Joint Neural Entity Disambiguation with Output Space Search

Hamed Shahbazi, Xiaoli Z. Fern, Reza Ghaeini, Chao Ma,
Rasha Obeidat, Prasad Tadepalli
Oregon State University, Corvallis, OR, USA
{shahbazh, xfern, ghaeinim, machao, obeidatr, tadepall}@eecs.oregonstate.edu

Abstract

In this paper, we present a novel model for entity disambiguation that combines both local contextual information and global evidences through Limited Discrepancy Search (LDS). Given an input document, we start from a complete solution constructed by a local model and conduct a search in the space of possible corrections to improve the local solution from a global view point. Our search utilizes a heuristic function to focus more on the least confident local decisions and a pruning function to score the global solutions based on their local fitness and the global coherences among the predicted entities. Experimental results on CoNLL 2003 and TAC 2010 benchmarks verify the effectiveness of our model.

1 Introduction

The goal of entity disambiguation is to link a set of given query mentions in a document to their referent entities in a Knowledge Base (KB). As an essential and challenging task in Knowledge-Base Population (KBP) for text Analysis (Ji et al., 2014; Heng et al., 2015), entity disambiguation has attracted many research efforts from the NLP community. Recently, deep learning based approaches have demonstrated strong performances on this task (Ganea and Hofmann, 2017; Sil et al., 2018).

A main challenge for entity disambiguation is to best identify and represent the appropriate context, which can be local or global. Different methods have been proposed to capture and represent different types of contexts. Textual context has been heavily investigated for local models that score each query's candidates independently. Representations of the textual contexts range from weighted combination of the word embeddings based on attention (Ganea and Hofmann, 2017), to more fine-grained contextual representations using recurrent neural networks (Sil et al., 2018). The global context of other entities in the document has also been studied for a more global and joint prediction view on the problem. Ganea and Hofmann (2017) use a Conditional Random Field (CRF) based model to capture the interrelationship among entities in the same document, whereas Globerson et al. (2016) introduce a soft k-max attention model to weigh the importance of other entities in the document in making prediction for any given query.

Working with the recently proposed models, we observe that local models that employ an appropriate attention mechanism often have a solid linking performance. In a single document, there are often a small number of hard queries for which the local model fails to make a correct decision. We conjecture that if some of these mistakes can be corrected, a global model that enforces coherence among entities will be able to propagate these corrections to improve the overall solution quite effectively.

This inspires us to consider the Limited Discrepancy Search (LDS) framework (Doppa et al., 2014), which conducts a search over possible corrections on a complete output with the goal of improving the final output. Critically, LDS works well for cases where only a small number of local corrections are needed to reach a good global solution. This nicely matches up with our observation of the behavior of entity disambiguating models.

Proceedings of the 27th International Conference on Computational Linguistics, pages 2170–2180
Santa Fe, New Mexico, USA, August 20-26, 2018.

In this paper, we propose a LDS based global entity disambiguating system. Given a document and its query mentions, our system first applies a local disambiguation model to produce an initial solution. We then use LDS to conduct a shallow search in the space of possible corrections (with the focus on hard/least confident mentions) to find a better solution. Evaluation on CoNLL 2003 and TAC 2010 shows that our method outperforms the current state-of-the-art models. We also conduct an extensive ablation study on different variants of our model, providing insight into the strength of our method.

2 Proposed Approach

We are given a document D containing n mentions $[x_1, ...x_n]$. We assume that all mentions are linkable to a Knowledge Base (KB) by excluding the NILL mentions. We are interested in finding a joint assignment of all mentions to $\mathbf{Y} = [y_1, ...y_n]$ of referent entities to maximize the following score:

$$s(\mathbf{Y}) = \sum_{i=1}^{n} \psi(x_i, y_i) + \sum_{i=1}^{n} \sum_{j=1:j\neq i}^{n} \phi(y_i, y_j) \quad (1)$$

where function $\psi(x_i, y_i)$ gives the local compatibility score between the mention x_i and its candidate y_i and $\phi(y_i, y_j)$ indicates the amount of relatedness between the assigned candidates y_i and y_j. Optimizing this objective, however, is NP-hard. In this work, we develop a LDS-based search strategy for optimizing this objective.

