ActiveEA: Active Learning for Neural Entity Alignment

Bing Liu[1,✉], Harrisen Scells[1], Guido Zuccon[1], Wen Hua[1], Genghong Zhao[2]
[1]The University of Queensland, Australia
[2]Neusoft Research of Intelligent Healthcare Technology, Co. Ltd., China
{bing.liu, h.scells, g.zuccon, w.hua}@uq.edu.au
zhaogenghong@neusoft.com

Abstract

Entity Alignment (EA) aims to match equivalent entities across different Knowledge Graphs (KGs) and is an essential step of KG fusion. Current mainstream methods – neural EA models – rely on training with seed alignment, i.e., a set of pre-aligned entity pairs which are very costly to annotate. In this paper, we devise a novel Active Learning (AL) framework for neural EA, aiming to create highly informative seed alignment to obtain more effective EA models with less annotation cost. Our framework tackles two main challenges encountered when applying AL to EA:

(1) How to exploit dependencies between entities within the AL strategy. Most AL strategies assume that the data instances to sample are independent and identically distributed. However, entities in KGs are related. To address this challenge, we propose a structure-aware uncertainty sampling strategy that can measure the uncertainty of each entity as well as its impact on its neighbour entities in the KG.

(2) How to recognise entities that appear in one KG but not in the other KG (i.e., *bachelors*). Identifying bachelors would likely save annotation budget. To address this challenge, we devise a bachelor recognizer paying attention to alleviate the effect of sampling bias.

Empirical results show that our proposed AL strategy can significantly improve sampling quality with good generality across different datasets, EA models and amount of bachelors.

1 Introduction

Knowledge Graphs (KGs) store entities and their relationships with a graph structure and are used as knowledge drivers in many applications (Ji et al., 2020). Existing KGs are often incomplete but complementary to each other. A popular approach used to tackle this problem is KG fusion, which attempts to combine several KGs into a single, comprehensive one. Entity Alignment (EA) is an essential step for KG fusion: it identifies equivalent entities across different KGs, supporting the unification of their complementary knowledge. For example, in Fig. 1 *Donald Trump* and *US* in the first KG correspond to *D.J. Trump* and *America* respectively in the second KG. By aligning them, the political and business knowledge about *Donald Trump* can be integrated within one KG.

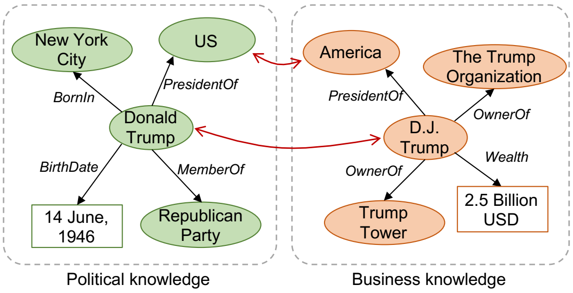

Figure 1: An example of Entity Alignment.

Neural models (Chen et al., 2017, 2018; Wang et al., 2018; Cao et al., 2019) are the current state-of-the-art in EA and are capable of matching entities in an end-to-end manner. Typically, these neural EA models rely on a seed alignment as training data which is very labour-intensive to annotate. However, previous EA research has assumed the availability of such seed alignment and ignored the cost involved with their annotation. In this paper, we seek to reduce the cost of annotating seed alignment data, by investigating methods capable of selecting the most informative entities for labelling so as to obtain the best EA model with the least annotation cost: we do so using Active Learning. Active Learning (AL) (Aggarwal et al., 2014) is a Machine Learning (ML) paradigm where the annotation of data and the training of a model are performed iteratively so that the sampled data is highly informative for training the model. Though many general AL strategies have been proposed (Settles, 2012; Ren et al., 2020), there are some unique challenges in applying AL to EA.

The first challenge is **how to exploit the dependencies between entities**. In the EA task, neighbouring entities (context) in the KGs naturally affect each other. For example, in the two KGs of Fig. 1, we can infer *US* corresponds to *America* if we already know that *Donald Trump* and *D.J. Trump* refer to the same person: this is because a single person can only be the president of one country. Therefore, when we estimate the value of annotating an entity, we should consider its impact on its context in the KG. Most AL strategies assume data instances are independent, identically distributed and cannot capture dependencies between entities (Aggarwal et al., 2014). In addition, neural EA models exploit the structure of KGs in different and implicit ways (Sun et al., 2020b). It is not easy to find a general way of measuring the effect of entities on others.

The second challenge is **how to recognize the entities in a KG that do not have a counterpart in the other KG** (i.e., *bachelors*). In the first KG of Fig. 1, *Donald Trump* and *US* are matchable entities while *New York City* and *Republican Party* are bachelors. Selecting bachelors to annotate will not lead to any aligned entity pair. The impacts of recognizing bachelors are twofold:

1. From the perspective of data annotation, recognizing bachelors would automatically save annotation budget (because annotators will try to seek a corresponding entity for some time before giving up) and allow annotators to put their effort in labelling matchable entities. This is particularly important for the existing neural EA models, which *only* consider matchable entities for training: thus selecting bachelors in these cases is a waste of annotation budget.
2. From the perspective of EA, bachelor recognition remedies the limitation of existing EA models that assume all entities to align are matchable, and would enable them to be better used in practice (i.e., real-life KGs where bachelors are popular).

To address these challenges, we propose a novel AL framework for EA. Our framework follows the typical AL process: entities are sampled iteratively, and in each iteration a batch of entities with the highest acquisition scores are selected. Our novel acquisition function consists of two components: a structure-aware uncertainty measurement module and a bachelor recognizer. The structure-aware uncertainty can reflect the uncertainty of a single entity as well as the influence of that entity in the context of the KG, i.e., how many uncertainties it can help its neighbours eliminate. In addition, we design a bachelor recognizer, based on Graph Convolutional Networks (GCNs). Because the bachelor recognizer is trained with the sampled data and used to predict the remaining data, it may suffer from bias (w.r.t. the preference of sampling strategy) of these two groups of data. We apply model ensembling to alleviate this problem.

