

Entity Synonym Discovery via Multipiece Bilateral Context Matching

Chenwei Zhang^{1*}, Yaliang Li², Nan Du³, Wei Fan³ and Philip S. Yu⁴

¹Amazon, Seattle, WA 98109 USA

²Alibaba Group, Bellevue, WA 98004 USA

³Tencent Medical AI Lab, Palo Alto, CA 94306 USA

⁴University of Illinois at Chicago, Chicago, IL 60607 USA

cwzhang@amazon.com, yaliang.li@alibaba-inc.com, {ndu,davidwfan}@tencent.com, psyu@uic.edu

Abstract

Being able to automatically discover synonymous entities in an open-world setting benefits various tasks such as entity disambiguation or knowledge graph canonicalization. Existing works either only utilize entity features, or rely on structured annotations from a single piece of context where the entity is mentioned. To leverage diverse contexts where entities are mentioned, in this paper, we generalize the distributional hypothesis to a multi-context setting and propose a synonym discovery framework that detects entity synonyms from free-text corpora with considerations on effectiveness and robustness. As one of the key components in synonym discovery, we introduce a neural network model SYNONYMNET to determine whether or not two given entities are synonym with each other. Instead of using entities features, SYNONYMNET makes use of multiple pieces of contexts in which the entity is mentioned, and compares the context-level similarity via a bilateral matching schema. Experimental results demonstrate that the proposed model is able to detect synonym sets that are not observed during training on both generic and domain-specific datasets: Wiki+Freebase, PubMed+UMLS, and MedBook+MKG, with up to 4.16% improvement in terms of Area Under the Curve and 3.19% in terms of Mean Average Precision compared to the best baseline method. Code and data are available¹.

1 Introduction

Discovering synonymous entities from a massive corpus is an indispensable task in automated knowledge discovery. For each entity, its synonyms refer to the entities that can be used interchangeably under certain contexts. For example, `clogged nose` and `nasal congestion` are synonyms relative to the context in which they are mentioned. Given two entities, the synonym discovery task determines how likely these two entities are synonym with each other. The

main goal of synonym discovery is to learn a metric that distinguishes synonym entities from non-synonym ones.

The synonym discovery task is challenging to deal with for the following reasons. First of all, entities are expressed with variations. For example, `U.S.A/United States of America/United States/U.S.` refer to the same idea but are expressed quite differently. Recent works on synonym discovery focus on learning the similarity from entities and their character-level features [Neculoiu *et al.*, 2016; Mueller and Thyagarajan, 2016]. These methods work well for synonyms that share a lot of character-level features like `airplane/aeroplane` or an entity and its abbreviation like `Acquired Immune Deficiency Syndrome/AIDS`. However, a large number of synonym entities in the real world do not share a lot of character-level features, such as `JD/law degree`, or `clogged nose` expressed on social media vs. `nasal congestion` mentioned in medical books. With only character-level features being used, these models hardly obtain the ability to discriminate entities that share similar semantics but are not alike verbatim. Secondly, the nature of synonym discovery tasks in real-world scenarios makes it common yet more difficult under an open-world setting: new entities and synonyms emerge and need to be discovered from the text corpora.

Context information helps indicate entity synonymy. The distributional semantics theory [Harris, 1954; Firth, 1957] hypothesizes that the meaning of an entity can be reflected by its neighboring words in the text. Current works achieve decent performance on entity similarity learning, but still suffer from the following issues: 1) **Semantic Structure**. Context, as a snippet of natural language sentence, is semantically structured. Some existing models encode the semantic structures in contexts implicitly during the entity representation learning process [Mikolov *et al.*, 2013; Pennington *et al.*, 2014; Peters *et al.*, 2018]. The entity representations embody meaningful semantics: entities with similar contexts are likely to live in proximity in the embedding space. Some other works explicitly incorporate structured annotations to model contexts. Dependency parsing tree [Qu *et al.*, 2017], user click information [Wei *et al.*, 2009], or signed heterogeneous graphs [Ren and Cheng, 2015] are introduced as the structured information to help discover synonyms. However, structured annotations are time-consuming to obtain and may not even exist in an open-world setting. 2) **Di-**

*Work done while at the University of Illinois at Chicago

¹<https://github.com/czhang99/SynonymNet>

verse Contexts. A single entity can be mentioned in different contexts, let alone the case for multiple synonymous entities. Previous works on context-based synonym discovery either focus on entity information only [Neculoiu *et al.*, 2016; Mueller and Thyagarajan, 2016], or a single piece of context for each entity [Liao *et al.*, 2017; Qu *et al.*, 2017] for context matching. Notably, in specific domains such as medical, individuals (patients/doctors) may provide different context information when mentioning the same entity. Thus, using a single piece of context may suffer from noises. Incorporating multiple pieces of contexts explicitly for entity matching has the potential to improve both accuracy and robustness, which is less studied in existing works. Moreover, it is not practical to assume that multiple pieces of contexts are equally informative to represent the meaning of an entity: a context may contribute differently when being matched to different entities. Thus it is imperative to focus on multiple pieces of contexts with a dynamic matching schema.