2.1 Overview of the Approach

Given a document with n mentions, we initialize the search with a solution acquired based solely on the local scoring function $\psi(\cdot, \cdot)$. We then conduct a greedy beam search in the space of possible discrepancies (changes/corrections) to this initial solution while focusing on mentions with least confident local scores in the hope of finding a better solution. Fig 1 shows the overview of our search framework

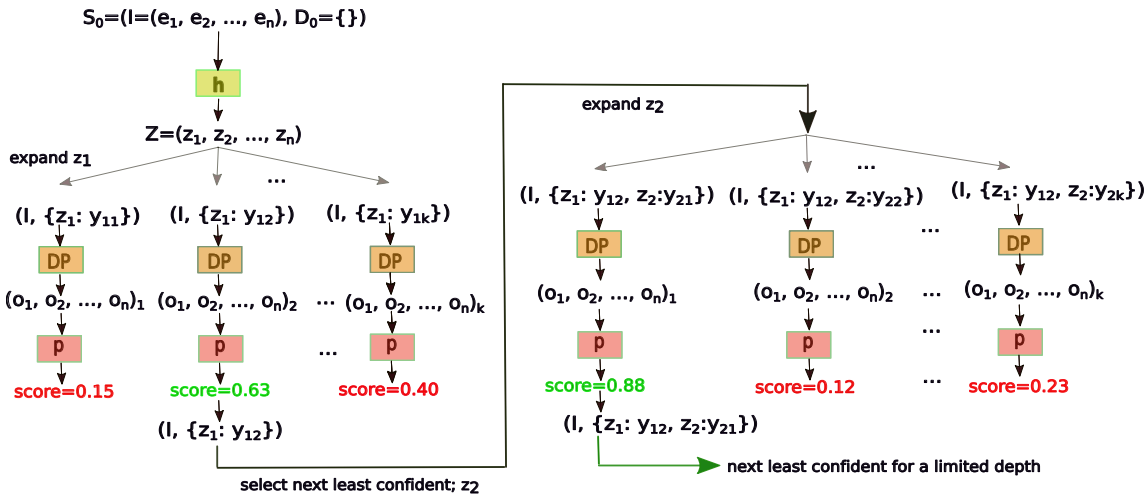

Figure 1: The general framework of our proposed model with beam size b=1.

for beam size $b = 1$. Each state S_i is a pair (I, D_i) where $I = (e_1, e_2, ..., e_n)$ is the initial solution given by the local model and D_i is the discrepancy set for state S_i. For example, a discrepancy set $\{x_{k_1} : y_{k_1}, x_{k_2} : y_{k_2}\}$ contains two discrepancies, changing the assignment for mentions x_{k_1} and x_{k_2} from e_{k_1} and e_{k_2} in I to y_{k_1} and y_{k_2} respectively. Starting with initial state (I,), we utilize a heuristic function h (Section 2.3.2) to sort the mentions in the increasing order of their local confidence. Let $(z_1, ..., z_n)$ be the ordered mentions where z_1 is the least confident mention. Each iteration of the greedy beam search explores and prunes the space of discrepancy sets as follows:

We select the least confident mention z_1 and expand state $(I, \{\})$ to k new states $(I, \{z_1 : y_{1j}\})$ for $j = 1 \cdots k$ where y_{1j} is the j^{th} candidate for mention z_1 (we consider top k most probable candidates). Each

expanded state $(I, \{z_1 : y_{1j}\}), j = 1, \ldots, k$ is given to a Discrepancy Propagator (DP) (Section 2.3.1) to get its discrepancy set propagated throughout the document and produce an updated complete solution $(o_1, ...o_n)_j$, for $j = 1, \ldots, k$. Utilizing a trained pruning function p (Section 2.3.3) we then rank the k complete solutions and prune them to top b states (b is beam size).

The search continues with the next least confident mention z_2 as shown in Fig 1. Each iteration increases the size of the discrepancy set by one. Note that a mention is not repeated in a discrepancy set. We consider two different strategies for terminating the search depending on the used heuristic function h (Section 2.3.3). Each strategy causes the search to terminate at different depth (length) of the discrepancy set. The output of the search is selected by the pruning function p from the last set of complete solutions.

In the following, we will first explain our **local model** ($\psi(\cdot, \cdot)$) for producing the initial solution. We will then introduce our LDS search framework, and its key components including the **Discrepancy Propagator** to propagate a set of discrepancies to other mentions in the document; the **Heuristic Function** to compute the local confidence of the mentions in the document and identify the least confident ones; and finally the **Pruning Function** to guide the search and select the final solution.

2.2 Local Model

Our local model utilizes contextual, lexical and prior evidences to compute the compatibility score $\psi(x_i, y_i)$ for assigning candidate y_i to mention x_i. These evidences are extracted and used as follows:

2.2.1 Contextual Evidence

Given a query mention, a key challenge for the local model is to identify a minimum but sufficient contextual evidence to disambiguate the query. We first extract all sentences in the document relevant to the query mention. This is achieved by applying CoreNLP (Manning and McClosky, 2014) to perform coreference resolution to all mentions in the document and extract all sentences containing a mention in the query's coreference chain.