Our major contributions in this paper are:

1. A novel AL framework for neural EA, which can produce more informative data for training EA models while reducing the labour cost involved in annotation. To our knowledge, this is the first AL framework for neural EA.
2. A structure-aware uncertainty sampling strategy, which models uncertainty sampling and the relation between entities in a single AL strategy.
3. An investigation of bachelor recognition, which can reduce the cost of data annotation and remedy the defect of existing EA models.
4. Extensive experimental results that show our proposed AL strategy can significantly improve the quality of data sampling and has good generality across different datasets, EA models, and bachelor quantities.

2 Background

2.1 Entity Alignment

Entity alignment is typically performed between two KGs $\mathcal{G}^1$ and $\mathcal{G}^2$, whose entity sets are denoted as $\mathcal{E}^1$ and $\mathcal{E}^2$ respectively. The goal of EA is to find the equivalent entity pairs $\mathcal{A} = \{(e^1, e^2) \in \mathcal{E}^1 \times \mathcal{E}^2 | e^1 \sim e^2\}$, where $\sim$ denotes an equivalence relationship and is usually assumed to be a one-to-one mapping. In supervised and semi-supervised models, a subset of the alignment $\mathcal{A}^{seed} \subset \mathcal{A}$, called seed alignment, are annotated manually beforehand and used as training data. The remaining alignment form the test set $\mathcal{A}^{test} = \mathcal{A} \setminus \mathcal{A}^{seed}$. The core of an EA model F is a scoring function $F(e^1, e^2)$, which takes two entities as input and returns a score for how likely they match. The effectiveness of an EA model is essentially determined by $\mathcal{A}^{seed}$ and we thus denote it as $m(\mathcal{A}^{seed})$.

2.2 Active Learning

An AL framework consists of two components: (1) an *oracle* (annotation expert), which provides labels for the *queries* (data instances to label), and

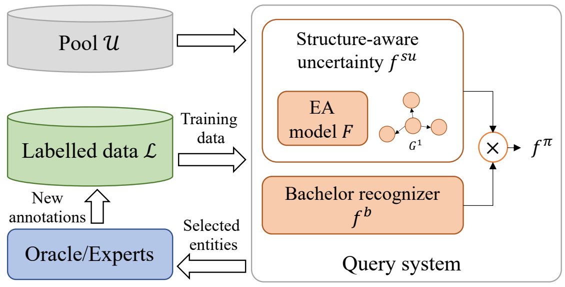

Figure 2: Overview of ActiveEA.

(2) a *query* system, which selects the most informative data instances as queries. In pool-based scenario, there is a pool of unlabelled data $\mathcal{U}$. Given a budget B, some instances $\mathcal{U}_{\pi,B}$ are selected from the pool following a strategy π and sent to the experts to annotate, who produce a training set $\mathcal{L}_{\pi,B}$. We train the model on $\mathcal{L}_{\pi,B}$ and the effectiveness $m(\mathcal{L}_{\pi,B})$ of the obtained model reflects how good the strategy π is. The goal is to design an optimal strategy π_* such that $\pi_* = \text{argmax}_\pi m(\mathcal{L}_{\pi,B})$.

3 ActiveEA: Active Entity Alignment

3.1 Problem Definition

Given two KGs $\mathcal{G}^1$, $\mathcal{G}^2$ with entity sets $\mathcal{E}^1$, $\mathcal{E}^2$, an EA model F, a budget B, the AL strategy π is applied to select a set of entities $\mathcal{U}_{\pi,B}$ so that the annotators label the counterpart entities to obtain the labelled data $\mathcal{L}_{\pi,B}$. $\mathcal{L}_{\pi,B}$ consists of annotations of matchable entities $\mathcal{L}^+_{\pi,B}$, which form the seed alignment $\mathcal{A}^{seed}_{\pi,B}$, and bachelors $\mathcal{L}^-_{\pi,B}$. We measure the effectiveness $m(\mathcal{A}^{seed}_{\pi,B})$ of the AL strategy π by training the EA model on $\mathcal{A}^{seed}_{\pi,B}$ and then evaluating it with $\mathcal{A}^{test}_{\pi,B} = \mathcal{A} \setminus \mathcal{A}^{seed}_{\pi,B}$. Our goal is to design an optimal entity sampling strategy π_* so that $\pi_* = \text{argmax}_\pi m(\mathcal{A}^{seed}_{\pi,B})$.

In our annotation setting, we select entities from one KG and then let the annotators identify their counterparts from the other KG. Under this setting, we assume the pool of unlabelled entities is initialized with $\mathcal{U} = \mathcal{E}^1$. The labelled data will be like $\mathcal{L}^+_{\pi,B} = \{(e^1 \in \mathcal{E}^1, e^2 \in \mathcal{E}^2)\}$ and $\mathcal{L}^-_{\pi,B} = \{(e^1 \in \mathcal{E}^1, null)\}$.

3.2 Framework Overview

The whole annotation process, as shown in Fig. 2, is carried out iteratively. In each iteration, the query system selects N entities from $\mathcal{U}$ and sends them to the annotators. The query system includes (1) a structure-aware uncertainty measurement module f^{su}, which combines uncertainty sampling with the structure information of the KGs, and (2) a bachelor recognizer f^b, which helps avoid selecting bachelor entities. The final acquisition f^π used to select which entities to annotate is obtained by combining the outputs of these two modules. After the annotators assign the ground-truth counterparts to the selected entities, the new annotations are added to the labelled data $\mathcal{L}$. With the updated $\mathcal{L}$, the query system updates the EA model and the bachelor recognizer. This process repeats until no budget remains. To simplify the presentation, we omit the sampling iteration when explaining the details.