In light of these challenges, we propose a framework to discover synonym entities from a massive corpus without additional structured annotations. A neural network model SYNONYMNET is proposed to detect entity synonyms based on two given entities via a bilateral matching among multiple pieces of contexts in which each entity appears. A leaky unit is designed to explicitly alleviate the noises from uninformative context during the matching process. We generate synonym entities that are completely unseen during training in the experiments.

The contribution of this work is summarized as follows:

- We propose SYNONYMNET, a context-aware bilateral matching model to detect entity synonyms. SYNONYMNET generalizes the distributional hypothesis to multiple pieces of contexts.
- We introduce a synonym discovery framework that adopts SYNONYMNET to obtain synonym entities from a free-text corpus without additional annotation.
- Experiments are conducted with an open-world setting on generic and domain-specific datasets in English and Chinese, which demonstrate the effectiveness of the proposed model for synonym discovery.

2 SYNONYMNET

We introduce SYNONYMNET, our proposed model that detects whether or not two entities are synonyms to each other based on a bilateral matching between multiple pieces of contexts in which entities appear. Figure 1 gives an overview of the proposed model.

2.1 Context Retriever

For each entity e , the context retriever randomly fetches P pieces of contexts from the corpus D in which the entity appears. We denote the retrieved contexts for e as a set $C = \{c_1, \dots, c_P\}$, where P is the number of context pieces. Each piece of context $c_p \in C$ contains a sequence of words $c_p = (w_p^{(1)}, \dots, w_p^{(T)})$, where T is the length of the context, which varies from one instance to another. $w_p^{(t)}$ is the t -th word in the p -th context retrieved for an entity e .

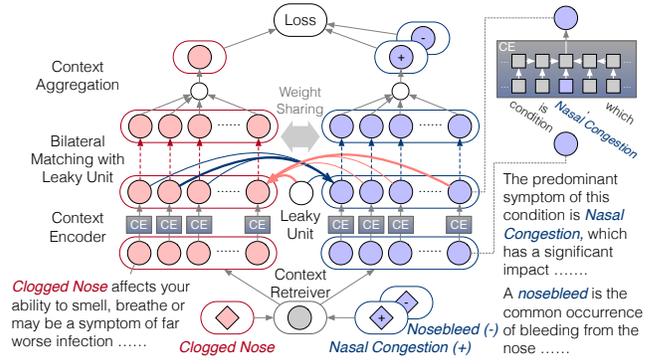


Figure 1: Overview of the proposed model SYNONYMNET. The diamonds are entities. Each circle is associated with a piece of context in which an entity appears. SYNONYMNET learns to minimize the loss calculated using multiple pieces of contexts via bilateral matching with leaky units.

2.2 Context Encoder

For the p -th context c_p , an encoder tries to learn a continuous vector that represents the context. For example, a recurrent neural network (RNN) such as a bidirectional LSTM (Bi-LSTM) [Hochreiter and Schmidhuber, 1997] can be applied to sequentially encode the context into hidden states:

$$\begin{aligned} \vec{\mathbf{h}}_p^{(t)} &= \text{LSTM}_{fw}(\mathbf{w}_p^{(t)}, \vec{\mathbf{h}}_p^{(t-1)}), \\ \overleftarrow{\mathbf{h}}_p^{(t)} &= \text{LSTM}_{bw}(\mathbf{w}_p^{(t)}, \overleftarrow{\mathbf{h}}_p^{(t+1)}), \end{aligned} \quad (1)$$

where $\mathbf{w}_p^{(t)}$ is the word embedding vector used for the word $w_p^{(t)}$. We introduce a simple encoder architecture that models contexts for synonym discovery, which learns to encode the local information around the entity from the raw context without utilizing additional structured annotations. It focuses on both forward and backward directions. However, the encoding process for each direction ceases immediately after it goes

beyond the entity word in the context: $\mathbf{h}_p = [\mathbf{h}_p^{(t_e)}, \overleftarrow{\mathbf{h}}_p^{(t_e)}]$, where t_e is the index of the entity word e in the context and $\mathbf{h}_p \in \mathbb{R}^{1 \times d_{CE}}$. By doing this, the context encoder summarizes the context while explicitly considers the entity's location in the context. Note that more advanced and sophisticated encoding methods can be used, such as EIMo, BERT, or XLNet. The encoder itself is not the main focus of this work.

2.3 Bilateral Matching with Leaky Unit

Considering the base case, where we want to identify whether or not two entities, say e and k , are synonyms with each other, we propose to find the consensus information from multiple pieces of contexts via a bilateral matching schema. Recall that for entity e , P pieces of contexts $H = \{\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_P\}$ are randomly fetched and encoded. And for entity k , we denote Q pieces of contexts being fetched and encoded as $G = \{\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_Q\}$. Instead of focusing on a single piece of context to determine entity synonymity, we adopt a bilateral matching between multiple pieces of encoded contexts for both accuracy and robustness.