The set of sentences are then concatenated to form the context for the query; $w_i = [w_{i1}, ...w_{im}]$, where $w_{ij} \in R^d$ is the embedding of the j-th word in the context. We then use an attention model introduced by (Ganea and Hofmann, 2017) to compress the context into a single embedding. Specifically, we define the contextual representation $c_i \in R^d$ for mention x_i with candidate set $\{y_{i1}, y_{i2}, ...y_{ik}\}$ over context w_i as

$$c_i = \sum_{l=1}^{m} \alpha_{il} w_{il} \tag{2}$$

The weight vector α is computed using the following attention considering all k entity candidates:

$$\alpha_i = \textbf{softmax}([\max_{j \in 1..k} y_{ij}^T A w_{i1}, ..., \max_{j \in 1..k} y_{ij}^T A w_{im}]) \tag{3}$$

where A is a learned matrix that scores the relatedness between word and entity. This attention model computes the relatedness of each word with all of the entity candidates and takes the max as the score for each word. The scores of all context words are then passed through softmax to compute their weight. Under this model, if a word is strongly related to one of the candidates, it will be given a high weight. Subsequently, using c_i we define the following contextual features for candidate y_{ij} as $f_{ij}^{(c)} = [y_{ij}^T B c_i; y_{ij}^T B; c_i]$, where B is a learned $R^d \times R^d$ matrix. Note that $f_{ij}^{(c)} \in R^{2d+1}$.

2.2.2 Lexical Evidence

The contextual features extracted above ignores any lexical/surface information between query mention and entity title, which can be useful. To this end we include some lexical features to our local model including variants of the edit distance between the surface strings of the mention and the candidate title, whether their surface strings follow an acronym pattern and etc. Detailed list can be found in the Table 1(a). The extracted features are scalar real values. We use RBF binning (Sil et al., 2018) to transform each scalar to a 10-d vector. Hence for each query x_i and candidate y_{ij} we have lexical vector feature $f_{ij}^{(l)} \in R^{10 \times |f|}$ where $|f|$ is the number of features listed in Table 1(a).

(a)

mention: m = [m_1, ... m_a], **entity title:** e = [t_1, ... t_b]
f_1: mention length = a
f_2: entity title length = b
f_3: $\sum_{i=1}^{b}$ (occurrence counts of t_i in the document)
f_4: m is acronym
f_5: e is acronym
f_6: m and e acronym patterns are exact match
f_7: m and e are non-acronyms and exact match
f_8: min-edit(m, e)
f_{10}: sum of partial min edits: $\sum_{i=1}^{a}(\min_{j=1}^{b}(\text{min-edit}(m_i, t_j)))$

(b)

local score for mention m: l = **sotmax**([$\psi(x_i, y_{i1})$, ... $\psi(x_i, y_{ik})$])
f_1: max(l)
f_2: second-max(l)
f_3: entropy(l)
f_4: m is an acronym
f_5: length(m)

Table 1: (a) list of lexical features in the local model (b) list of the features to learn heuristic function h_2

2.2.3 Prior Evidence

We consider $p(e|m)$, the prior probability that an entity e is linked to a mention string m as prior evidence. This is computed using hyper-link statistics from Wikipedia and aliases mapping from (Hoffart et al., 2011) obtained by extending the means tables of YAGO (Hoffart et al., 2013). The $p(e|m)$ is also transformed to a 10-d vector using RBF binning (Sil et al., 2018) to create prior feature $f_{ij}^{(p)} \in R^{10}$.

2.2.4 Overall local model

The contextual, lexical and prior features are concatenated and fed through a Multi Layer Perceptron (MLP) with 2 hidden layers (relu, 200-d and 50-d respectively) to produce a final local score for all candidates as follows:

$$\psi(x_i, y_{ij}) = \sigma(W_s[f_{ij}^{(c)}; f_{ij}^{(l)}; f_{ij}^{(p)}] + b_s) \tag{4}$$

Where W_s and b_s are weight and bias parameters for the MLP. The learning of the model is two tiered. We first pre-train the matrices A and B using the cross-entropy loss by using **softmax**($[y_{ij}^T B c_i]_{j \in 1..k}$) as the predicted probability for each candidate for mention x_i, Keeping A and B fixed, we then train the MLP weights with a drop-out rate of 0.7 minimizing again the cross-entropy loss. Note that we use the entity embeddings produced by (Ganea and Hofmann, 2017) and the word embeddings produced by (Mikolov et al., 2013) using skip-gram model.

2.3 Global Model via LDS

Our local model primarily focuses on the textual context of each mention in making its decisions and ignores the relationship between mentions. Our global model takes as input the local predictions for all the mentions in a single document and constructs a globally coherent final solution using Limited Discrepancy Search. Before we introduce our LDS search procedure, we will first describe the discrepancy propagator which is critical toward achieving an efficient search space.