3.3 Structure-aware Uncertainty Sampling

We define the influence of an entity on its context as the amount of uncertainties it can help its neighbours remove. As such, we formulate the structure-aware uncertainty f^{su} as

$$\begin{aligned} f^{su}(e^1_i) = \alpha \sum_{e^1_i \to e^1_j, e^1_j \in \mathcal{N}^{out}_i} w_{ij} f^{su}(e^1_j) \\ + (1-\alpha) \frac{f^u(e^1_i)}{\sum_{e^1 \in \mathcal{E}^1} f^u(e^1)}, \end{aligned} \tag{1}$$

where $\mathcal{N}^{out}_i$ is the outbound neighbours of entity e^1_i (i.e. the entities referred to by e^1_i) and w_{ij} measures the extent to which e^1_i can help e^1_j eliminate uncertainty. The parameter α controls the trade-off between the impact of entity e^1_i on its context (first term in the equation) and the normalized uncertainty (second item). Function $f^u(e^1)$ refers to the margin-based uncertainty of an entity. For each entity e^1, the EA model can return the matching scores $F(e^1, e^2)$ with all unaligned entities e^2 in $\mathcal{G}^2$. Since these scores in existing works are not probabilities, we exploit the margin-based uncertainty measure for convenience, outlined in Eq. 2:

$$f^u(e^1) = -\left(F(e^1, e^2_*) - F(e^1, e^2_{**})\right) \tag{2}$$

where $F(e^1, e^2_*)$ and $F(e^1, e^2_{**})$ are the highest and second highest matching scores respectively. A large margin represents a small uncertainty.

For each entity e^1_j, we assume its inbound neighbours can help it clear all uncertainty. Then, we have $\sum_{e^1_i \to e^1_j, e^1_i \in \mathcal{N}^{in}_j} w_{ij} = 1$, where $\mathcal{N}^{in}_j$ is the inbound neighbour set of e^1_j. In this work, we assume all inbound neighbours have the same impact on e^1_j. In this case, $w_{ij} = \frac{1}{\text{degree}(e^1_j)}$, where $\text{degree}(\cdot)$ returns the in-degree of an entity.

Using matrix notion, Eq. 1 can be rewritten as

$$\mathbf{f}^{su} = \alpha \mathbf{W} \mathbf{f}^{su} + (1-\alpha) \frac{\mathbf{f}^u}{|\mathbf{f}^u|}$$

where $\mathbf{f}^{su}$ is the vector of structure-aware uncertainties, $\mathbf{f}^{u}$ is the vector of uncertainties, and $\mathbf{W}$ is a matrix encoding influence between entities, i.e., $w_{ij} > 0$ if e_i^1 is linked to e_j^1, otherwise 0.

As $\mathbf{W}$ is a stochastic matrix (Gagniuc, 2017), we solve Eq. 1 iteratively, which can be viewed as the power iteration method (Franceschet, 2011), similar to *Pagerank* (Brin and Page, 1998). Specifically, we initialize the structure-aware uncertainty vector as $\mathbf{f}_0^{su} = \mathbf{f}^u$. Then we update $\mathbf{f}_t^{su}$ iteratively:

$$\mathbf{f}_t^{su} = \alpha \mathbf{W} \mathbf{f}_{t-1}^{su} + (1-\alpha)\frac{\mathbf{f}^u}{|\mathbf{f}^u|}, t = 1, 2, 3, ...$$

The computation ends when $|\mathbf{f}_t^{su} - \mathbf{f}_{t-1}^{su}| < \epsilon$.

3.4 Bachelor Recognizer

The bachelor recognizer is formulated as a binary classifier, which is trained with the labelled data and used to predict the unlabelled data. One challenge faced here is the bias between the labelled data and the unlabelled data caused by the sampling strategy (since it is not random sampling). We alleviate this issue with a model ensemble.

3.4.1 Model Structure

We apply two GCNs (Kipf and Welling, 2017; Hamilton et al., 2017) as the encoders to get the entity embeddings $\mathbf{H}^1 = \mathbf{GCN}^1(\mathcal{G}^1), \mathbf{H}^2 = \mathbf{GCN}^2(\mathcal{G}^2)$, where each row in $\mathbf{H}^1$ or $\mathbf{H}^2$ corresponds to a vector representation of a particular entity. The two GCN encoders share the same structure but have separate parameters. With each GCN encoder, each entity e_i is first assigned a vector representation $\mathbf{h}_i^{(0)}$. Then contextual features of each entity are extracted:

$$\mathbf{h}_i^{(l)} = \mathrm{norm}(\sigma(\sum_{j \in \mathcal{N}_i \cup \{i\}} \mathbf{V}^{(l)} \mathbf{h}_j^{(l-1)} + \mathbf{b}^{(l)})),$$

where l is the layer index, $\mathcal{N}_i$ is the neighbouring entities of entity e_i, and σ is the activation function, $\mathrm{norm}(\cdot)$ is a normalization function, and $\mathbf{V}^{(l)}, \mathbf{b}^{(l)}$ are the parameters in the l-th layer. The representations of each entity e_i obtained in all GCN layers are concatenated into a single representation: $\mathbf{h}_i = \mathrm{concat}(\mathbf{h}_i^{(0)}, \mathbf{h}_i^{(1)}, ..., \mathbf{h}_i^{(L)})$, where L is the number of GCN layers.

After getting the representations of entities, we compute the similarities of each entity in $\mathcal{E}^1$ with all entities in $\mathcal{E}^2$ ($\mathbf{S} = \mathbf{H}^1 \cdot \mathbf{H}^{2^T}$) and obtain its corresponding maximum matching score as in $f^s(e_i^1) = \max(\mathbf{S}_{i,:})$. The entity e_i^1 whose maximum matching score is greater than a threshold γ is considered to be a matchable entity as in $f^b(e_i^1) = \mathbb{1}_{f^s(e_i^1) > \gamma}$, otherwise a bachelor.

3.4.2 Learning

In each sampling iteration, we train the bachelor recognizer with existing annotated data $\mathcal{L}$ containing matchable entities $\mathcal{L}^+$ and bachelors $\mathcal{L}^-$. Furthermore, $\mathcal{L}$ is divided into a training set $\mathcal{L}^t$ and a validation set $\mathcal{L}^v$.

