datasets. Empirically, EIGEN requires around 2–15 minutes, while τMIL-ND (Le and Titov, 2019) requires around 200–220 minutes to perform entity linking for the considered datasets. Thus, our approach is approximately 10 to 100 times more efficient. (3) It portrays the best efficacy across all metrics, settings, and datasets. While DEGREE (classified as "ES") is a promising technique, its lack of robustness to the varying easy-hard proportion of queries in the datasets serves as a concerning disadvantage. As argued in the "challenges" paragraph of § 1, robustness is a desired characteristic in a setting where no annotated data is available.

Some of the additional advantages of our method include: (1) language independence owing to the reliance on entity embeddings alone, and (2) capability to improve existing methods in settings where annotated data is available (Appendix F). In summary, EIGENTHEMES provides an effective, efficient, and scalable solution to the entity linking problem in the absence of annotated data.

EIGENTHEMES *vs*. Clustering. While Fig. 2 shows that gold entities tend to be clustered together in the eigenspace, producing internally-tight, mutually-far clusters (as obtained by, say, k-means (Lloyd, 1982; MacQueen, 1967)) is neither the optimization objective nor a requirement for EIGEN-THEMES to work. Moreover, using clustering to disambiguate gold entities from other candidate entities is easier said than done, as even with a perfect clustering where gold entities lie in a single coherent cluster, it is unclear how to identify which cluster contains these gold entities.

Domain-specific KGs. Domain-specific KGs—associated with text corpora in niche domains such as cinema, law, medicine, or science—are usually sparser than Web-scale KGs such as Wikidata, which could lead to low-quality entity embeddings (Pujara et al., 2017), thereby directly affecting the performance of EIGENTHEMES. To this end, we compare the information density (#relationships-per-entity) (Pujara et al., 2017) of the Wikidata KG variant (Appendix C.2) used in the current work to learn entity embeddings, with a plethora of domain-specific and Web-scale KGs. Specifically, the information density of our Wikidata KG (5.5) is similar to that of domain-specific KGs, namely: IMDb (4.5) (Rossi and Ahmed, 2015) and SNOMED (7.1) (Chang et al., 2020), and much smaller than Web-scale KGs, namely: DBpedia (26) (Auer et al., 2007), Freebase (16) (Bollacker et al., 2008), and Wikidata (12.5) (Vrandečić and Krötzsch, 2014). Moving beyond aggregate statistics, even the degree distribution of our Wikidata KG is similar to that of the aforementioned domain-specific KGs. These statistics indicate that EIGENTHEMES can readily perform EL in domains with sparse KGs.

Unobserved Entities. In its current state, EIGEN-THEMES cannot perform EL for KG entities that were not observed during training of entity embeddings. Inductive learning (Hamilton et al., 2017) of entity embeddings serves as a potential fix, however, such extensions constitute as future work.

8 Conclusions

In this paper, we addressed the problem of EL in the absence of annotated data with a light-weight method, EIGENTHEMES, that relies solely on the availability of entity names and a referent KB. Experiments on benchmark datasets portrayed the effectiveness of our proposed approach. In the future, our aim is to validate the effectiveness of our approach in performing multi-lingual entity linking with a special focus on low-resource languages.

Acknowledgements

Broader Impact

Entity linking is a broad problem with diverse application areas ranging from natural language processing to web data management. A less popular yet important application of entity linkers is their ability to act as an enabling technology for improving the navigability of networks such as Wikipedia, which depends heavily on the amount and diversity of hyperlinks joining pairs of articles. Moreover, they are also critical in facilitating users and machines to obtain a better understanding of text corpora by having ambiguous pieces of text linked to their corresponding unambiguous concepts. Entity linking could be deemed as a solved problem in scenarios where annotated data is available, however, there is a scarcity of methods capable of performing entity linking without access to annotated data. Consequently, designing effective and efficient solutions for entity linking without annotated data is an important avenue, and our current work is a step in that direction. The core contribution of this paper is a light-weight and language agnostic approach to unsupervised entity linking. While we are not the first to formulate the unsupervised entity linking problem, we are the first to propose a solution capable of operating on Web-scale datasets, which is a fundamental requirement for practical entity linkers. Moreover, we are also the first to introduce a suite of simple and intuitive, yet effective baselines that would serve as strong benchmarks thereby enabling researchers to conduct follow-up research in this nascent but important research area.

References

Satya Almasian, Andreas Spitz, and Michael Gertz. 2019. Word Embeddings for Entity-annotated Texts. In *ECIR*, pages 307–322.

Apache. 2019. The Apache OpenNLP Tagger. https://opennlp.apache.org/.

Google Code Archive. 2013. The Word2vec Project. https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/edit?usp=sharing.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A Simple but Tough-to-Beat Baseline for Sentence Embeddings. In *ICLR*.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. DBpedia: A Nucleus for a Web of Open Data. In *ISWC*, page 722–735. https://www.dbpedia.org/.

Chandra Sekhar Bhagavatula, Thanapon Noraset, and Doug Downey. 2015. TabEL: Entity Linking in Web Tables. In *ISWC*, pages 425–441.

Ayan Kumar Bhowmick, Koushik Meneni, Maximilien Danisch, Jean-Loup Guillaume, and Bivas Mitra. 2020. LouvainNE: Hierarchical Louvain Method for High Quality and Scalable Network Embedding. In *WSDM*, page 43–51.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. In *SIGMOD*, page 1247–1250. https://developers.google.com/freebase.