2.3.1 Discrepancy Propagator

The purpose of the discrepancy propagator is to allow the influence of the local changes to propagate to other parts of the solution, thus reducing the necessary number of discrepancies needed. Our discrepancy propagator is essentially a new scoring function that evaluates each candidate for a given mention not only based on the local compatibility but also the global coherence among the predictions.

Given that not all mentions in a document need to be related to one another, for each mention x_i we construct an entity context E_i considering a window of 30 mentions centered at query mention x_i. Given the current entity assignment $e_1, e_2, ...e_n$ to the mentions in the document, the entity context for mention x_i will be $E_i = e_{i-15}..., e_{i-1}, e_{i+1}, ...e_{i+15}$.
For a given mention x_i and its entity context E_i, the new score of a candidate y_{ij} is defined as:

$$g(x_i, y_{ij}) = \psi(x_i, y_{ij}) + \frac{1}{|E_i|} \sum_{e \in E_i} \phi(e, y_{ij}) \tag{5}$$

In equation 5, the score of a candidate now considers both its local score ψ as well as its average coherence with the other predictions in its entity context. The coherence score ϕ between two entities e and y_{ij} is defined as:

$$\phi(e, y_{ij}) = e^T C y_{ij} + w^T r(e, y_{ij}) \tag{6}$$

where e and y_{ij} are entity embeddings and $r(e, y_{ij})$ is a vector of three pairwise features between the two entities: log transformed counts of co-ocurrences, shared in-links and shared out-links; $C \in R^{d \times d}$ and $w \in R^3$ are learned weights.

The score given by equation 5 is fed through a softmax activation to assign probabilities to each candidate of x_i. We train W and c in a mention-wise procedure using cross entropy loss. Specifically for each mention x_i we use ground truth entities for the other mentions in E_i and update weights of C and w such that the true candidate of x_i gets higher score than its false candidates.

Note that we do not intend to use this scoring function to do joint inference on all mentions. Instead, it is used with LDS to propagate the discrepancies to the output of all related mentions as follows. Given the original solution $e_1, e_2, ...e_n$ and a set of discrepancies $\{x_k : y_k\}$, we first update e_k to y_k. Then we re-evaluate equation 5 for all mentions that have y_k in their entity context to produce a complete solution based on new entity context containing y_k.

2.3.2 Heuristic Function

The heuristic function takes the initial solution and sort all the mentions based on their prediction confidence. In this work we consider two heuristic functions for computing the prediction confidence of the local model. For a given query x_i, our first heuristic function h_1 computes its confidence simply by taking the max of the local scores normalized by a softmax funtion:

$$h_1(x_i) = \textbf{max}(\textbf{softmax}([\psi(x_i, y_{i1}), ...\psi(x_i, y_{ik})])) \tag{7}$$

where $\psi(x_i, y_{ij})$ is the local score for candidate y_{ij}.

For our second heuristic function h_2, we learn a binary classifier (a two layer MLP) to produce local confidence for each mention x_i. The binary classifier utilizes the features listed in Table 1(b). Each feature value is transformed into a 10-d vector using RBF binning (Sil et al., 2018). The training samples for the binary classifier are generated from the prediction of the local model on the training set. One training example is generated for each local prediction — if the local model is correct about its decision then the label for the sample is 1. In order to balance the positive and negative training samples, we randomly sub-sample positive local predictions. For each local prediction, the learned classifier outs a probability of its being correct, which is used as the confidence score.

2.3.3 Pruning Function

At each iteration of the search, each beam is expanded to k new states. Following the expansion, the DP is applied to the $b \times k$ expanded nodes and re-evaluates equation 5 for all mentions with updated entity contexts due to the discrepancies. It hence propagates the discrepancies and produces a complete solution. Subsequently, the pruning function evaluates each of the $b \times k$ complete solutions and reduces the expansion list to the beam size b.

The pruning function is similar to equation 5 but scores all the mentions collectively. Given all the mentions in the document $x_1, ..., x_n$ and their predicted entities denoted as $o = o_1, ..., o_n$, our pruning function scores the solution o as follows:

$$s(o) = \sum_i \psi(x_i, o_i) + \sum_{(o_i \in E_j \text{ or } o_j \in E_i)} \phi_g(o_i, o_j) \tag{8}$$

Where constraint ($o_i \in E_j$ or $o_j \in E_i$) indicates that we consider relation among pair of entities (o_i, o_j) if they are in the entity context of one another. Here ϕ_g takes the same form as equation 6 but uses a different set of weights C_g and w_g.

Despite the similarity in forms, the pruning function serves a different purpose from that of equation 6 and requires different training. We train the pruning function by reducing it to a rank learning problem.