$H \rightarrow G$ matching phrase: For each \mathbf{h}_p in H and \mathbf{g}_q in G , the matching score $m_{p \rightarrow q}$ is calculated as:

$$m_{p \rightarrow q} = \frac{\exp(\mathbf{h}_p \mathbf{W}_{\text{BM}} \mathbf{g}_q^T)}{\sum_{p' \in P} \exp(\mathbf{h}_{p'} \mathbf{W}_{\text{BM}} \mathbf{g}_q^T)}, \quad (2)$$

where $\mathbf{W}_{\text{BM}} \in \mathbb{R}^{d_{CE} \times d_{CE}}$ is a bi-linear weight matrix.

Similarly, the $H \leftarrow G$ matching phrase considers how much each context $\mathbf{g}_q \in G$ could be useful to $\mathbf{h}_p \in H$:

$$m_{p \leftarrow q} = \frac{\exp(\mathbf{g}_q \mathbf{W}_{\text{BM}} \mathbf{h}_p^T)}{\sum_{q' \in Q} \exp(\mathbf{g}_{q'} \mathbf{W}_{\text{BM}} \mathbf{h}_p^T)}. \quad (3)$$

Note that $P \times Q$ matching needs to be conducted in total for each entity pair. We write the equations for each $\mathbf{h}_p \in H$ and $\mathbf{g}_q \in G$ for clarity. Regarding the implementation, the bilateral matching can be easily written and effectively computed in a matrix form, where a matrix multiplication is used $\mathbf{H} \mathbf{W}_{\text{BM}} \mathbf{G}^T \in \mathbb{R}^{P \times Q}$ where $\mathbf{H} \in \mathbb{R}^{P \times d_{CE}}$ and $\mathbf{G} \in \mathbb{R}^{Q \times d_{CE}}$. The matching score matrix \mathbf{M} can be obtained by taking softmax on the $\mathbf{H} \mathbf{W}_{\text{BM}} \mathbf{G}^T$ matrix over certain axis (over 0-axis for $\mathbf{M}_{p \rightarrow q}$, 1-axis for $\mathbf{M}_{p \leftarrow q}$).

Not all contexts are informative during the matching for two given entities. Some contexts may contain intricate contextual information even if they mention the entity explicitly. In this work, we introduce a leaky unit during the bilateral matching, so that uninformative contexts can be routed via the leaky unit rather than forced to be matched with any informative contexts. The leaky unit is a dummy vector $\mathbf{l} \in \mathbb{R}^{1 \times d_{CE}}$, where its representation can be either trained with the whole model for each task/dataset, or kept as a fixed zero vector. We adopt the later design for simplicity. If we use the $H \rightarrow G$ matching phrase as an example, the matching score from the leaky unit \mathbf{l} to the q -th encoded context in \mathbf{g}_q is:

$$m_{\mathbf{l} \rightarrow q} = \frac{\exp(\mathbf{l} \mathbf{W}_{\text{BM}} \mathbf{g}_q^T)}{\exp(\mathbf{l} \mathbf{W}_{\text{BM}} \mathbf{g}_q^T) + \sum_{p' \in P} \exp(\mathbf{h}_{p'} \mathbf{W}_{\text{BM}} \mathbf{g}_q^T)}. \quad (4)$$

If there is any uninformative context in H , say the \tilde{p} -th encoded context, $\mathbf{h}_{\tilde{p}}$ will contribute less when matched with \mathbf{g}_q due to the leaky effect: when $\mathbf{h}_{\tilde{p}}$ is less informative than the leaky unit \mathbf{l} . Thus, the matching score between $\mathbf{h}_{\tilde{p}}$ and \mathbf{g}_q is now calculated as follows:

$$m_{\tilde{p} \rightarrow q} = \frac{\exp(\mathbf{h}_{\tilde{p}} \mathbf{W}_{\text{BM}} \mathbf{g}_q^T)}{\exp(\mathbf{l} \mathbf{W}_{\text{BM}} \mathbf{g}_q^T) + \sum_{p' \in P} \exp(\mathbf{h}_{p'} \mathbf{W}_{\text{BM}} \mathbf{g}_q^T)}. \quad (5)$$

2.4 Context Aggregation

The informativeness of a context for an entity should not be a fixed value: it heavily depends on the other entity and the other entity's contexts that we are comparing with. The bilateral matching scores indicate the matching among multiple pieces of encoded contexts for two entities. For each piece of encoded context, say \mathbf{g}_q for the entity k , we use the highest matched score with its counterpart as the relative informativeness score of \mathbf{g}_q to k , denote as $a_q = \max(m_{p \rightarrow q} | p \in P)$. Here the intuition is that the informativeness of a piece of

context for one entity is characterized by how much it can be matched with the most similar context for the other entity. We further aggregate multiple pieces of encoded contexts for each entity to a global context based on the relative informativeness scores:

$$\begin{aligned} \text{for entity } e: \quad \bar{\mathbf{h}} &= \sum_{p \in P} a_p \mathbf{h}_p, \\ \text{for entity } k: \quad \bar{\mathbf{g}} &= \sum_{q \in Q} a_q \mathbf{g}_q. \end{aligned} \quad (6)$$

Note that due to the leaky effect, less informative contexts are not forced to be heavily involved during the aggregation: the leaky unit may be more competitive than contexts that are less informative, thus having a larger matching score. However, as the leaky unit and its matching score are not used for aggregation—scores on informative contexts become more salient during context aggregation.