Leo Breiman. 2001. Random Forests. *Machine Learning*, 45(1):5–32.

David Chang, Ivana Balažević, Carl Allen, Daniel Chawla, Cynthia Brandt, and Andrew Taylor. 2020. Benchmark and Best Practices for Biomedical Knowledge Graph Embeddings. In *BioNLP Workshop*, pages 167–176.

Eric Charton, Marie-Jean Meurs, Ludovic Jean-Louis, and Michel Gagnon. 2014. Improving Entity Linking using Surface Form Refinement. In *LREC*, pages 4609–4615.

Silviu Cucerzan. 2007. Large-scale Named Entity Disambiguation based on Wikipedia Data. In *EMNLP-CoNLL*, pages 708–716.

Carl Eckart and Gale Young. 1936. The Approximation of One Matrix by Another of Lower Rank. *Psychometrika*, 1(3):211–218.

Miao Fan, Qiang Zhou, and Thomas Fang Zheng. 2015. Distant Supervision for Entity Linking. In *PACLIC*, pages 79–86.

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). http://lemurproject.org/clueweb12/.

Octavian-Eugen Ganea and Thomas Hofmann. 2017. Deep Joint Entity Disambiguation with Local Neural Attention. In *EMNLP*, pages 2619–2629.

Yuqing Gao, Jisheng Liang, Benjamin Han, Mohamed Yakout, and Ahmed Mohamed. 2018. Building a Large-scale, Accurate and Fresh Knowledge Graph. In *KDD*.

Zhaochen Guo and Denilson Barbosa. 2018. Robust Named Entity Disambiguation with Random Walks. *Semantic Web*, 9(4):459–479.

Nitish Gupta, Sameer Singh, and Dan Roth. 2017. Entity Linking via Joint Encoding of Types, Descriptions, and Context. In *EMNLP*, pages 2681–2690.

Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *NeurIPS*.

Grant Henning, Norman Gary, and Judith O. Gary. 1983. Listening recall—a listening comprehension test for low proficiency learners. *System*, 11(3):287 – 293.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust Disambiguation of Named Entities in Text. In *EMNLP*, pages 782–792.

Jan-Christoph Klie. 2019. Wikimapper. https://github.com/jcklie/wikimapper.

Nikolaos Kolitsas, Octavian-Eugen Ganea, and Thomas Hofmann. 2018. End-to-End Neural Entity Linking. In *CoNLL*.

Phong Le and Ivan Titov. 2018. Improving Entity Linking by Modeling Latent Relations between Mentions. In *ACL*, pages 1595–1604.

Phong Le and Ivan Titov. 2019. Distant Learning for Entity Linking with Automatic Noise Detection. In *ACL*, pages 4081–4090.

Omer Levy and Yoav Goldberg. 2014. Neural Word Embedding as Implicit Matrix Factorization. In *NIPS*, pages 2177–2185.

Girija Limaye, Sunita Sarawagi, and Soumen Chakrabarti. 2010. Annotating and Searching Web Tables Using Entities, Types and Relationships. *PVLDB*, 3(1-2):1338–1347.

Stuart Lloyd. 1982. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137.

Lajanugen Logeswaran, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Jacob Devlin, and Honglak Lee. 2019. Zero-Shot Entity Linking by Reading Entity Descriptions. In *ACL*.

James MacQueen. 1967. Some Methods for Classification and Analysis of Multivariate Observations. In *Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and Their Compositionality. In *NIPS*, pages 3111–3119.

David Milne and Ian H. Witten. 2008. Learning to Link with Wikipedia. In *CIKM*, page 509–518.

Jiaqi Mu, Suma Bhat, and Pramod Viswanath. 2016. Geometry of Polysemy. In *ICLR*.

Jiaqi Mu, Suma Bhat, and Pramod Viswanath. 2017. Representing Sentences as Low-Rank Subspaces. In *ACL*.

NumPy. 2019. The NumPy Linear Algebra Library. https://docs.scipy.org/doc/numpy/reference/routines.linalg.html.

Xiaoman Pan, Taylor Cassidy, Ulf Hermjakob, Heng Ji, and Kevin Knight. 2015. Unsupervised Entity Linking with Abstract Meaning Representation. In *NAACL*, pages 1130–1139.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *JMLR*, 12:2825–2830.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. DeepWalk: Online Learning of Social Representations. In *KDD*, pages 701–710.

Jay Pujara, Eriq Augustine, and Lise Getoor. 2017. Sparsity and Noise: Where Knowledge Graph Embeddings Fall Short. In *EMNLP*, pages 1751–1756.

Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Chi Wang, Kuansan Wang, and Jie Tang. 2019. NetSMF: Large-Scale Network Embedding As Sparse Matrix Factorization. In *WWW*, page 1509–1520.

Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 2011. Local and Global Algorithms for Disambiguation to Wikipedia. In *NAACL-HLT*, pages 1375–1384. Association for Computational Linguistics.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling Relations and Their Mentions without Labeled Text. In *ECML-PKDD*, pages 148–163.

Ryan A. Rossi and Nesreen K. Ahmed. 2015. The Network Data Repository with Interactive Graph Analytics and Visualization. In *AAAI*.

Avirup Sil, Ernest Cronin, Penghai Nie, Yinfei Yang, Ana-Maria Popescu, and Alexander Yates. 2012. Linking Named Entities to Any Database. In *EMNLP*, pages 116–127.

