Evaluating the Impact of Knowledge Graph Context on Entity Disambiguation Models

Isaiah Onando Mulang
isaiam.mulang.onando@iais.fraunhofer.de
Fraunhofer IAIS and Zerotha Research, Germany

Kuldeep Singh
kuldeep.singh1@cerence.com
Cerence GmbH and Zerotha Research
Germany

Chaitali Prabhu
s6chprab@uni-bonn.de
University of Bonn, Germany

Abhishek Nadgeri
abhishhek22596@gmail.com
Zerotha Research, India

Johannes Hoffart
johannes.hoffart@gs.com
Goldman Sachs, Germany

Jens Lehmann
jens.lehmann@cs.uni-bonn.de
University of Bonn, Germany

ABSTRACT

Pretrained Transformer models have emerged as state-of-the-art approaches that learn contextual information from text to improve the performance of several NLP tasks. These models, albeit powerful, still require specialized knowledge in specific scenarios. In this paper, we argue that context derived from a knowledge graph (in our case: Wikidata) provides enough signals to inform pretrained transformer models and improve their performance for named entity disambiguation (NED) on Wikidata KG. We further hypothesize that our proposed KG context can be standardized for Wikipedia, and we evaluate the impact of KG context on state-of-the-art NED model for the Wikipedia knowledge base. Our empirical results validate that the proposed KG context can be generalized (for Wikipedia), and providing KG context in transformer architectures considerably outperforms the existing baselines, including the vanilla transformer models.

ACM Reference Format:

1 INTRODUCTION

Entity Linking (EL) generally consists of two subtasks namely: surface form extraction (mention detection) and named entity disambiguation (NED). A surface form is a contiguous span of text that refers to a named entity. The NED task aims to link the identified named entity to ground truth entities in a given knowledge base [26]. For a long time, researchers focused on NED tasks over semi-structured knowledge repositories such as Wikipedia1 or publicly available KGs such as DBpedia [21], Freebase [2], and YAGO [27]. Wikidata [30] has recently attracted the community’s attention as a rich source of knowledge, and new approaches have been developed to target NED over Wikidata [3].

Motivation: A peculiarity of Wikidata is that the contents are collaboratively edited. As of April 2020, Wikidata contains 83,151,903 items and a total of over 1.2B edits since the project launch2. Considering Wikidata is collaboratively edited, the user-created entities add additional noise and non standard labels (e.g. labels have several numeric and special, non-alphanumeric ASCII characters, also contains multi word labels up to 62 words, etc) [7] since users do not follow a strict naming convention. For instance, there are 17,88,134 labels in which each label matches with at least two different URLs. Hence, NED on Wikidata is quite challenging as pointed out by initial studies [7, 24]. For example, consider the sentence from Wikidata-Disamb [3] dataset: "the short highway in New South Wales and the Australian Capital Territory in Australia, it is part of Sydney-Canberra National Highway link". The entity surface form National Highway matches four(4) different entities in Wikidata that share the same entity label (i.e., "National Highway") while 2,055 other entities contain the whole mention in their labels. The correct entity wikidata:Q19672983 refers to Highway System of Australia, whereas wikidata:Q1967342 refers to the highway system in India. Having these two entities as candidates may require extra information in addition to the surface form or the sentence context. Attention-based Neural Networks [29] and pretrained transformer models [13, 16], have provided an avenue for encoding the context within text, howbeit, in cases such as our example, pure textual context may not be sufficient [11]. As such, a method to obtain matching contextual information from the KG itself could be beneficial to disambiguate in such close scenarios. Inspired by the work of [3] and inherited Wikidata NED challenges (nonstandard multi-word, long, implicit, case-sensitive), we hypothesize that the performance of pretrained transformer models improves by considering further context from the KG. We investigate three research questions: RQ1: How does applying KG context impact the performance of transformer models on NED over Wikidata? RQ2: What is the performance of different configurations of KG context as new information signals on the NED task? RQ3: Can we generalize our proposed context in a state of the art NED model for other knowledge bases such as Wikipedia? The structure of the paper is follows: next section defines the task followed by

---

1https://www.wikipedia.org/
2https://www.wikidata.org/wiki/Wikidata:Statistics
3wikidata:Q1967298 binds to https://www.wikidata.org/wiki/Q1967298
related work in 3. Section 4 describes approach and we present experiments in section 5. We conclude in section 6.