We optimize the parameters, including $\{\mathbf{V}^{(l)}, \mathbf{b}^{(l)}\}_{1 \le l \le L}$ of each GCN encoder and the threshold γ, in two phases, sharing similar idea with supervised contrastive learning (Khosla et al., 2020). In the first phase, we optimize the scoring function f^s by minimizing the constrastive loss shown in Eq. 3.

$$\begin{aligned} loss = &\sum_{(e_i^1, e_j^2) \in \mathcal{L}^{t,+}} \| \mathbf{h}_i^1 - \mathbf{h}_j^2 \| \\ &+ \beta \sum_{(e_{i'}^1, e_{j'}^2) \in \mathcal{L}^{t,neg}} [\lambda - \| \mathbf{h}_{i'}^1 - \mathbf{h}_{j'}^2 \|]_+ \end{aligned} \tag{3}$$

Here, β is a balance factor, and $[\cdot]_+$ is $\max(0, \cdot)$, and $\mathcal{L}^{t,neg}$ is the set of negative samples generated by negative sampling (Sun et al., 2018). For a given pre-aligned entity pair in $\mathcal{L}^+$, each entity of it is substituted for N^{neg} times. The distance of negative samples is expected to be larger than the margin λ. In the second phase, we freeze the trained f^s and optimize γ for f^b. It is easy to optimize γ, e.g. by simple grid search, so that f^b can achieve the highest performance on $\mathcal{L}^v$ (denoted as $q(f^s, \gamma, \mathcal{L}^v)$) using:

$$\gamma^* = \mathrm{argmax}_\gamma q(f^s, \gamma, \mathcal{L}^v).$$

3.4.3 Model Ensemble for Sampling Bias

The sampled data may be biased, since they have been preferred by the sampling strategy rather than selected randomly. As a result, even if the bachelor recognizer is well trained with the sampled data it may perform poorly on data yet to sample. We apply a model ensemble to alleviate this problem. Specifically, we divide the $\mathcal{L}$ into K subsets evenly. Then we apply K-fold cross-validation to train K scoring functions $\{f_1^s, ..., f_K^s\}$, each time using $K-1$ subsets as the training set and the left out portion as validation set. Afterwards, we search for an effective γ threshold:

$$\gamma^* = \mathrm{argmax}_\gamma \frac{1}{K} \sum_{1 \le k \le K} q(f_k^s, \gamma, \mathcal{L}_k^v)$$

At inference, we ensemble by averaging the K scoring functions f_k^s to form the final scoring function f^s as in Eq. 4 and base f^b on it.

$$f^s(e_i^1) = \frac{1}{K} \sum_{1 \leq k \leq K} f_k^s(e_i^1) \tag{4}$$

3.5 Final Acquisition Function

We combine our structure-aware uncertainty sampling with the bachelor recognizer to form the final acquisition function:

$$f^{\pi}(e_i^1) = f^{su}(e_i^1) f^b(e_i^1)$$

4 Experimental Setup

4.1 Sampling Strategies

We construct several baselines for comparison:

rand random sampling used by existing EA works.

degree selects entities with high degrees.

pagerank (Brin and Page, 1998) measures the centrality of entities by considering their degrees as well as the importance of its neighbours.

betweenness (Freeman, 1977) refers to the number of shortest paths passing through an entity.

uncertainty sampling selects entities that the current EA model cannot predict with confidence. Note that in this work we measure uncertainty using Eq. 2 for fair comparison.

degree, *pagerank* and *betweenness* are purely topology-based and do not consider the current EA model. On the contrary, *uncertainty* is fully based on the current EA model without being able to capture the structure information of KG. We compare both our structure-aware uncertainty sampling (***struct_uncert***) and the full framework ***ActiveEA*** with the baselines listed above. We also examine the effect of ***Bayesian Transformation***, which aims to make deep neural models represent uncertainty more accurately (Gal et al., 2017).

4.2 EA Models

We apply our ActiveEA framework to three different EA models, which are a representative spread of neural EA models and varied in KG encoding, considered information and training method (Liu et al., 2020; Sun et al., 2018):

BootEA (Sun et al., 2018) encodes the KGs with the translation model (Bordes et al., 2013), exploits the structure of KGs, and uses self-training.

Alinet (Sun et al., 2020a) also exploits the structure of KGs but with a GCN-based KG encoder, and is trained in a supervised manner.

RDGCN (Wu et al., 2019) trains a GCN in a supervised manner, as Alinet, but it can incorporate entities' attributes.

Our implementations and parameter settings of the models rely on OpenEA[1] (Sun et al., 2020b).

4.3 Datasets

We use three different datasets: D-W-15K V1 (*DW*), EN-DE-15K V1 (*ENDE*), and EN-FR-100K V1 (*ENFR*), obtained from OpenEA (Sun et al., 2020b). Each dataset contains two KGs and equivalent entity pairs. The KGs used in these datasets were sampled from real KGs, i.e. *DBpedia* (Lehmann et al., 2015), *Wikidata* (Vrandecic and Krötzsch, 2014), and *YAGO* (Rebele et al., 2016), which are widely used in EA community. These datasets differ in terms of KG sources, languages, sizes, etc. We refer the reader to Sun et al. (2020b) for more details.

Existing work on EA assumes all entities in the KGs are matchable, thus only sampling entities with counterparts when producing the datasets. For investigating the influence of bachelors on AL strategies, we synthetically modify the datasets by excluding a portion of entities from the second KG.

4.4 Evaluation Metrics

We use Hit@1 as the primary evaluation measure of the EA models. To get an overall evaluation of one AL strategy across different sized budgets, we plot the curve of a EA model's effectiveness with respect to the proportion of annotated entities, and calculate the Area Under the Curve (AUC).

4.5 Parameter Settings

We set $\alpha = 0.1$, $\epsilon = 1e^{-6}$ for the structure-aware uncertainty. We use $L = 1$ GCN layer for our bachelor recognizer with 500 input and 400 output dimensions. We set $K = 5$ for its model ensemble and $\lambda = 1.5$, $\beta = 0.1$, $N^{neg} = 10$ for its training. The sampling batch size is set to $N = 100$ for 15K data and $N = 1000$ for 100K data.