Avirup Sil and Alexander Yates. 2013. Re-ranking for Joint Named-entity Recognition and Linking. In *CIKM*, pages 2369–2374.

Valentin I. Spitkovsky and Angel X. Chang. 2012. A Cross-Lingual Dictionary for English Wikipedia Concepts. In *LREC*, pages 3168–3175.

Wilson L. Taylor. 1953. "Cloze Procedure": A New Tool for Measuring Readability. *Journalism Quarterly*, 30(4):415–433.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 Shared Task: Language-independent Named Entity Recognition. In *CONLL*, pages 142–147.

Chen-Tse Tsai and Dan Roth. 2016. Cross-lingual Wikification Using Multilingual Embeddings. In *NAACL-HLT*, pages 589–598.

Matthew A Turk and Alex P Pentland. 1991. Face Recognition using Eigenfaces. In *CVPR*, pages 586–591.

Shyam Upadhyay, Nitish Gupta, and Dan Roth. 2018. Joint Multilingual Supervision for Cross-lingual Entity Linking. In *EMNLP*, pages 2486–2495.

Shikhar Vashishth, Denis Newman-Griffis, Rishabh Joshi, Ritam Dutt, and Carolyn P. Rosé. 2021. Improving broad-coverage medical entity linking with semantic type prediction and large-scale datasets. *Journal of Biomedical Informatics*, 121:103880.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A Free Collaborative Knowledgebase. *Commun. ACM*, 57(10):78–85. https://www.wikidata.org/.

Han Wang, Jin Guang Zheng, Xiaogang Ma, Peter Fox, and Heng Ji. 2015. Language and Domain Independent Entity Linking with Quantified Collective Validation. In *EMNLP*, pages 695–704.

Robert West, Doina Precup, and Joelle Pineau. 2009. Completing Wikipedia's Hyperlink Structure through Dimensionality Reduction. In *CIKM*, page 1097–1106.

Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. 2019. Simplifying Graph Convolutional Networks. In *ICML*, pages 6861–6871.

Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2016. Joint Learning of the Embedding of Words and Entities for Named Entity Disambiguation. In *CoNLL*, pages 250–259.

Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2017. Learning Distributed Representations of Texts and Entities from Knowledge Base. *TACL*, 5:397–411.

Ikuya Yamada, Hiroyuki Shindo, and Yoshiyasu Takefuji. 2018. Representation Learning of Entities and Documents from Knowledge Base Descriptions. In *COLING*.

Appendix

A Additional Related Work

Sil et al. (2012) address a related task in a setting similar to ours. Given a mention, they generate a restricted number of candidate entities, which very often contain the gold entity, by exploiting associated structured information. Very recently, Vashishth et al. (2021) address the same issue for medical entity linking by pruning irrelevant candidate entities based on the prediction of semantic type of a mention. We believe that these works are complementary to EIGENTHEMES and could bring important benefits to disambiguation techniques in the absence of annotated data.

B Additional Details about EIGENTHEMES

The pipeline of EIGENTHEMES and its pseudocode are depicted in Fig. 5.

Discussion: The role of T and k in subspace learning. As stated in Sec. 4, the eigenthemes are components that are learned so as to decompose the ensemble of the entity embeddings as a linear combination of these components. If the number k of eigenthemes is chosen to be small, these components will constitute a good basis to approximate the dense region of the document embedding matrix. From the fundamental assumption of the existence of topical relatedness across the gold entities in a document and that such relatedness is captured by their corresponding embeddings (Almasian et al., 2019), the gold entities will form a dense region and, consequently, will define the subspace. However, this is only possible if there is no other subset of candidate entities whose relatedness is larger than that of the set of gold entities. The latter point relates to the hyperparameter T, which controls the maximum number of candidate entities per mention. A low value of T reduces the possibility of having another subset of entities with larger relatedness as well as the recall—the number of times that the gold entity is contained in the set of candidate entities. A large value of T increases both. In general, the larger the number of candidate entities per mention, the more the learned subspace will be affected by embeddings other than those from the gold entities.

C Datasets

C.1 Detailed Dataset Description

• The **AIDA-CoNLL** dataset (Hoffart et al., 2011) is one of the most popular datasets in the EL literature. It is based on the CoNLL 2003 shared task (Tjong Kim Sang and De Meulder, 2003) and contains high quality manual annotations for mention strings linking them to their target named entities. The dataset consists of three parts: training, validation, and test, but, we use only the validation (CoNLL-Val) and test (CoNLL-Test) sets owing to the fully unsupervised nature of our approach.
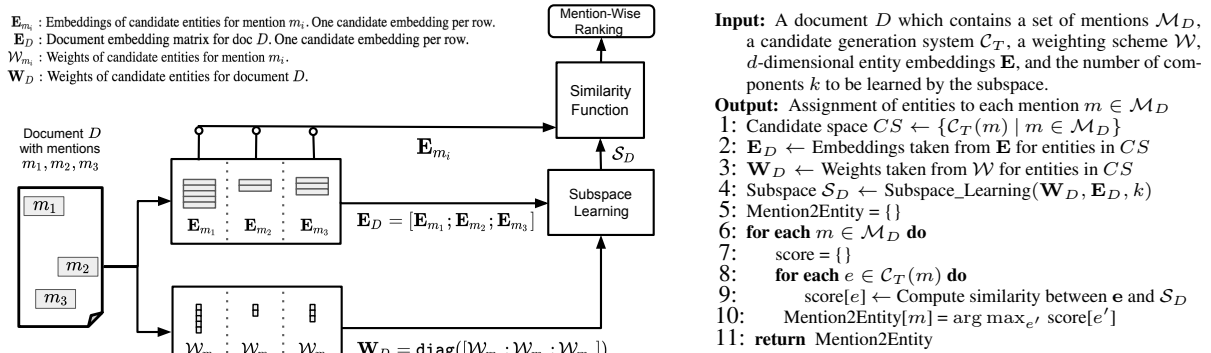
$\mathbf{E}_{m_i}$: Embeddings of candidate entities for mention m_i. One candidate embedding per row.
$\mathbf{E}_D$: Document embedding matrix for doc D. One candidate embedding per row.
$\mathcal{W}_{m_i}$: Weights of candidate entities for mention m_i.
$\mathbf{W}_D$: Weights of candidate entities for document D.