Specifically, in training, we collect all the complete solutions that are considered in each pruning step, and compute their hamming losses from the ground truth. Given a set of solutions in a pruning step, we will create ranking pairs to require the solution with the least hamming loss to score higher than all the others. Given a ranking pair, o^t and o^f, assuming o^t has lower hamming loss than o^f, we use the following ranking loss for training:

$$max\left(0, \Delta(o^t, o^f) - s(o^t) + s(o^f)\right) \tag{9}$$

where $\Delta(o^{(t)}, o^{(f)})$ is the absolute difference of the hamming-loss between $o^{(t)}$ and $o^{(f)}$. This loss function penalizes the scoring function if it fails to score o^t higher than o^f by a margin specified by $\Delta(o^{(t)}, o^{(f)})$.

Terminating the search. We consider two different strategies for terminating the search depending on the heuristic function in use. When using h_1 as the heuristic function, search terminates when we reach a depth limit τ (a maximum τ discrepancies). The strategy with h_2 uses a flexible depth. For this strategy, we terminate the search when discrepancies have been introduced for all queries that are predicted to be incorrect by h_2.

3 Experiments

3.1 Data Sets

We use two datasets CoNLL 2003 (Hoffart et al., 2011) and TAC 2010 (Ji et al., 2010) for evaluation. The CoNLL dataset is partitioned into train, test-a and test-b with 946, 216 and 231 documents respectively. Following our baselines we only use 27816 mentions with valid links to the KB. The TAC 2010 dataset is yet another popular NED dataset released by the Text Analysis Conference (TAC). The dataset contains training and test set with 1043 and 1013 documents respectively. Similar to CoNLL we only consider linkable mentions in TAC and report our performance on 1020 query mentions in the test set.
To learn and tune the parameters of the local models for CoNLL and TAC we use their own training and development splits. However, to learn and tune the parameters of the global model (the discrepancy propagator, the heuristic and pruning functions) we only use the CoNLL training and dev-sets.
The number of queries per document in the test sets of the CoNLL and TAC are approximately 20 and 1 respectively. In order to have a global setup for TAC we apply CoreNLP (Manning and McClosky, 2014) mention extractor to the test documents of TAC and perform joint disambiguation of the extracted mentions together with the query mentions, increasing the number of mentions to approximately 4 per document. We only report the performance on the query mentions with the given standard ground truth by TAC.

3.2 Hyper-parameters and Dimensions

Our model settings for the hyper-parameters and dimensionality of the embeddings and weights are as follows: We use entity/word embeddings of 300-d. We learned the embeddings using the mechanism proposed by (Ganea and Hofmann, 2017). We use 2-layers MLPs for the local model and the heuristic h_2 with hidden layer sizes 200×50 and 100×20 for each respectively. The RBF binning always transfers a scalar to 10-d. Although we analyze and compare different configurations for beam-size and depth limit in section 3.4, we use beam size 5 and flexible depth limit τ with heuristic h_2 in our reported results.

3.3 Results

To evaluate the model performance we use the standard micro-average accuracies of the top-ranked candidate entities. We use different alias mappings for TAC and CoNLL. Specifically, for TAC we only use anchor-title alias mappings constructed from hyper-links in the Wikipedia. For CoNLL, in order to follow the experimental setup reported by (Sil et al., 2018), in addition to the alias mappings of Wikipedia anchor-titles, we use the mappings produced by (Pershina et al., 2015) and (Hoffart et al., 2011).
Tables 2(a) and (b) show our performance on the CoNLL and TAC datasets for our local and global models along with other competitive systems respectively. The prior state of the art performances on

these two datasets are achieved by (Sil et al., 2018). The results show that our global model outperforms all the competitors for both CoNLL 2003 and TAC 2010. It is interesting to note that our local model is solid, but is noticeably inferior to the stat-of-the-art local model. With an even stronger local model like that of (Sil et al., 2018), one can potentially expect our LDS-based global model to push the state-of-the-art even further.

models	In-KB acc%
local	
(He et al., 2013)	85.6
(Francis-Landau et al., 2016)	85.5
(Sil and Florian, 2016)	86.2
(Nevena Lazic and Pereira, 2015)	86.4
(Yamada et al., 2016)	90.9
(Sil et al., 2018)	94.0
global	
(Hoffart et al., 2011)	82.5
(Pershina et al., 2015)	91.8
(Globerson et al., 2016)	92.7
(Yamada et al., 2016)	93.1
our model	
local	90.89
global	**94.44**

(a) CoNLL 2003

models	In-KB acc%
local	
(Sil and Florian, 2016)	78.6
(He et al., 2013)	81.0
(Sun et al., 2015)	83.9
(Yamada et al., 2016)	84.6
(Sil et al., 2018)	87.4
global	
(Yamada et al., 2016)	85.2
(Globerson et al., 2016)	87.2
our model	
local	85.73
global	**87.9**

(b) TAC-2010

Table 2: Evaluation on CoNLL 2003 Test-b and TAC-2010

3.4 Ablation Study and Performance Analysis

For ablation we analyze the impact of the features and search. We also analyze the behavior of our model based on the rarity of the entities.