4.6 Reproducibility Details

Our experiments are run on a GPU cluster. We allocate 50G memory and one 32GB nVidia Tesla V100 GPU for each job on 15K data, and 100G memory for each job on 100K data. The training and evaluation of *ActiveEA* take approximately 3h with Alinet on 15K data, 10h with BootEA on 15K

[1] https://github.com/nju-websoft/OpenEA

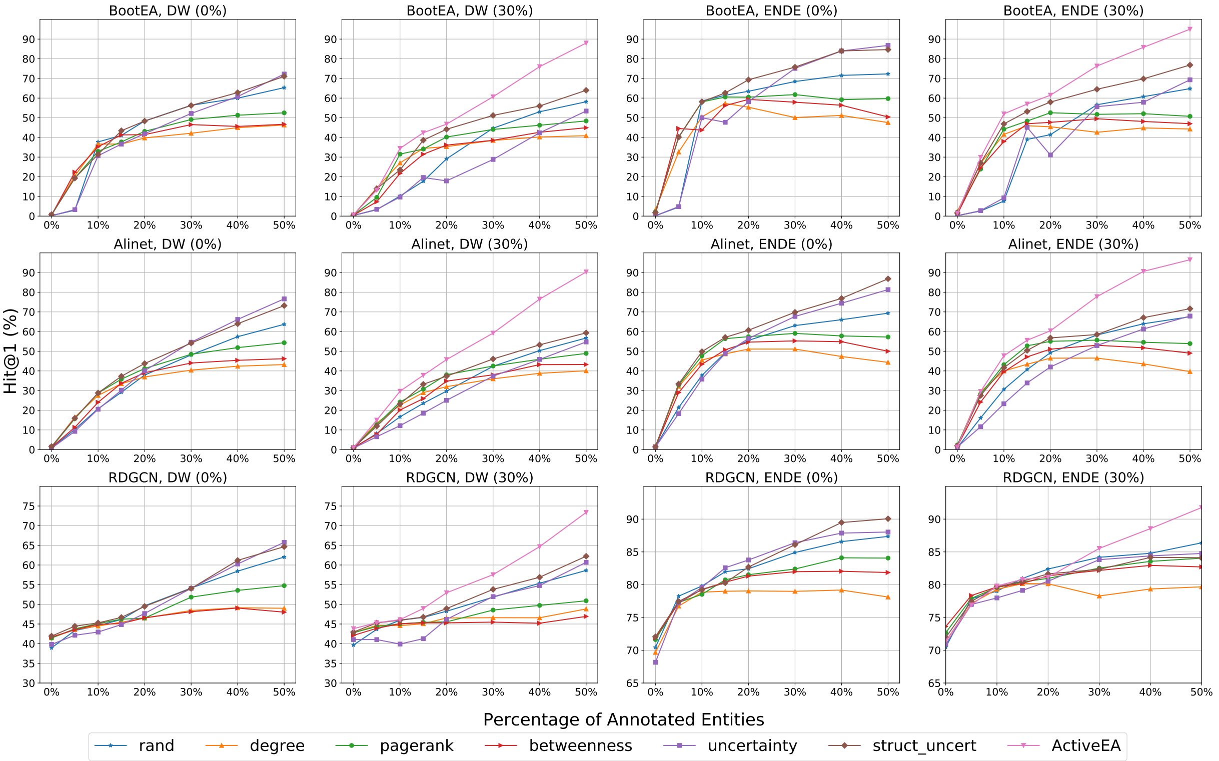

Figure 3: HIT@1 of sampling strategies for all EA models on DW and ENDE, as annotation portion increases. Top row shows experiments that do not include bachelors; bottom row shows experiments that include 30% bachelors. *ActiveEA* is equivalent to *struct_uncert* in absence of bachelors, and is thus shown only for the second row.

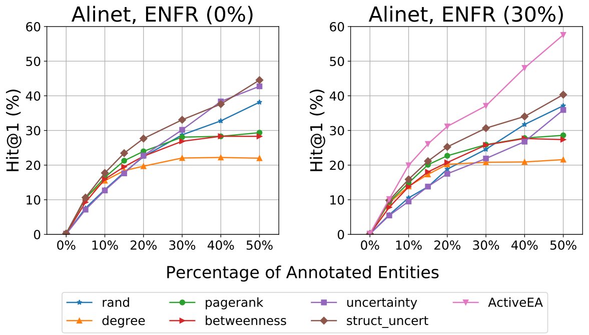

Figure 4: Hit@1 for all sampling strategies on the Alinet EA model on ENFR. Left shows experiments without bachelors, right shows with 30% bachelors.

data, 10h with RDGCN on 15K data, and 48h with Alinet on 100K data. Most baseline strategies take less time than *ActiveEA* on the same dataset except *betweenness* on 100K data, which takes more than 48h. We apply grid search for setting α and N (shown in Sec. 5.4). Hyper-parameters of the bachelor recognizer are chosen by referring the settings of OpenEA and our manual trials. Code and datasets are available at https://github.com/UQ-Neusoft-Health-Data-Science/ActiveEA.

5 Experimental Results

5.1 Comparison with Baselines

Fig. 3 presents the overall performance of each strategy with three EA models on two datasets, each of which we also synthetically modify to include 30% bachelors. We also report the AUC@0.5 values of these curves in Tab. 1. *ActiveEA* degenerates into *struct_uncert* when there is no bachelor.

Random Sampling. Random sampling usually performs poorly when the annotation proportion is small, while it becomes more competitive when the amount of annotations increases. But for most annotation proportions, random sampling exhibits a large gap in performance compared to the best method. This observation highlights the need to investigate data selection for EA.

Topology-based Strategies. The topology-based strategies are effective when few annotations are provided, e.g., $< 20\%$. However, once annotations increase, the effectiveness of topology-based strategies is often worse than random sampling. This may be because these strategies suffer more from the bias between the training set and test set. Therefore, only considering the structural information of KGs has considerable drawbacks for EA.

Uncertainty Sampling. On the contrary, the un-

Strategy	BootEA				AliNet				RDGCN			
	DW (0%)	DW (30%)	ENDE (0%)	ENDE (30%)	DW (0%)	DW (30%)	ENDE (0%)	ENDE (30%)	DW (0%)	DW (30%)	ENDE (0%)	ENDE (30%)
rand	23.5^n	17.0	28.1	21.3	19.4	16.7	26.0	23.7	25.8	25.0	41.3^n	41.0
degree	19.5	16.0	24.0	20.0	17.1	15.2	22.2	20.5	23.3	22.9	39.1	39.4
pagerank	22.3	18.3	27.6	23.0	19.9	17.3	25.8	24.1	24.5	23.9	40.5	40.6
betweenness	20.5	16.3	26.1	21.1	17.8	15.6	23.7	22.3	23.2	22.7	40.2	40.3
uncertainty	23.9	16.1	29.8	21.2	21.6	15.4	28.2	22.2	24.7	23.9	40.9^n	40.5
struct_uncert	**26.3**	20.8	**33.6**	27.4	**23.1**	19.1	**30.6**	26.8	**26.5**	25.6	**41.9**	41.0
ActiveEA		**26.7**		**31.5**		**25.7**		**32.8**		**28.1**		**42.3**