2.5 Training Objectives

We introduce two architectures for training the SYNONYMNET: a siamese architecture and a triplet architecture. **Siamese Architecture** The Siamese architecture takes two entities e and k , along with their contexts H and G as the input. The following loss function L_{Siamese} is used in training for the Siamese architecture [Neculoiu *et al.*, 2016]:

$$L_{\text{Siamese}} = yL_+(e, k) + (1 - y)L_-(e, k), \quad (7)$$

where it contains losses for two cases: $L_+(e, k)$ when e and k are synonyms to each other ($y = 1$), and $L_-(e, k)$ when e and k are not ($y = 0$):

$$\begin{aligned} L_+(e, k) &= \frac{1}{4}(1 - s(\bar{\mathbf{h}}, \bar{\mathbf{g}}))^2, \\ L_-(e, k) &= \max(s(\bar{\mathbf{h}}, \bar{\mathbf{g}}) - m, 0)^2, \end{aligned} \quad (8)$$

where $s(\cdot)$ is a similarity function, e.g. cosine similarity, and m is the margin value. $L_+(e, k)$ decreases monotonically as the similarity score becomes higher within the range of $[-1, 1]$. $L_+(e, k) = 0$ when $s(\bar{\mathbf{h}}, \bar{\mathbf{g}}) = 1$. For $L_-(e, k)$, it remains zero when $s(\bar{\mathbf{h}}, \bar{\mathbf{g}})$ is smaller than a margin m . Otherwise $L_-(e, k)$ increases as $s(\bar{\mathbf{h}}, \bar{\mathbf{g}})$ becomes larger.

Triplet Architecture The Siamese loss makes the model assign rational pairs with absolute high scores and irrational ones with low scores, while the rationality of entity synonymy could be dynamic based on entities and contexts. The triplet architecture learns a metric such that the global context $\bar{\mathbf{h}}$ of an entity e is relatively closer to a global context $\bar{\mathbf{g}}_+$ of its synonym entity, say k_+ , than it is to the global context $\bar{\mathbf{g}}_-$ of a negative example $\bar{\mathbf{g}}_-$ by some margin m . The following loss function L_{Triplet} is used in training for the Triplet architecture:

$$L_{\text{Triplet}} = \max(s(\bar{\mathbf{h}}, \bar{\mathbf{g}}_-) - s(\bar{\mathbf{h}}, \bar{\mathbf{g}}_+) + m, 0). \quad (9)$$

2.6 Inference

The objective of the inference phase is to discover synonym entities for a given query entity from the corpus effectively. We utilize context-aware word representations to obtain candidate entities that narrow down the search space. The SYNONYMNET verifies entity synonymy by assigning a synonym score for two entities based on multiple pieces of contexts. The overall framework is described in Figure 2.

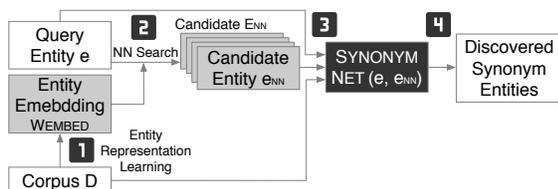


Figure 2: Synonym discovery during the inference phase with SYNONYMNET.

When given a query entity e , it is tedious and very ineffective to verify its synonymity with all the other possible entities. In the first step, we train entity representation unsupervisedly from the massive corpus D using methods such as skipgram [Mikolov *et al.*, 2013] or GloVe [Pennington *et al.*, 2014]. An embedding matrix can be learned $\mathbf{W}_{\text{EMBED}} \in \mathbb{R}^{v \times d_{\text{EMBED}}}$, where v is the number of unique tokens in D . Although these unsupervised methods utilize the context information to learn semantically meaningful representations for entities, they are not tailored for the entity synonym discovery task. For example, `nba championship`, `chicago black hawks` and `american league championship series` have similar representations because they tend to share some similar neighboring words. But they are not synonyms with each other. However, they do serve as an effective way to obtain candidates because they tend to give entities with similar neighboring context words similar representations. In the second step, we construct a candidate entity list E_{NN} by finding nearest neighbors of a query entity e in the entity embedding space of $\mathbb{R}^{d_{\text{EMBED}}}$. Ranking entities by their embedding proximities with the query entity significantly narrows down the search space for synonym discovery. In the third step, for each candidate entity $e_{NN} \in E_{NN}$ and the query entity e , we randomly fetch multiple pieces of contexts in which entities are mentioned, and feed them into the proposed SYNONYMNET model. In the last step, SYNONYMNET calculates a score $s(e, e_{NN})$ based on the bilateral matching with leaky units over multiple pieces of contexts. The candidate entity e_{NN} is considered as a synonym to the query entity e when it receives a higher score $s(e, e_{NN})$ than other non-synonym entities, or exceeds a specific threshold.

3 Experiments

3.1 Experiment Setup

Datasets Three datasets are prepared to show the effectiveness of the proposed model on synonym discovery. The Wiki dataset contains 6.8M documents from Wikipedia² with generic synonym entities obtained from Freebase³. The PubMed is an English dataset where 0.82M research paper abstracts are collected from PubMed⁴ and UMLS⁵ contains existing entity synonym information in the medical domain.