3.4.1 Feature Analysis

We study the impact of features on the local and global models. For the local model, starting with only considering the contextual evidence as described in Section 2.2.1, we see that the performance steadily increases as we add the prior and lexical features. As shown in tables 3(a) and (b) the prior and lexical features have very strong impact on TAC. The binning technique that projects the prior and lexical features to 10 dimensions gives an average of 0.94% and 0.61% percentage points for TAC and CoNLL respectively.

For the global model, entity pair compatibility is computed using both entity embeddings and log transformed counts of co-occurences, shared in-links and shared out-links. Using these features leads to a gain of 1.1% and 0.43% in accuracy for CoNLL and TAC respectively compared to the global model using only the embeddings in equation 6.

models	In-KB acc%
local	
Context only	85.37
Context + Prior + Lexical	90.28
Context + Prior + Lexical + Bin	90.89
global	
Our global	94.44
- log count features	93.34
- LDS + 1-step global prop.	93.14
- LDS + conv. global prop.	93.63

(a) CoNLL 2003

models	In-KB acc%
local	
Context	70.86
Context + Prior + Lexical	84.79
Context + Prior + Lexical + Bin	85.73
global	
Our global	87.9
- log count features	87.47
- LDS + 1 step global prop.	86.21
- LDS + conv. global prop.	86.29

(b) TAC-2010

Table 3: The performance of variants of our Local/Global models on CoNLL 2003 and TAC-2010

3.4.2 Search Analysis

In the last two rows of Table 3, we list variants of our model where LDS search is removed. Specifically, given the initial local solution, our discrepancy propagator (equation 5) can be applied without any discrepancy to re-evaluate the candidates for all mentions and obtain a more globally compatible solution. Naturally, one can repeat this for multiple iterations till convergence, which gives us an alternative global model that does not rely on search. We consider this model and show its single iteration performance (-LDS + 1-step global prop.) as well as its converged performance (-LDS + conv. global prop.) in Table 3.

From the results we can see that a single iteration of the global propagation was able to significantly improve the initial local solution, but later iterations lead to very mild gain. This is because after the first propagation the convergence is almost immediate and there is a limited improvement after the first iteration. In contrast, our search based global model is able to achieve significant further performance gain 1.3% for CoNLL and 1.68% for TAC. The reason is that with local discrepancies introduced by LDS, we force the solution to escape the current local optimum and search for better ones.

Additionally, several parameters/choices in the search can potentially impact the performance: the beam size b, the depth of search, and the different heuristics for prioritizing the discrepancy locations. The following set of experiments explore these choices of parameters:

Beam size. We compare model with beam size $b = 5$ and a simple greedy model with $b = 1$. For CoNLL 2003 and TAC-2010, the model with $b = 5$ gives a gain of 0.24% and 0.13% compared to the greedy. Increasing the size of the beam further beyond 5 did not show significant gain.

Depth of search. For heuristic h_1, we use a fixed depth strategy. In particular, given a document with n mentions, we consider two different depth limits: $\tau = 25\%n$ and $\tau = 50\%n$, which lead to an average depth of 5 and 10 per document respectively. For heuristic h_2, the depth is flexible and determined by the number of mentions that are predicted to be incorrect by h_2. This strategy leads to an average depth of 4. In Tables 4 (a) and (b), we report the confusion matrices given by h_1 applied to the local prediction on test-b of CoNLL 2003 with $\tau = 25\%n$ and $\tau = 50\%n$ respectively. In these tables correct/incorrect mean whether the local prediction is correct or not (comparing to the ground truth). Therefore the first cell with value 1134 indicates that there are 1134 mentions in test-b that are correctly predicted by the local model, but deemed as among the top 25% least confident mentions (aka the hard queries) by h_1. The confusion matrix for a good heuristic will have small diagonal values and large anti-diagonal values.

In Table 4 (c), we apply heuristic h_2 to the same data with flexible depth. These results show that heuristic h_2 gives the best precision as well as recall of the real mistakes made by the local model.

$\tau = 25\%n$	r $\leq$ 25%	r > 25%
correct	1134	2937
incorrect	276	136

(a)

$\tau = 50\%n$	r $\leq$ 50%	r > 50%
correct	2071	2000
incorrect	331	81

(b)

τ=flexible	label=0	label=1
correct	805	3266
incorrect	333	79

(c)

Table 4: Confusion Matrices for (a) using heuristic h_1 with $\tau = 25\%n$, (b) heuristic h_1 with $\tau = 50\%n$ and (c) heuristic h_2 with flexible τ.