Table 1: Overall performance (AUC@0.5 (%)) for each sampling strategy. The highest performing strategy in each column is indicated in bold. We run each strategy 5 times; most results for *ActiveEA* show statistically significant differences over other methods (paired t-test with Bonferroni correction, $p < 0.05$), except the few cells indicated by ^{n}.

certainty sampling strategy performs poorly when the proportion of annotations is small but improves after several annotations have been accumulated. One reason for this is that neural EA models cannot learn useful patterns with a small number of annotations. On datasets with bachelors, uncertainty sampling always performs worse than random sampling. Thus, it is clear that uncertainty sampling cannot be applied directly to EA.

Structure-aware Uncertainty Sampling. Structure-aware uncertainty is effective across all annotation proportions. One reason for this is that it combines the advantages of both topology-based strategies and uncertainty sampling. This is essential for AL as it is impossible to predict the amount of annotations required for new datasets.

ActiveEA. *ActiveEA*, which enhances structure-aware sampling with a bachelor recognizer, greatly improves EA when KGs contain bachelors.

5.1.1 Generality

The structure-aware uncertainty sampling mostly outperforms the baselines, while *ActiveEA* performs even better in almost all cases. *ActiveEA* also demonstrates generality across datasets, EA models, and bachelor proportions.

When the dataset has no bachelors, our uncertainty-aware sampling is exceeded by uncertainty sampling in few large-budget cases. However, the real-world datasets always have bachelors. In this case, our structure-aware uncertainty shows more obvious advantages.

In addition, the strategies are less distinguishable when applied to RDGCN. The reason is that RDGCN exploits the name of entities for pre-alignment and thus all strategies achieve good performance from the start.

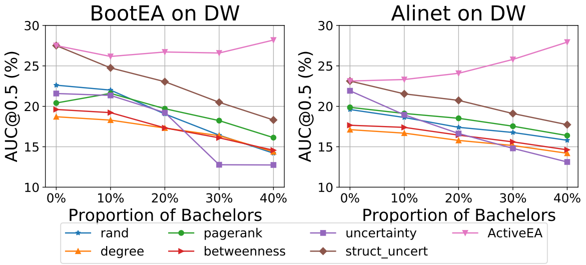

Figure 5: Comparison demonstrating the effect of bachelors (0% – 40%) on the BootEA and Alinet models.

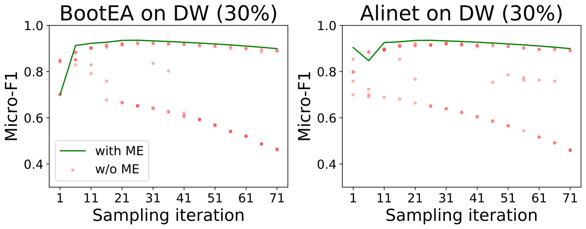

Figure 6: Comparison demonstrating the effectiveness of the bachelor recognizer and the effect of the model ensemble (ME) on BootEA and Alinet.

To assess the generality across datasets of different sizes, we evaluate the sampling strategies with Alinet using ENFR (100K entities), which is larger than DW and ENDE (15K entities). We choose Alinet because it is more scalable than BootEA and RDGCN (Zhao et al., 2020). Fig. 4 presents comparable results to the 15K datasets.

5.2 Effect of Bachelors

To investigate the effect of bachelors, we removed different amounts of entities randomly (each larger sample contains the subset from earlier samples) from $\mathcal{G}^2$ so that $\mathcal{G}^1$ had different percentages of bachelors. Fig. 5 shows the results of applying all strategies to these datasets. We further make the

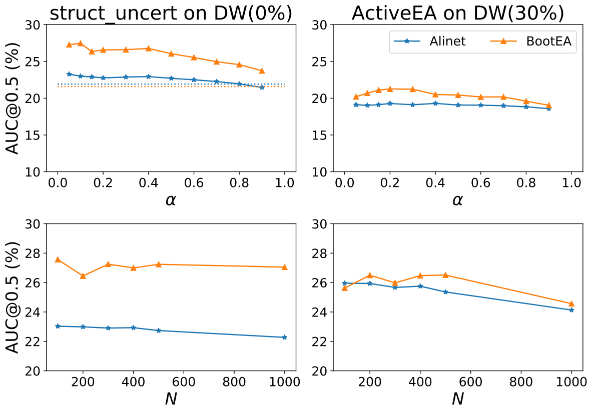

Figure 7: Comparison demonstrating the effects different parameters have on our sampling strategies.

following four observations:

1. The performance of all strategies except *ActiveEA* decrease as bachelors increase. How to avoid selecting bachelors is an important issue in designing AL strategies for EA.

2. Among all strategies, uncertainty sampling is affected the most, while topology-based methods are only marginally affected.

3. Our structure-aware uncertainty outperforms the baselines in all tested bachelor proportions.

4. *ActiveEA* increases performance as the proportion of bachelors increases. The reason is: if $\mathcal{G}^1$ is fixed and the bachelors can be recognized successfully, a certain budget can lead to larger ratio of annotated matchable entities in datasets with more bachelors than in those with less bachelors.

5.3 Effectiveness of Bachelor Recognizer

Fig. 6 shows the effectiveness of our bachelor recognizer in the sampling process and the effect of model ensemble. The green curve shows the Micro-F1 score of our bachelor recognizer using the model ensemble. Our bachelor recognizer achieves high effectiveness from the start of sampling, where there are few annotations. Each red dot represents the performance of the bachelor recognizer trained with a certain data partition without using the model ensemble. Performance varied because of the bias problem. Therefore, our model ensemble makes the trained model obtain high and stable performance.

5.4 Sensitivity of Parameters

To investigate the sensitivity of parameters, we ran our strategy with AliNet and BootEA on two DW variants with bachelor proportions of 0% and 30%.