²<https://www.wikipedia.org/>

³<https://developers.google.com/freebase>

⁴<https://www.ncbi.nlm.nih.gov/pubmed>

⁵<https://www.nlm.nih.gov/research/umls/>

Dataset	Wiki + FreeBase	PubMed + UMLS	MedBook + MKG
#ENTITY	9274	6339	32,002
#VALID	394	386	661
#TEST	104	163	468
#SYNSET	4615	708	6600
#CONTEXT	6,839,331	815,644	514,226
#VOCAB	472,834	1,069,061	270,027
#LANGUAGE	English	English	Chinese

Table 1: Dataset Statistics.

The Wiki + FreeBase and PubMed + UMLS are public available datasets used in previous synonym discovery tasks [Qu *et al.*, 2017]. The MedBook is a Chinese dataset collected by authors where we collect 0.51M pieces of contexts from Chinese medical textbooks as well as online medical question answering forums. Synonym entities in the medical domain are obtained from MKG, a medical knowledge graph. Table 1 shows the dataset statistics.

Preprocessing Wiki +Freebase and PubMed + UMLS come with entities and synonym entity annotations, we adopt the Stanford CoreNLP package to do the tokenization. For MedBook, a Chinese word segmentation tool Jieba is used to segment the corpus into meaningful phrases. We remove redundant contexts in the corpus and filter out entities if they appear in the corpus less than five times. For entity representations, the proposed model works with various unsupervised word embedding methods. Here for simplicity, we adopt 200-dimensional word vectors using skip-gram [Mikolov *et al.*, 2013]. Context window is set as 5 with a negative sampling of 5 words for training.

Evaluation Metrics For synonym detection using SYNONYMNET and other alternatives, we train the models with existing synonym and randomly sampled entity pairs as negative samples. During testing, we also sample random entity pairs as negative samples to evaluate the performance. *Note that all test synonym entities are from unobserved groups of synonym entities: none of the test entities is observed in the training data.* The area under the curve (AUC) and Mean Average Precision (MAP) are used to evaluate the model. A single-tailed t-test is conducted to evaluate the significance of performance improvements when we compare the proposed SYNONYMNET model with all the other baselines.

For synonym discovery during the inference phase, we obtain candidate entities E_{NN} from K-nearest neighbors of the query entity in the entity embedding space, and rerank them based on the output score $s(e, e_{NN})$ of the SYNONYMNET for each $e_{NN} \in E_{NN}$. We expect candidate entities in the top positions are more likely to be synonym with the query entity. We report the precision at position K (P@K), recall at position K (R@K), and F1 score at position K (F1@K).

Baselines We compare the proposed model with the following alternatives. (1) **word2vec** [Mikolov *et al.*, 2013]: a word embedding approach based on entity representations learned from the skip-gram algorithm. We use the learned word embedding to train a classifier for synonym discovery. A scoring function $Score_D(u, v) = x_u \mathbf{W} x_v^T$ is used as the objective. (2) **GloVe** [Pennington *et al.*, 2014]: another word embedding approach. The entity representations are learned based on the GloVe algorithm. The classifier is trained with the

MODEL	Wiki + Freebase		PubMed + UMLS		MedBook + MKG	
	AUC	MAP	AUC	MAP	AUC	MAP
word2vec	0.9272	0.9371	0.9301	0.9422	0.9393	0.9418
GloVe	0.9188	0.9295	0.8890	0.8869	0.7250	0.7049
SRN	0.8864	0.9134	0.9517	0.9559	0.9419	0.9545
MaLSTM	0.9178	0.9413	0.8151	0.8554	0.8532	0.8833
DPE	0.9461	0.9573	0.9513	0.9623	0.9479	0.9559
SYNONYMNET (Pairwise)	0.9831 [†]	0.9818 [†]	0.9838[†]	0.9872[†]	0.9685	0.9673
w/o Leaky Unit	0.9827 [†]	0.9817 [†]	0.9815 [†]	0.9847 [†]	0.9667	0.9651
with Bi-LSTM Encoder	0.9683 [†]	0.9625 [†]	0.9495	0.9456	0.9311	0.9156
SYNONYMNET (Triplet)	0.9877[†]	0.9892[†]	0.9788 [†]	0.9800 [†]	0.9410	0.9230
w/o Leaky Unit	0.9705 [†]	0.9631 [†]	0.9779 [†]	0.9821 [†]	0.9359	0.9214
with Bi-LSTM Encoder	0.9582 [†]	0.9531 [†]	0.9412	0.9288	0.9047	0.8867

Table 2: Test performance in AUC and MAP on three datasets. † indicates the significant improvement over all baselines ($p < 0.05$).

same scoring function $Score_D$, but with the learned glove embedding for synonym discovery. (3) **SRN** [Neculoiu *et al.*, 2016]: a character-level approach that uses a siamese multi-layer bi-directional recurrent neural networks to encode the entity as a sequence of characters. The hidden states are averaged to get an entity representation. Cosine similarity is used in the objective. (4) **MaLSTM** [Mueller and Thyagarajan, 2016]: another character-level approach. We adopt MaLSTM by feeding the character-level sequence to the model. Unlike SRN that uses Bi-LSTM, MaLSTM uses a single direction LSTM and $l-1$ norm is used to measure the distance between two entities. (5) **DPE** [Qu *et al.*, 2017]: a model that utilizes dependency parsing results as the structured annotation on a single piece of context for synonym discovery.