Models	Heuristic	Depth (τ)	Beam Size (b)	In-KB acc %
Global + LDS	h_1	$\tau = 25\%n$	1	93.88
Global + LDS	h_1	$\tau = 25\%n$	5	94.12
Global + LDS	h_1	$\tau = 50\%n$	5	94.23
Global + LDS	h_2	flexible	5	94.44

Table 5: CoNLL 2003 Test-b

In Table 5, we report the performance of our model on CoNLL 2003 with different choices for the heuristic, search depth, and beam size. The results show that using a beam of size 5 improves upon single greedy search, and using heuristic h_2 with flexible depth gives the best performance in terms of both prediction accuracy and efficiency (due to smaller search trees). When h_1 is used, doubling the depth of the search tree brings about only a marginal improvement in accuracy at the cost of doubling the search depth and thus the prediction time.

3.4.3 Performance Analysis Based on Entity Rarity

We also analyze the behavior of the coherent global models based on the rarity of the entities for the given mention. Specifically, we measure the rarity of e for given query mention m by $p(e|m)$, as defined in 2.2.3, scaled to (0, 100). To this end we quantize the value of rarity measure into different bins. For each bin we compute the difference of accuracy between a coherent global model and the local model. Two coherent models are considered here; a global model without LDS (-LDS + conv.global prop in Table 3) and global model with LDS using h_2. Figure 2 shows the difference of accuracy between the global and local models per bin. As shown in this figure both global models achieve gains mostly on bins with small $p(e|m)$, which are related to the mentions with rare true entity. Additionally, the impact of

the global model when LDS is used is more significant, especially for the mentions whose true entities are most rare according to $p(e|m)$.

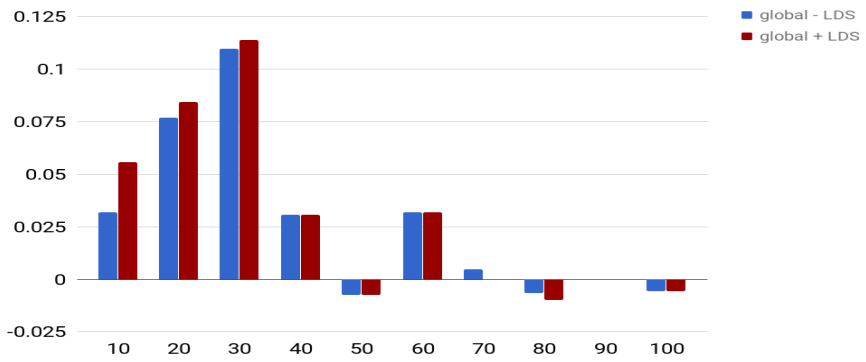

Figure 2: Global accuracy minus local accuracy per bin of rarity measure $p(e|m)$ which is scaled to (0, 100) for two coherent global models; model without LDS and model with LDS.

4 Related Work

Deep learning has been leveraged in recent local and global models. In local models Sun et al. (2015) use neural tensor networks to model mention, context and entity embeddings. Ganea and Hofmann (2017) develop an attention based model to weigh the importance of the words in the query document. The model by Sil et al. (2018) utilizes neural tensor network, multi-perspective cosine similarity and lexical composition and decomposition. In global models, the early models (Milne and Witten, 2008; Ferragina and Scaiella, 2010) decomposes the problem over mentions. Hoffart et al. (2011) use an iterative heuristic to prune edges among mention and entity. Cheng and Roth (2013) use an integer linear program solver and Lev-Arie Ratinov and Anderson (2011) apply SVM to use relation scores as ranking features.

In recent global models, Personalized PageRank (PPR) (Jeh and Widom, 2003) is adopted by several studies (Han and Sun, 2011; He et al., 2013; Alhelbawy and Gaizauskas, 2014; Pershina et al., 2015). Yamada et al. (2016) extend the skip-gram model (Mikolov et al., 2013) to learn the relatedness of entities using the linking structure of the KB. Ganea and Hofmann (2017) use a Conditional Random Field (CRF) based model to capture the interrelationship among entities in the same document whereas Globerson et al. (2016) introduce a soft k-max attention model to weight the importance of other entities in the document in making prediction for any given query.

In our proposed global model we address the intractable global optimization in a search framework. Specifically we use Limited Discrepancy Search (Doppa et al., 2014). Initialized with a local solution, LDS only explores a small number corrections to the hard queries. The propagation of the corrections enables the correction of other mistakes (even those with high local confidence) and allows us to reach high quality solutions with shallow searches. Moreover, the heuristic to prioritize the discrepancies can noticeably reduce the time without hurting the performance.