Input: A document D which contains a set of mentions $\mathcal{M}_D$, a candidate generation system $\mathcal{C}_T$, a weighting scheme $\mathcal{W}$, d-dimensional entity embeddings $\mathbf{E}$, and the number of components k to be learned by the subspace.
Output: Assignment of entities to each mention $m \in \mathcal{M}_D$
1: Candidate space $CS \leftarrow \{\mathcal{C}_T(m) \mid m \in \mathcal{M}_D\}$
2: $\mathbf{E}_D \leftarrow$ Embeddings taken from $\mathbf{E}$ for entities in CS
3: $\mathbf{W}_D \leftarrow$ Weights taken from $\mathcal{W}$ for entities in CS
4: Subspace $\mathcal{S}_D \leftarrow$ Subspace_Learning($\mathbf{W}_D, \mathbf{E}_D, k$)
5: Mention2Entity = {}
6: **for each** $m \in \mathcal{M}_D$ **do**
7: score = {}
8: **for each** $e \in \mathcal{C}_T(m)$ **do**
9: score[e] $\leftarrow$ Compute similarity between $\mathbf{e}$ and $\mathcal{S}_D$
10: Mention2Entity[m] = $\arg\max_{e'}$ score[e']
11: **return** Mention2Entity

Figure 5: (Left) Pipeline of EIGENTHEMES. (Right) Its pseudocode. $[\mathbf{a}; \mathbf{b}]$ is row-wise concatenation of $\mathbf{a}$ and $\mathbf{b}$.

• **WNED-Wiki** and **WNED-Clueweb** are recently introduced benchmark datasets by Guo and Barbosa (Guo and Barbosa, 2018) aimed at reducing the bias in the AIDA-CoNLL dataset towards popular entities. Specifically, these datasets are generated by uniformly sampling mentions with different levels of difficulty (determined by the prior scores) from the '2013-06-06' dump of English Wikipedia and Freebase annotations of the 2012 Clueweb corpora (FACC1) (Gabrilovich et al., 2013) respectively.

• The **Wikilinks-Random** dataset (Bhagavatula et al., 2015; Limaye et al., 2010) consists of tables extracted from the English Wikipedia with mentions and their corresponding links to Wikipedia pages. Note that a key difference between this and the other datasets is the scarcity of textual content in the table data.

C.2 Dataset Preprocessing Details

Referent KB. Wikidata (Vrandečić and Krötzsch, 2014) is considered as the referent KB. We use the n-triples format of the Wikidata dump[3] downloaded on May 3, 2019. Since the gold entity annotations for mentions in all the datasets benchmarked in this study are provided as either page-titles or page-ids of the English Wikipedia, we restrict our entity universe to the ones having links to English Wikipedia resulting in about 3.7M Wikidata entities. The mapping between Wikipedia and Wikidata entities was extracted using the Wikimapper tool (Klie, 2019) from the 'page_props.sql.gz' file available in the '2019-05-20' dump[4] of English Wikipedia. Eventually, each entity is denoted by its unique Wikidata identifier (QID). For example, the entity MICHAEL JORDAN (COMPUTER SCIENTIST) is denoted by his Wikidata QID: Q3308285.

nored from the CoNLL-Val, WNED-Wiki, WNED-Clueweb, and Wikilinks-Random datasets respectively as their ground truth Wikipedia entities could not be mapped to their corresponding Wikidata entities. After investigating the reason behind this, we found that the Wikipedia pages corresponding to these entities do not exist any more.

Candidate generation. The starting point of most existing techniques in the entity linking literature is the existence of a large amount of high quality annotated data. This allows to reliably estimate the probability that a mention m refers to an entity e, formally denoted as $P(e|m)$ and commonly referred to as *prior* probability. Thus, a common choice for most of the works (Bhagavatula et al., 2015; Yamada et al., 2016; Ganea and Hofmann, 2017; Gupta et al., 2017; Kolitsas et al., 2018) is to leverage the prior information collected by Spitkovsky and Chang (Spitkovsky and Chang, 2012), who crawled the Web for associating strings of text with entities in the underlying English Wikipedia knowledge graph. This prior information constitutes the basis of the candidate generation system used by most entity linking techniques. It has been shown multiple times that prior information itself represents the bulk of the performance of the entity linking methods (Ratinov et al., 2011; Bhagavatula et al., 2015; Tsai and Roth, 2016; Upadhyay et al., 2018), and hence, some authors (Bhagavatula et al., 2015; Yamada et al., 2016) have spent much effort in crawling additional Web resources to complement Spitkovsky and Chang's corpus.