3.2 Performance Evaluation

We report Area Under the Curve (AUC) and Mean Average Precision (MAP) in Table 2. From the upper part of Table 2 we can see that SYNONYMNET performances consistently better than those from baselines on three datasets. SYNONYMNET with the triplet training objective achieves the best performance on Wiki +Freebase, while the Siamese objective works better on PubMed + UMLS and MedBook + MKG. Word2vec is generally performing better than GloVe. SRNs achieve decent performance on PubMed + UMLS and MedBook + MKG. This is probably because the synonym entities obtained from the medical domain tend to share more character-level similarities, such as 6-aminohexanoic acid and aminocaproic acid. However, even if the character-level features are not explicitly used in our model, our model still performs better, by exploiting multiple pieces of contexts effectively. DPE has the best performance among other baselines, by annotating each piece of context with dependency parsing results. However, the dependency parsing results could be error-prone for the synonym discovery task, especially when two entities share the similar usage but with different semantics, such as *NBA finals* and *NFL playoffs*. Table 4 reports the performance on Synonym Discovery in P@K, R@K, and F1@K.

We conduct statistical significance tests to validate the performance improvement. The single-tailed t-test is performed for all experiments, which measures whether or not the results from the proposed model are significantly better than ones from baselines. The numbers with † markers in Table 2 indicate that the improvement is significant with $p < 0.05$.

Table 3 shows a case for entity UNGA. In the upper part of

Table 3, candidate entities are generated with nearest neighbor search on pretrained word embeddings using skip-gram. The lower part of Table 3 shows the discovered synonym entities by refining the candidates using the proposed SYNONYMNET model, where a threshold score of 0.8 is used.

CANDIDATE ENTITIES	COSINE SIMILARITY
united_nations_general_assembly m.07vp7	0.847374
un_human_rights_council	0.823727
the_united_nations_general_assembly	0.813736
un_security_council m.07vnr	0.794973
palestine_national_council	0.791135
world_health_assembly m.05_g19	0.790837
united_nations_security_council m.07vnr	0.787999
general_assembly_resolution	0.784581
the_un_security_council	0.784280
ctbt	0.777627
north_atlantic_council m.05pmgy	0.775703
resolution_1441	0.773064
non-binding_resolution m.02pj22f	0.771475
unga m.07vp7	0.770623
FINAL ENTITIES	SYNONYMNET SCORE
united_nations_general_assembly m.07vp7	0.842602
the_united_nations_general_assembly	0.801745
unga m.07vp7	0.800719

Table 3: Candidate entities retrieved using nearest neighbors on Word2vec (upper) and the discovered synonym entities using SYNONYMNET for UNGA (lower).

Ablation Study To study the contribution of different modules of SYNONYMNET for synonym discovery, we also report ablation test results in the lower part of Table 2. “with Bi-LSTM Encoder” uses Bi-LSTM as the context encoder. The last hidden states in both forward and backward directions in Bi-LSTM are concatenated; “w/o Leaky Unit” does not have the ability to ignore uninformative contexts during the bilateral matching process: all contexts retrieved based on the entity, whether informative or not, are utilized in bilateral matching. From the lower part of Table 2 we can see that both modules (Leaky Unit and the Context Encoder) contribute to the effectiveness of the model. The leaky unit contributes 1.72% improvement in AUC and 2.61% improvement in MAP on the Wiki dataset when trained with the triplet objective. The Context Encoder gives the model an average of 3.17% improvement in AUC on all three datasets, and up to 5.17% improvement in MAP.

Hyperparameters We train the proposed model with a wide range of hyperparameter configurations, as shown in Table 5. For the model architecture, we vary the number of randomly sampled contexts $P = Q$ for each entity from 1 to 20. Each piece of context is chunked by a maximum length of T . For the context encoder, we vary the hidden dimension d_{CE} from 8 to 1024. The margin value m in triplet loss function is varied from 0.1 to 1.75. For the training, we try different optimizers, vary batch sizes and learning rates. We apply random search to obtain the best-performing hyperparameter setting

	Wiki + Freebase			PubMed + UMLS			MedBook + MedKG		
	P@K	R@K	F1@K	P@K	R@K	F1@K	P@K	R@K	F1@K
K=1	0.3455	0.3455	0.3455	0.2400	0.0867	0.1253	0.3051	0.2294	0.2486
K=5	0.1818	0.9091	0.3030	0.2880	0.7967	0.3949	0.2388	0.8735	0.3536
K=10	0.1000	1.0000	0.1818	0.1800	1.0000	0.2915	0.1418	1.0000	0.2360

Table 4: Performance on Synonym Discovery.

on the validation dataset, listed in Table 6.