The sensitivity w.r.t. α is shown in the top row of Fig. 7. We observe that our method is not sensitive to α. The effectiveness fluctuates when $\alpha < 0.5$, and decreases when $\alpha > 0.5$. This indicates uncertainty is more informative than structural information. When $\alpha = 0$, our struct_uncert degenerates to uncertainty sampling (Eq. 2). In the upper left plot, we show the corresponding performance with dotted lines. Under most settings of α, the struct_uncert is much better than uncertainty sampling. This means that introducing structure information is beneficial.

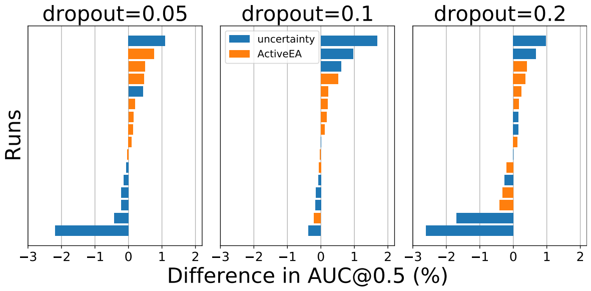

Figure 8: Effect of Bayesian Transformation on *uncertainty* and *ActiveEA* across the DW and ENDE datasets and different bachelor percentages.

The bottom row of Fig. 7 shows the effect of sampling batch size N. The overall trend is that larger batch sizes decrease performance. This observation confirms the intuition that more frequent updates to the EA model lead to more precise uncertainty. Therefore, the choice of value of sampling batch size is a matter of trade-off between computation cost and sampling quality.

5.5 Examination of Bayesian Transformation

We enhanced the uncertainty sampling and *ActiveEA* with Bayesian Transformation, implemented with Monte Carlo (MC) dropout, and applied them to Alinet and RDGCN on DW and ENDE as in Sec. 5.1. Fig. 8 shows improvements with different settings of MC dropout rate. We find (1) the variation of effects on *uncertainty* sampling is greater than that on *ActiveEA*; (2) Bayesian Transformation with small dropout (e.g., 0.05) results in slight improvements to *ActiveEA* in most cases.

6 Related Works

Entity Alignment. Entity Alignment refers to the matching of entities across different KGs that refer to the same real-world object. Compared with Entity Resolution (Mudgal et al., 2018), which matches duplicate entities in relational data, EA deals with graph data and emphasizes on exploiting the structure of KGs. Neural models (Chen et al., 2017, 2018; Wang et al., 2018; Cao et al., 2019) replaced conventional approaches (Jiménez-Ruiz and Grau, 2011; Suchanek et al., 2011) as the core methods used in recent years. Typically they rely on seed alignment as training data – this is expensive to annotate. Iterative training (i.e., self-training) has been applied to improve EA models by generating more training data automatically (Sun et al., 2018; Mao et al., 2020). These works concern better training methods with given annotated data. However, the problem of reducing the cost of annotation has been neglected. Berrendorf et al. (2021) have been the first to explore AL strategies for EA task. They compared several types of AL heuristics including node centrality, uncertainty, graph coverage, unmatchable entities, etc. and they empirically showed the impact of sampling strategies on the creation of seed alignment. In our work, we highlight the limitations of single heuristics and propose an AL framework that can consider information structure, uncertainty sampling and unmatchable entities at the same time. In addition, existing neural models assume all KGs entities have counterparts: this is a very strong assumption in reality (Zhao et al., 2020). We provide a solution to recognizing the bachelor entities, which is complementary to the existing models.

Active Learning. Active Learning is a general framework for selecting the most informative data to annotate when training Machine Learning models (Aggarwal et al., 2014). The pool-based sampling scenario is a popular AL setting where a base pool of unlabelled instances is available to query from (Settles, 2012; Aggarwal et al., 2014). Our proposed AL framework follows this scenario. Numerous AL strategies have been proposed in the general domain (Aggarwal et al., 2014). *Uncertainty sampling* is the most widely used because of its ease to implement and its robust effectiveness (Lewis, 1995; Cohn et al., 1996). However, there are key challenges that general AL strategies cannot solve when applying AL to EA. Most AL strategies are designed under the assumption that the data is independent and identically distributed. However, KGs entities in the AL task are correlated, as in other graph-based tasks, e.g., node classification (Bilgic et al., 2010) and link prediction (Ostapuk et al., 2019). In addition, bachelor entities cause a very special issue in EA. They may have low informativeness but high uncertainty. We design an AL strategy to solve these special challenges. Few existing works (Qian et al., 2017; Malmi et al., 2017) have applied AL to conventional EA but do not consider neural EA models, which have now become of widespread use. Only Berrendorf et al. (2021) empirically explored general AL strategies for neural EA but did not solve the aforementioned challenges.

7 Conclusion

Entity Alignment is an essential step for KG fusion. Current mainstream methods for EA are neural models, which rely on seed alignment. The cost of labelling seed alignment is often high, but how to reduce this cost has been neglected. In this work, we proposed an Active Learning framework (named ActiveEA), aiming to produce the best EA model with the least annotation cost. Specifically, we attempted to solve two key challenges affecting EA that general AL strategies cannot deal with. Firstly, we proposed a structure-aware uncertainty sampling, which can combine uncertainty sampling with the structure information of KGs. Secondly, we designed a bachelor recognizer, which reduces annotation budget by avoiding the selection of bachelors. Specially, it can tolerate sampling biases. Extensive experimental showed ActiveEA is more effective than the considered baselines and has great generality across different datasets, EA models and bachelor percentages.

In future, we plan to explore combining active learning and self-training which we believe are complementary approaches. Self-training can generate extra training data automatically but suffers from incorrectly labelled data. This can be addressed by amending incorrectly labelled data using AL strategies.

Acknowledgements

This research is supported by the Shenyang Science and Technology Plan Fund (No. 20-201-4-10), the Member Program of Neusoft Research of Intelligent Healthcare Technology, Co. Ltd.(No. NRMP001901)). Dr Wen Hua is the recipient of an Australian Research Council DECRA Research Fellowship (DE210100160). Dr Guido Zuccon is the recipient of an Australian Research Council DECRA Research Fellowship (DE180101579).