However, in the absence of annotated data, such prior information is not available, and there is a requirement of mechanisms to generate a set of plausible candidate entities for a given mention string. To this end, and because we do not want to presume information more complex than what is readily available for arbitrary knowledge graphs,

---
[3] The 'latest-truthy.nt.gz' file at https://dumps.wikimedia.org/wikidatawiki/entities/
[4] https://dumps.wikimedia.org/enwiki/
Note that 1, 7, 12, and 6903 mentions were ig-

we employ a simple and efficient approach to generate candidates. For each entity e in the knowledge graph $\mathcal{G}$, we have access to its name and a list of aliases used to commonly refer to e. Note that the mention strings, entity names, and aliases are tokenized into words. Given a mention m, the entities that contain all the tokens from m are considered candidate entities. For example, MICHAEL JORDAN (BASKETBALL PLAYER) and MICHAEL JORDAN (COMPUTER SCIENTIST) are candidates for the mention MICHAEL JORDAN, while MICHAEL JACKSON is not. Although simple, the recall of this candidate generation approach is quite high and practical (as shown in Fig. 6) for the data sets considered in this work. The methodology to perform disambiguation is orthogonal to the candidate generator, and while the latter could be improved (Sil et al., 2012; Wang et al., 2015), which will only improve the performance of any technique, it is not in the scope of this work.

As the candidate generation system $\mathcal{C}_T$ only accounts for at most T most likely entities of a mention, a sorting criterion is required. To this end, candidate entities are sorted based on the degree of their corresponding vertices in the undirected version of the knowledge graph. The degree of an entity roughly captures its popularity.

A naïve implementation of the aforementioned string matching based candidate generator is impractical considering the sheer size of real-world knowledge graphs. To this end, we employ an *inverted index* for scaling up the candidate generator. Specifically, for each token we maintain a set of entities containing that token in their name or in one of their aliases. Thus, for a mention string, we obtain the sets of entities corresponding to each of its constituent tokens, and the final set of candidates is computed by finding an intersection of the previously obtained sets. This ensures scalability and practicality of the candidate generation module.

We follow the aforementioned approach and use the 'name' and 'alias' information (available for 43.7M and 6.6M entities, respectively), described by the `<http://schema.org/name>` and `<http://www.w3.org/2004/02/skos/core#altLabel>` Wikidata relationships respectively, to construct an inverted index of tokens in the mentions to their corresponding Wikidata QIDs. This index is then used to generate candidate entities for each mention, where the entities are sorted based on their degree in Wikidata.

Fixing the maximum number of candidates per mention T. We analyze the effect of the parameter T on the oracle recall for the CoNLL-Val dataset. As portrayed in Fig. 6, the simple string matching based candidate generator achieves an overall oracle recall of 76%, which is improved to 87% by incorporating the use of alias information. Furthermore, it is evident that increasing T provides diminishing marginal gains in the oracle recall: retaining at most 20 candidates per mention already results in an oracle recall of 83%, which is about 95% of the overall oracle recall. With this observation and similar to the existing techniques in the literature (Ganea and Hofmann, 2017; Le and Titov, 2019), we fix T to be 20 for all the datasets. While the candidate generator could be improved by employing smarter tokenization rules, fuzzy string matching, and using word embeddings to capture semantics, we believe the obtained oracle recall of 87% is already quite high, as even with the use of prior information in the presence of annotated data the oracle recall for the aforementioned datasets are in the range of 90 to 95% (Ganea and Hofmann, 2017).

D Experimental Setup

All the experiments were done using code written in Python on an Intel(R) Xeon(R) E5-2680 24-core machine with 2.50GHz CPU, and 256 GB RAM running Linux Ubuntu 20.04. For EIGENTHEMES and WEIGHTED EIGENTHEMES, we use the open source implementation of the singular value decomposition and eigendecomposition, respectively, available in the NumPy linear algebra library (NumPy, 2019). We adapt the Python implementation[5] of τMIL-ND (Le and Titov, 2019) made available by the authors themselves to work with Wikidata as the referent KB instead of Freebase. For details, please see Appendix D.2.

D.1 Code and Datasets

The code, datasets, and all the other resources such as word and entity embeddings, required to reproduce the results reported in this paper are available at https://github.com/epfl-dlab/eigenthemes.

D.2 Note on the Implementation of τMIL-ND

Le and Titov (Le and Titov, 2019) use Freebase as the referent KB. All information pertaining
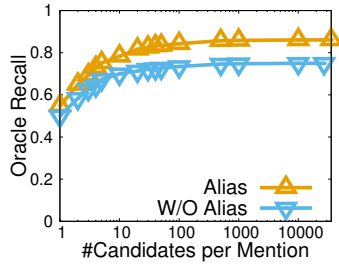
---

[5] https://github.com/lephong/dl4el

Figure 6: Oracle recall as a function of T on the CoNLL-Val dataset (x-axis in log scale).

to candidate entities, such as entity types, name, aliases, description, popularity etc., was extracted from Freebase. To make the implementation of τMIL-ND compliant with our setup, while at the same time ensuring the use of their original resources as much as possible, we use the 'Freebase ID', described by the property `<https://www.wikidata.org/wiki/Property:P646>`, available in the Wikidata dump to map Freebase entities to Wikidata entities. This enabled us to identify a mapping from Freebase to Wikidata for approximately 1.3M entities. After obtaining the mapping, it was straightforward to adapt the implementation of τMIL-ND to work on our preprocessed datasets (Sec. 5.3) using their original resources. We provide modified source files for τMIL-ND's implementation in our GitHub repository https://github.com/epfl-dlab/eigenthemes, which can be simply used to replace the corresponding files at https://github.com/lephong/dl4el to obtain the results for τMIL-ND reported in this paper. Detailed instructions are provided in the README of our GitHub repository.