HYPERPARAMETERS	VALUE
P (context number)	{1, 3, 5, 10, 15, 20}
T (maximum context length)	{10, 30, 50, 80}
d_{CE} (layer size)	{8, 16, 32, 64, 128, 256, 512, 1024}
m (margin)	{0.1, 0.25, 0.5, 0.75, 1.25, 1.5, 1.75}
Optimizer	{Adam, RMSProp, Adadelta, Adagrad}
Batch Size	{4, 8, 16, 32, 64, 128}
Learning Rate	{0.0003, 0.0001, 0.001, 0.01}

Table 5: Hyperparameter settings.

DATASETS	P	T	d_{CE}	m	Optimizer	Batch	LR
Wiki + Freebase	20	50	256	0.75	Adam	16	0.0003
PubMed + UMLS	20	50	512	0.5	Adam	16	0.0003
MedBook + MKG	5	80	256	0.75	Adam	16	0.0001

Table 6: Hyperparameters.

Furthermore, we provide sensitivity analysis of the proposed model with different hyperparameters in Wiki + Freebase dataset in Figure 3. Figure 3 shows the performance curves when we vary one hyperparameter while keeping the remaining fixed. As the number of contexts P increases, the model generally performs better. Due to limitations on computing resources, we are only able to verify the performance of up to 20 pieces of randomly sampled contexts. The model achieves the best AUC and MAP when the maximum context length $T = 50$; longer contexts may introduce noise while shorter contexts may be less informative.

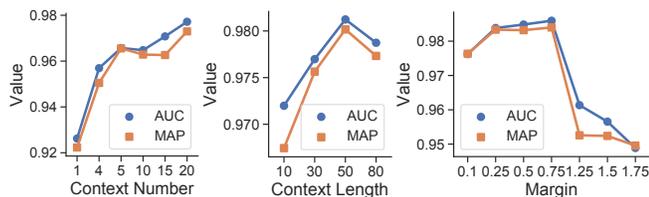


Figure 3: Sensitivity analysis.

4 Related Works

Synonym Discovery The synonym discovery focuses on detecting entity synonyms. Most existing works try to achieve this goal by learning from structured information such as query logs [Ren and Cheng, 2015; Chaudhuri *et al.*, 2009; Wei *et al.*, 2009]. While in this work, we focus on synonym discovery from free-text natural language contexts, which requires less annotation and is more challenging. Some existing works try to detect entity synonyms by entity-level similarities [Lin *et al.*, 2003; Roller *et al.*, 2014; Neculoiu *et al.*, 2016; Wieting *et al.*, 2016]. For example, distributional features are introduced in [Roller *et al.*, 2014] for hypernym detection. Character-level encoding approaches such as [Neculoiu *et al.*, 2016] treat each entity as a sequence of characters, and use a Bi-LSTM to encode the entity information. Such approach may be helpful for synonyms with similar spellings, or abbreviations. Without considering the context information, it is hard for the aforementioned methods to infer synonyms that share similar semantics but are

not alike verbatim. Various approaches [Snow *et al.*, 2005; Sun and Grishman, 2010; Liao *et al.*, 2017; Cambria *et al.*, 2018] are proposed to incorporate context information to characterize entity mentions. These models are not designed for synonym discovery. Dependency parsing result and manually crafted rules on the contexts are used in [Qu *et al.*, 2017] as the structured annotations for synonym discovery. [Mudgal *et al.*, 2018; Kasai *et al.*, 2019] assume that entities are given as structured records extracted from texts, where each entity record provides contextual information about the entity. The goal is to determine whether two entities are the same by comparing and aligning their attributes. We discover synonym entities without such structured annotations.

Sentence Matching There is another related research area that studies sentence matching. Early works try to learn a meaningful single vector to represent the sentence [Tan *et al.*, 2015; Mueller and Thyagarajan, 2016]. DSSM style convolution encoders are adopted in [Huang *et al.*, 2013; Shen *et al.*, 2014; Palangi *et al.*, 2016] to learn sentence representations. They utilize user click-through data and learn query/document embeddings for information retrieval and web search ranking tasks. Although the above methods achieve decent performance on sentence-level matching, the sentence matching task is different from context modeling for synonym discovery in essence. Context matching focuses on local information, while the overall sentence could contain much more information, which is useful to represent the sentence-level semantics, but can be quite noisy for context modeling. Matching schemes on multiple instances with varying granularities are introduced in [Wang and Jiang, 2017; Wang *et al.*, 2016; Wang *et al.*, 2017]. However, these models do not consider the word-level interactions from two sentences during the matching. Sentence matching models do not explicitly deal with uninformative instances. In context matching, missing such property could be unsatisfactory as noisy contexts exist among multiple contexts for an entity. We adopt a bilateral matching which involves a leaky unit to explicitly deal with uninformative contexts while preserving the expression diversity from multiple pieces of contexts.

5 Conclusions

In this paper, we propose a framework for synonym discovery from free-text corpus in an open-world setting. A novel neural network model SYNONYMNET is introduced for synonym detection, which tries to determine whether or not two given entities are synonym with each other. SYNONYMNET makes use of multiple pieces of contexts in which each entity is mentioned, and compares the context-level similarity via a bilateral matching schema to determine synonymity. Experiments on three real-world datasets show that the proposed method SYNONYMNET has the ability to discover synonym entities effectively on both generic and domain-specific datasets with an improvement up to 4.16% in AUC and 3.19% in MAP.