5 Conclusions

In this paper we study the problem of entity disambiguation. We are inspired by the observation that local models in this task tend to produce reasonable solutions, such that with only a small number of corrections and a proper propagation of these corrections throughout the document, one can quickly find superior solutions that are globally more coherent.

Based on this observation, we propose a search based approach that starts from an initial solution from a local model, and uses Limited Discrepancy Search (LDS) to search through the space of possible corrections with the goal of improving the linking performance. The experimental results show that our global model improves the state of the art on both the CoNLL 2003 and TAC 2010 benchmarks. For future research, we are interested in further understanding of the strengths and weaknesses of the LDS-based approach for different types of queries and entities. We are also interested in applying different local models to initialize the LDS search.

References

Ayman Alhelbawy and Robert Gaizauskas. 2014. Graph ranking for collective named entity disambiguation. *In Proc. 52nd Annual Meeting of the Association for Computational Linguistics, ACL.*

Xiao Cheng and Dan Roth. 2013. Relational inference for wikification. *In Proc. of EMNLP*.

Janardhan Rao Doppa, Alan Fern, and Prasad Tadepalli. 2014. Structured prediction via output space search. *Journal of Machine Learning Research*, 15:1317–1350.

Paolo Ferragina and Ugo Scaiella. 2010. Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). *In Proc. of the 19th ACM International Conference on Information Knowledge and Management, CIKM*.

Matthew Francis-Landau, Greg Durrett, and Dan Klein. 2016. semantic similarity for entity linking with convolutional neural networks. *Annual Conference of the North American Chapter of the Association for Computational Linguistics, NAACL.*

Octavian-Eugen Ganea and Thomas Hofmann. 2017. Deep joint entity disambiguation with local neural attention. *In Proc. of Empirical Methods in Natural Language Processing*.

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

Xianpei Han and Le Sun. 2011. A generative entity-mention model for linking entities with knowledge base. *In Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, ACLHLT.*

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

Ji Heng, Nothman Joel, and Hachey Ben. 2015. Overview of tac-kbp2015 tri-lingual entity discovery and linking. *In Proc. Text Analysis Conference, TAC*.

Johannes Hoffart, Klaus Berberich, and Gerhard Weikum. 2013. A spatially and temporally enhanced knowledge base from wikipedia; yago2. *Artificial Intelligence*.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, and etc. 2011. Robust disambiguation of named entities in text. *In Proc. of Empirical Methods in Natural Language Processing, EMNLP*.

Glen Jeh and Jennifer Widom. 2003. Scaling personalized web search. *In Proceedings of the 12th international conference on World Wide Web, pages 271279. ACM*.

Heng Ji, Ralph Grishman, Hoa Trang Dang, Kira Griffitt, and Joe Ellis. 2010. Overview of the tac 2010 knowledge base population track. *In Proc. of the 3rd Text Analysis Conference, TAC*.

Heng Ji, Nothman Joel, and Hachey Ben. 2014. Overview of tac-kbp2014 entity discovery and linking tasks. *In Proc. Text Analysis Conference, TAC*.

Doug Downey Lev-Arie Ratinov, Dan Roth and Mike Anderson. 2011. Local and global algorithms for disambiguation to wikipedia. *In Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, ACL HLT*.

Surdeanu Mihai Bauer John Finkel Jenny Bethard Steven J. Manning, Christopher D. and David McClosky. 2014. The stanford corenlp natural language processing toolkit. *Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ACL*, pages 55–60.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *In Advances in Neural Information Processing Systems, NIPS*.

David N. Milne and Ian H. Witten. 2008. Learning to link with wikipedia. *In Proc. of the 17th ACM Conference on Information and Knowledge Management, CIKM*.

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

Maria Pershina, Yifan He, and Ralph Grishman. 2015. Personalized page rank for named entity disambiguation. *Annual Conference of the North American Chapter of the Association for Computational Linguistics, NAACL*.

Avirup Sil and Radu Florian. 2016. Towards language independent named entity linking. *Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ACL*.

Avirup Sil, Gourab Kundu, Radu Florian, and Wael Hamza. 2018. Neural cross-lingual entity linking. *Thirty-Second AAAI Conference on Artificial Intelligence, AAAI*.

Yaming Sun, Lei Lin, Duyu Tang, Nan Yang, Zhenzhou Ji, and Xiaolong Wang. 2015. Modeling mention, context and entity with neural networks for entity disambiguation. *International Joint Conference on Artificial Intelligence, IJCAI*.

Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2016. Joint learning of the embedding of words and entities for named entity disambiguation. *International Conference on Computational Linguistics, COLING*.