D.3 Entity Embeddings

The entity embeddings can be computed using the following two approaches:

• **Graph structure:** This approach presents a natural way of learning entity embeddings by building models that preserve the neighborhood of entities in an underlying graph. To this end, we construct a subgraph of the Wikidata knowledge graph by retaining only the edges existent between Wikidata entities that have a mapping in the English Wikipedia. The entity embeddings are then learned by training DEEPWALK (Perozzi et al., 2014) on the resultant graph of 3.7M entities and 20.2M edges.

We also explored recent state of the art techniques NetSMF (Qiu et al., 2019) and LouvainNE (Bhowmick et al., 2020) for learning entity embed-

dings using the Wikidata graph. While NetSMF (Qiu et al., 2019) crashed on our machine (with 256 GB RAM) owing to going out of memory, LouvainNE (Bhowmick et al., 2020) was much worse than DEEPWALK on the entity relatedness task. Specifically, while DEEPWALK obtained an MRR of 0.62, LouvainNE obtained that of 0.44. Note that the authors of LouvainNE did not conduct any experiments on the entity relatedness task, and while it was shown to be better than DEEP-WALK on the node classification task, it cannot be used, since the ability to better capture entity relatedness is a desired property for any embedding technique to be useful in our setting.

• **Textual descriptions:** KGs usually offer a short textual description for each entity and Wikidata is no exception. Entity descriptions have been used to learn entity embeddings in the literature (Yamada et al., 2018). In the same vein, we learn entity embeddings by extracting entity descriptions provided by the `<http://schema.org/description>` Wikidata relationship, and computing the average of WORD2VEC embeddings of the description words.

Moving ahead, we study the utility of embeddings obtained by the aforementioned approaches on our downstream entity linking task. A recent study (Almasian et al., 2019) indicates the ability of node embedding based methods to better capture relatedness when compared to word embedding based methods, which are instead better in capturing similarity. Since the objective of subspace learning is to capture topical relatedness across the gold entities in a document D, the subspace $\mathcal{S}_D$ learned over the candidate embedding matrix $\mathbf{E}_D$ constructed using graph-structure based embeddings should perform better than those obtained using textual descriptions.

Results. We also empirically validate the aforementioned intuition. Figs. 7a and 7b show that there exists a stark difference in the entity linking quality (of around 8 to 12 percentage points in both precision@1 and MRR) for both AVG and EIGEN when using entity embeddings obtained by DEEPWALK than those obtained by average of WORD2VEC embeddings of the entity description words. This observation provides substantial evidence in favor of using DEEPWALK over WORD2VEC embeddings for learning the subspace representation. Therefore, we fix the method for

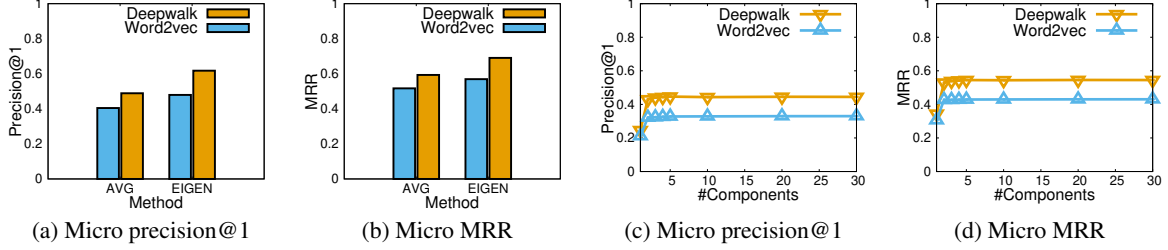(a) Micro precision@1   (b) Micro MRR   (c) Micro precision@1   (d) Micro MRR

Figure 7: Analyzing the effect of (a)-(b) the method for obtaining entity embeddings, and (c)-(d) the number of components used to construct the low-rank subspace using EIGEN, on the quality of entity linking for the CoNLL-Val dataset.

(a) Weights: Precision@1   (b) Weights: MRR   (c) WScheme: Precision@1   (d) WScheme: MRR
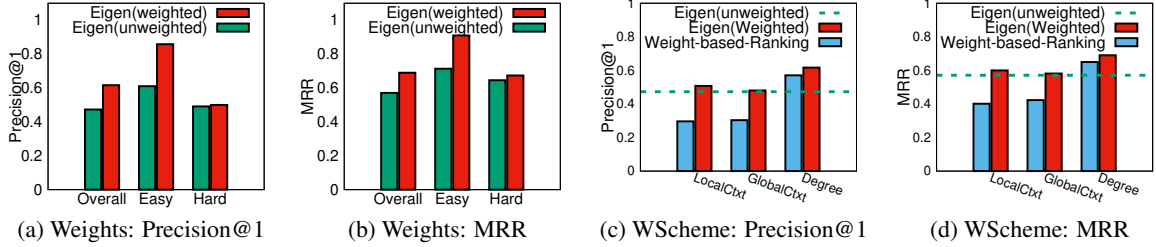
Figure 8: Comparing entity linking quality of unweighted and weighted EIGEN using (a) precision@1, and (b) MRR on the CoNLL-Test dataset. Analyzing the effect of the weighting scheme for constructing the weighted subspace on the entity linking performance of weighted EIGEN using (c) precision@1, and (d) MRR for the CoNLL-Test dataset.

obtaining entity embeddings to the graph-structure based approach using DEEPWALK.