Acknowledgments

We thank the reviewers for their valuable comments. This work is supported in part by NSF under grants III-1526499, III-1763325, III-1909323, and CNS-1930941.

References

- [Cambria *et al.*, 2018] Erik Cambria, Soujanya Poria, Devamanyu Hazarika, and Kenneth Kwok. Senticnet 5: discovering conceptual primitives for sentiment analysis by means of context embeddings. In *AAAI*, 2018.
- [Chaudhuri *et al.*, 2009] Surajit Chaudhuri, Venkatesh Ganti, and Dong Xin. Exploiting web search to generate synonyms for entities. In *WWW*, pages 151–160, 2009.
- [Firth, 1957] John R Firth. A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*, 1957.
- [Harris, 1954] Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.
- [Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [Huang *et al.*, 2013] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. Learning deep structured semantic models for web search using click-through data. In *CIKM*, pages 2333–2338, 2013.
- [Kasai *et al.*, 2019] Jungo Kasai, Kun Qian, Sairam Gurajada, Yunyao Li, and Lucian Popa. Low-resource deep entity resolution with transfer and active learning. *ACL*, 2019.
- [Liao *et al.*, 2017] Zhen Liao, Xinying Song, Yelong Shen, Saekoo Lee, Jianfeng Gao, and Ciya Liao. Deep context modeling for web query entity disambiguation. In *CIKM*, pages 1757–1765, 2017.
- [Lin *et al.*, 2003] Dekang Lin, Shaojun Zhao, Lijuan Qin, and Ming Zhou. Identifying synonyms among distributionally similar words. In *IJCAI*, volume 3, pages 1492–1493, 2003.
- [Mikolov *et al.*, 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013.
- [Mudgal *et al.*, 2018] Sidharth Mudgal, Han Li, Theodoros Rekatsinas, AnHai Doan, Youngchoon Park, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, and Vijay Raghavendra. Deep learning for entity matching: A design space exploration. In *SIGMOD*, pages 19–34, 2018.
- [Mueller and Thyagarajan, 2016] Jonas Mueller and Aditya Thyagarajan. Siamese recurrent architectures for learning sentence similarity. In *AAAI*, pages 2786–2792, 2016.
- [Neculoiu *et al.*, 2016] Paul Neculoiu, Maarten Versteegh, and Mihai Rotaru. Learning text similarity with siamese recurrent networks. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 148–157, 2016.
- [Palangi *et al.*, 2016] Hamid Palangi, Li Deng, Yelong Shen, Jianfeng Gao, Xiaodong He, Jianshu Chen, Xinying Song, and Rabab Ward. Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *TASLP*, 24(4):694–707, 2016.
- [Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014.
- [Peters *et al.*, 2018] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *NAACL*, volume 1, pages 2227–2237, 2018.
- [Qu *et al.*, 2017] Meng Qu, Xiang Ren, and Jiawei Han. Automatic synonym discovery with knowledge bases. In *KDD*, pages 997–1005, 2017.
- [Ren and Cheng, 2015] Xiang Ren and Tao Cheng. Synonym discovery for structured entities on heterogeneous graphs. In *WWW*, pages 443–453, 2015.
- [Roller *et al.*, 2014] Stephen Roller, Katrin Erk, and Gemma Boleda. Inclusive yet selective: Supervised distributional hypernymy detection. In *COLING*, pages 1025–1036, 2014.
- [Shen *et al.*, 2014] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. Learning semantic representations using convolutional neural networks for web search. In *WWW*, pages 373–374, 2014.
- [Snow *et al.*, 2005] Rion Snow, Daniel Jurafsky, and Andrew Y Ng. Learning syntactic patterns for automatic hypernym discovery. In *NIPS*, pages 1297–1304, 2005.
- [Sun and Grishman, 2010] Ang Sun and Ralph Grishman. Semi-supervised semantic pattern discovery with guidance from unsupervised pattern clusters. In *COLING*, pages 1194–1202, 2010.
- [Tan *et al.*, 2015] Ming Tan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. Lstm-based deep learning models for non-factoid answer selection. *arXiv preprint arXiv:1511.04108*, 2015.
- [Wang and Jiang, 2017] Shuohang Wang and Jing Jiang. A compare-aggregate model for matching text sequences. *ICLR*, 2017.
- [Wang *et al.*, 2016] Zhiguo Wang, Haitao Mi, Wael Hamza, and Radu Florian. Multi-perspective context matching for machine comprehension. *arXiv preprint arXiv:1612.04211*, 2016.
- [Wang *et al.*, 2017] Zhiguo Wang, Wael Hamza, and Radu Florian. Bilateral multi-perspective matching for natural language sentences. In *IJCAI*, pages 4144–4150, 2017.
- [Wei *et al.*, 2009] Xing Wei, Fuchun Peng, Huihsin Tseng, Yumao Lu, and Benoit Dumoulin. Context sensitive synonym discovery for web search queries. In *CIKM*, pages 1585–1588, 2009.
- [Wieting *et al.*, 2016] John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. Charagram: Embedding words and sentences via character n-grams. *arXiv preprint arXiv:1607.02789*, 2016.