References

Charu C. Aggarwal, Xiangnan Kong, Quanquan Gu, Jiawei Han, and Philip S. Yu. 2014. Active learning: A survey. In Charu C. Aggarwal, editor, *Data Classification: Algorithms and Applications*, pages 571–606. CRC Press.

Max Berrendorf, Evgeniy Faerman, and Volker Tresp. 2021. Active learning for entity alignment. In *Advances in Information Retrieval - 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28 - April 1, 2021, Proceedings, Part I*, volume 12656 of *Lecture Notes in Computer Science*, pages 48–62. Springer.

Mustafa Bilgic, Lilyana Mihalkova, and Lise Getoor. 2010. Active learning for networked data. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, pages 79–86. Omnipress.

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2787–2795.

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Comput. Networks*, 30(1-7):107–117.

Yixin Cao, Zhiyuan Liu, Chengjiang Li, Zhiyuan Liu, Juanzi Li, and Tat-Seng Chua. 2019. Multi-channel graph neural network for entity alignment. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1452–1461. Association for Computational Linguistics.

Muhao Chen, Yingtao Tian, Kai-Wei Chang, Steven Skiena, and Carlo Zaniolo. 2018. Co-training embeddings of knowledge graphs and entity descriptions for cross-lingual entity alignment. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 3998–4004. ijcai.org.

Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. 2017. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 1511–1517. ijcai.org.

David A. Cohn, Zoubin Ghahramani, and Michael I. Jordan. 1996. Active learning with statistical models. *J. Artif. Intell. Res.*, 4:129–145.

Massimo Franceschet. 2011. Pagerank: standing on the shoulders of giants. *Commun. ACM*, 54(6):92–101.

Linton C Freeman. 1977. A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41.

Paul A Gagniuc. 2017. *Markov chains: from theory to implementation and experimentation*. John Wiley & Sons.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. 2017. Deep bayesian active learning with image data. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1183–1192. PMLR.

William L. Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 1024–1034.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. 2020. A survey on knowledge graphs: Representation, acquisition and applications. *CoRR*, abs/2002.00388.

Ernesto Jiménez-Ruiz and Bernardo Cuenca Grau. 2011. Logmap: Logic-based and scalable ontology matching. In *The Semantic Web - ISWC 2011 - 10th International Semantic Web Conference, Bonn, Germany, October 23-27, 2011, Proceedings, Part I*, volume 7031 of *Lecture Notes in Computer Science*, pages 273–288. Springer.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. 2015. Dbpedia - A large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195.

David D. Lewis. 1995. A sequential algorithm for training text classifiers: Corrigendum and additional data. *SIGIR Forum*, 29(2):13–19.

Zhiyuan Liu, Yixin Cao, Liangming Pan, Juanzi Li, and Tat-Seng Chua. 2020. Exploring and evaluating attributes, values, and structures for entity alignment. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6355–6364. Association for Computational Linguistics.

Eric Malmi, Aristides Gionis, and Evimaria Terzi. 2017. Active network alignment: A matching-based approach. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, November 06 - 10, 2017*, pages 1687–1696. ACM.

Xin Mao, Wenting Wang, Huimin Xu, Man Lan, and Yuanbin Wu. 2020. MRAEA: an efficient and robust entity alignment approach for cross-lingual knowledge graph. In *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020*, pages 420–428. ACM.

Sidharth Mudgal, Han Li, Theodoros Rekatsinas, An-Hai Doan, Youngchoon Park, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, and Vijay Raghavendra. 2018. Deep learning for entity matching: A design space exploration. In *Proceedings of the 2018 International Conference on Management of Data, SIGMOD Conference 2018, Houston, TX, USA, June 10-15, 2018*, pages 19–34. ACM.

Natalia Ostapuk, Jie Yang, and Philippe Cudré-Mauroux. 2019. Activelink: Deep active learning for link prediction in knowledge graphs. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, pages 1398–1408. ACM.

Kun Qian, Lucian Popa, and Prithviraj Sen. 2017. Active learning for large-scale entity resolution. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, November 06 - 10, 2017*, pages 1379–1388. ACM.

Thomas Rebele, Fabian M. Suchanek, Johannes Hoffart, Joanna Biega, Erdal Kuzey, and Gerhard Weikum. 2016. YAGO: A multilingual knowledge base from wikipedia, wordnet, and geonames. In *The Semantic Web - ISWC 2016 - 15th International Semantic Web Conference, Kobe, Japan, October 17-21, 2016, Proceedings, Part II*, volume 9982 of *Lecture Notes in Computer Science*, pages 177–185.

Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Xiaojiang Chen, and Xin Wang. 2020. A survey of deep active learning. *CoRR*, abs/2009.00236.

Burr Settles. 2012. *Active Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers.

Fabian M. Suchanek, Serge Abiteboul, and Pierre Senellart. 2011. PARIS: probabilistic alignment of relations, instances, and schema. *Proc. VLDB Endow.*, 5(3):157–168.

Zequn Sun, Wei Hu, Qingheng Zhang, and Yuzhong Qu. 2018. Bootstrapping entity alignment with knowledge graph embedding. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 4396–4402. ijcai.org.

Zequn Sun, Chengming Wang, Wei Hu, Muhao Chen, Jian Dai, Wei Zhang, and Yuzhong Qu. 2020a. Knowledge graph alignment network with gated multi-hop neighborhood aggregation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 222–229. AAAI Press.

Zequn Sun, Qingheng Zhang, Wei Hu, Chengming Wang, Muhao Chen, Farahnaz Akrami, and Chengkai Li. 2020b. A benchmarking study of embedding-based entity alignment for knowledge graphs. *Proc. VLDB Endow.*, 13(11):2326–2340.

Denny Vrandecic and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

Zhichun Wang, Qingsong Lv, Xiaohan Lan, and Yu Zhang. 2018. Cross-lingual knowledge graph alignment via graph convolutional networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 349–357. Association for Computational Linguistics.

Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, Rui Yan, and Dongyan Zhao. 2019. Relation-aware entity alignment for heterogeneous knowledge graphs. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 5278–5284. ijcai.org.

Xiang Zhao, Weixin Zeng, Jiuyang Tang, Wei Wang, and Fabian Suchanek. 2020. An experimental study of state-of-the-art entity alignment approaches. *IEEE Annals of the History of Computing*, (01):1–1.