D.4 Weighting Scheme

The weighting scheme $\mathcal{W}$ establishes a descending sorting that relates to the (presumable) probability of each of the candidate entities being linked to the mention under consideration.

Let $\mathcal{W}(e)$ denote the weight for the candidate entity e, which is computed as follows:

$$\mathcal{W}(e) = \text{rank}(e)^{-\delta} \qquad (4)$$

where $\text{rank}(e)$ corresponds to the position of the entity e in the ranking computed. This is closely related to the reciprocal rank, which is a widely used in the information retrieval community. The parameter δ (> 0) controls the importance of the rank position. For large values of δ, the weights will decay very quickly with respect to the rank position, whereas for small values the weights will become more uniform.

A ranking is established for the candidate entities of each mention. Thus, for a document with 5 mentions we have 5 rankings. We explore two different ranking mechanisms.

Ranking based on entity degree. This ranking relies on graph information. It simply takes the order provided—based on entity degree in the knowledge graph—by the candidate generation system, and uses that ranking to compute the weights.

Ranking based on textual coherence. This ranking leverages text signals. For each entity of the knowledge graph, a small description is available. Examples of these descriptions are "British author and humorist" or "Republic in Southwestern Europe" for the entities *Douglas Adams* and *Portugal*, respectively. Similar to (Yamada et al., 2018), we use pre-trained word embeddings to represent entities as the average of the embeddings of the words in their description. We also use these word embeddings to compute a context representation as the average of the embeddings of the words that surround the mention under consideration. We then compute the cosine similarity between the description embedding of each of the candidate entities and the context embedding. The ranking is established based on these similarity scores. For the context embedding we consider a **local** context, which only takes into account words within a window size from the mention, and a **global** context, which takes into account all the words in the document.

Analysis: Do weights enrich the quality of subspaces learned by EIGEN? We empirically assess the utility of the extension (Sec. 4.2) for incorporating weights in the subspace learning step of EIGEN. It is evident from Figs. 8a and 8b that EIGEN (with weights) learns improved subspace representations and consequently, obtains better entity linking performance than EIGEN (without weights). Furthermore, while EIGEN (weighted) is
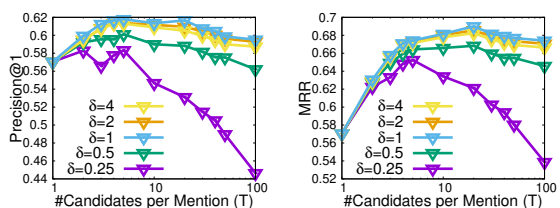
(a) T and δ: Precision@1   (b) T and δ: MRR

Figure 9: Analyzing the effect of the maximum number of candidates per mention (T) on the entity linking performance of EIGEN using (a) precision@1, and (b) MRR for the CoNLL-Test dataset.

only marginally better than EIGEN (unweighted) for 'hard' mentions, it is substantially better (about 25 percentage points) for 'easy' mentions. This is easily explained as the signal derived from the entity-degree-based weighting scheme is biased towards easy mentions by construction (Sec. 5.1). Thus, EIGEN with weighting is expected to obtain an improvement in quality for the easy mentions. In fact, the simultaneous (though marginal) improvement obtained for hard mentions indicates that EIGEN (weighted) successfully congregates the best of both the worlds. This also reinforces the robustness of the EIGENTHEMES framework, as the learned weighted subspace did not get biased towards improving the performance for easy mentions alone. Similar to EIGEN, weights also improve AVG, *i.e.*, AVG (weighted) is considerably better (about 10 points) than AVG (unweighted).

Analysis: Which weighting scheme is the most efficacious? We now empirically assess the strength of different weighting schemes towards improving the EL performance of other techniques, and use EIGEN as a representative technique for the analysis. Results for other techniques portray similar trends and are omitted for the sake of brevity. Figs. 8c and 8d present a comparison of the quality of three signals: (1) LOCAL CTXT, and (2) GLOBAL CTXT (text-based), and (3) entity degree (graph-based); and their impact on the quality of EIGEN (weighted). It is evident from the bar corresponding to 'Weight-based Ranking' in both Figs. 8c and 8d, *i.e.*, the ranking of candidates obtained solely using the weighting schemes, that the text-based signals are much weaker than the graph-based signal. Hence, not surprisingly, the quality of EIGEN (weighted) using text signals is only marginally better than the unweighted EIGEN. On the other hand, the graph-based signal results in a substantial improvement of around 14 percentage points for EIGEN (weighted) over EIGEN (un-

weighted). Thus, we choose the ranking induced by entity degree as the preferred weighting scheme.

Analysis: Effect of δ. Moving ahead, we also analyze the effect of the parameter δ, which reflects the intensity of the weights (Eq. 4) in the subspace learning step of EIGEN. For any fixed value of T, Figs. 9a– 9b portray that increasing δ, and therefore the intensity of weights, results in an improvement at first (until $\delta = 1$), beyond which the performance starts to slowly decline. An in-depth analysis revealed that increasing δ always results in an improvement for easy mentions, however, the performance on hard mentions improves until $\delta = 1$ while starts to decline beyond that. We will see later in Appendix D.5, that DEGREE (by construction) possesses a bias towards easy mentions, and this is exactly what is at play here. To summarize, beyond $\delta = 1$ EIGEN gets biased towards easy mentions and thus, the performance on hard mentions starts to decline, which is also reflected marginally in the overall performance. More elaborate weighting schemes might facilitate an even larger improvement in the performance of EIGEN, however, we leave the design of such schemes for future work.

Analysis: Effect of the number of candidates. While in the experiments, T was fixed to 20 based on the analysis in Appendix C.2, we now analyze the effect of variation in this parameter on the performance of EIGEN. Figs. 9a– 9b show that the performance of EIGEN improves at first (up to 5 candidates per mention), beyond which the performance starts to slowly decline (barring $\delta = 0.25$ for which the decline is steep). This is because increasing the number of candidates beyond a point makes the document embedding matrix noisy, thereby affecting the quality of the learned subspace. However, it can be observed that for larger values of $\delta \geq 1$, EIGEN gracefully handles the noise resulting from an increase in the number of candidates, and we only observe minor effects on the EL performance.

D.5 Results CoNLL: 'Easy' *vs.* 'Hard'

We found that the average number of candidates per mention for hard mentions is much higher than that for easy mentions, and this phenomenon has a stronger deteriorating effect on τMIL-ND than EIGEN. Consequently, the improvement (around 30 points) of EIGEN over τMIL-ND is even more profound for the hard mentions. Note that DEGREE obtains a performance of 100% for easy and 0%

Table 3: Supervised: precision@1 for CoNLL-Test. We run each model 5 times and report the mean and standard deviation.

Context Type	F1+F2+F3	F1+F2+F3+F4	Ceiling
Local	0.848 ± 0.002	**0.864 ± 0.0003**	0.974
Global	0.852 ± 0.002	**0.862 ± 0.002**	0.974

for hard mentions, which is simply due to the way these sets were constructed (*cf.* Sec. 5.1).

E Hyper-parameter Tuning

The only hyperparameter for EIGEN is the number k of components used to construct the low-rank subspace representation of each document D. In this section, we analyze the effect of k on the quality of the entity linking output of EIGEN (unweighted) using the CoNLL-Val dataset. Note that the outcome of the analyses presented in this section is generalizable to EIGEN (weighted) as it is a specialized instance of EIGEN (unweighted). Naturally, we observe similar trends for EIGEN (weighted) and the results are therefore omitted.

It is evident from both Figs. 7c and 7d that increasing the number of components results in an improvement in the entity linking quality measured using precision@1 and MRR respectively. Furthermore, the performance improves monotonically and then plateaus around 10 components, post which there is no considerable improvement. Therefore, we fix the number of components k to 10 for obtaining all the experimental results presented in this paper.

E.1 Extraneous Parameters

We use the parameters prescribed in (Perozzi et al., 2014) to train DEEPWALK, with the number and length of random walks per node set to 80, and the dimensionality of entity embeddings set to 128. Unless stated otherwise, the 'local' textual context of each word in a document D is computed as the average of the word embeddings of its 5 surrounding words (to the left and right), while the 'global' context is computed as the average of the embeddings of all the nouns (Yamada et al., 2016) in D. The Apache OpenNLP tagger (Apache, 2019) was used to detect nouns.

F EIGENTHEMES for Supervised Settings

Having established the superiority of EIGEN-THEMES on the CoNLL-Test dataset for the unsupervised setting, we now assess their applicability

in settings where annotated data is available. Note that the goal of this experiment is not to design state of the art supervised entity linking systems. Rather, the focus is to showcase the capability of EIGENTHEMES to improve them.

We first describe the setup, which is slightly different from the unsupervised case. For candidate generation, we use the mention–entity association dictionaries made available publicly by Ganea and Hofmann (Ganea and Hofmann, 2017). Similar to the unsupervised case, T was fixed to 20. In addition to facilitating candidate generation, this dictionary allows us to infer the prior probability $P(e|m)$. The availability of annotated data also allows for the learning of aligned word and entity embeddings. We employ 300-dimensional pre-trained, aligned entity and word embeddings (Ganea and Hofmann, 2017). Inspired by existing state of the art methods for supervised entity linking (Ganea and Hofmann, 2017; Yamada et al., 2016), for each mention–candidate pair, we use the following features to train a supervised model: (F1) the prior probability; (F2) textual context score: obtained by computing the cosine similarity between the candidate entity embedding and the (local or global) context embedding (Sec. D.4); (F3) global coherence score: obtained by computing the cosine similarity between the candidate entity embedding and the global entity context (Yamada et al., 2016); and (F4) the EIGEN score (Sec. 4.3).

We employ random forests (Breiman, 2001) as a point-wise learning-to-rank technique to appropriately combine the contribution of the aforementioned features. We rely on the publicly available implementation of random forests in scikit-learn (Pedregosa et al., 2011). The model is trained using the CoNLL-Train set, while the CoNLL-Test set is used to evaluate the entity linking quality. The results are presented in Table 3. Adding the EIGEN score as a feature into the supervised model results in an improvement of 1 to 2 percentage points. This result portrays the ability of our method to improve existing supervised entity linking systems. Furthermore, it validates the importance of an appropriate collective disambiguation method, even in the presence of other local scores such as prior information and contextual cues. Lastly, it is worth highlighting that the bulk of the improvement (more than 5 percentage points) is obtained for the hard mentions, which is both required and important as existing features already facilitate obtaining the performance of close to 100% on easy mentions.