2 TASK DEFINITION

Given a sentence, a recognized entity surface form, a set of candidate entities, and a Knowledge Graph (KG), the objective is to select the entity within the KG that matches the surface form in the text. A sentence \( \mathcal{D} = \{w_1, w_2, ..., w_n\} \) is a set of tokens of length \( n \). The set of entity surface forms \( \mathcal{S} = \{s_1, s_2, ..., s_k\} \) where \( s_i = (w' \in \mathcal{D}) \) contains recognized entities s.t. each \( s_i \) spans one or more tokens in \( \mathcal{D} \). We view a KG as a labeled directed graph. A KG \( \mathcal{KG} = (\mathcal{E}, \mathcal{H}^+, \mathcal{R}) \) is a 3-tuple where: i) set \( \mathcal{E} \) of all entities represent the vertices. ii) set \( \mathcal{R} \) is the set of all edges between the entity instances in the graph. And; iii) \( \mathcal{H}^+ \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E} \) is the ordered set of all triples. The function \( \ell(a \in \mathcal{E} \cup \mathcal{R}) \) is defined that retrieves labels of any given entity or relation from the KG.

Candidate Entities: A set containing selected entities from the KG given as: \( \mathcal{E}' = \{e'_1, e'_2, ..., e'_m\} \) where \( e'_j \in \mathcal{E} \) and \( \mathcal{E}' = \sigma(s_i \in \mathcal{S}) \), is obtained by a semantic selection operation \( \sigma \) on a given surface form. This paper addresses the problem of named entity disambiguation which selects an entity \( e^* \in \mathcal{E}' \) that matches the textual mention \( s \in \mathcal{S} \). We view this task as a classification \( f = \text{Classify}(h(X)) \) on the conditional probability \( h(X) = P(Y = 1|X) \). Taking \( x \in X = (s, e^*; \theta) \) we study configurations of the context parameters \( \theta \).

3 RELATED WORK

There is a wide variety of approaches in the literature for Entity Linking (EL) ranging from graph traversal [28], rule-based [23], to Neural Network-based approaches [7, 19]. For detailed information on entity linking we refer to the surveys in [1, 25]. Herein, we restrict ourselves to very closely related recent literature. Advances in Neural Networks and the introduction of self-attention based techniques [29] that allow for encoding of contextual meaning from text have advanced research in EL. Work in [18] improved EL performance by proposing a new approach that combines deep learning with more traditional methods such as graphical models and probabilistic mention-entity maps. Research in [19] aims to achieve an end-to-end EL through context-aware mention embeddings, entity embeddings, and a probabilistic mentions.

In the meantime, research on the learning of contextual data has advanced in two directions. On one hand, the powerful pretrained transformer models [13, 16] have emerged as state-of-the-art for representing context within text and have seen burgeoning reuse through fine-tuning for several NLP tasks [4]. On the other hand, KGs are increasingly being seen as a source of additional knowledge for Neural Networks. For instance researchers in [6] recently released an embedding library for the Wikidata KG while the work by [11] introduced an extension of BERT (KBERT) in which KG triples are injected into the sentences as domain knowledge. Specific to the EL task, the work by [7] employs information from a locally derived KG to improve the performance of end-to-end EL using attention-based Neural Networks. Researchers in [3] fetched a significant amount (as high as 1500) of 2-hop KG triples and used Recurrent Neural Networks (RNN) to encode this information. In this paper, we argue that a slight amount of KG triple context is enough for a pretrained transformer. We study the different configuration of KG context on transformer models to target NED on Wikidata.

4 APPROACH

Figure 1 illustrates the overall approach. For the classification : \( f(h(s, e^*; \theta)) = y \) such that \( s \), the mentioned surface form, and \( e^* \), the candidate entity, are known. A set of contextual parameters \( \theta \) is then provided to the model. By adding the original sentence as part of the input, we let the model learn source context. Our approach then models a set of information from the target KG in the form of KG triples \( \Phi \) as context. The aim is to maximize both the true positives and true negatives such that, for every input, if \( y = 1 \) then the \( e^* \) is the ground truth entity of \( s \) in the KG. The classifier employs the binary cross-entropy loss.

Knowledge Graph Context: We use a SPARQL endpoint to fetch triples of the identified entity in the sentence. There are two sets of triple configurations considered in our experiments, depending on the hop counts from the head entity. The parameter \( \Phi \) is therefore an ordered set of triples \((h^5, r^5, t^5)\) such that \( h^5 \), the head (subject) of any triple is the candidate entity to be classified whereas \( h^5 = 1|2 \) is the hop count. The \( i \) refers to the position of the triple in the set and can range between 1 and over 1000. To formulate our input, we consider the natural language labels of the retrieved triples \( h^5, r^5, t^5 \). A triple is therefore verbalized into i’s natural language form: \( [\text{whitespace}]; [\text{whitespace}]; [\text{whitespace}] \). The sequence of these verbalized triples are appended to the original sentence and surface form delimited by the [SEP] token. Figure 1 shows how the context input is handled such that the Segment Embeddings for every triple is different and provides a unique signal to the tokens at the embeddings layer of the network. When the total number of triples is too many, we use the maximum sequence length to limit the input where the final context representation \( \Phi^{\text{max}} \subseteq \Phi \). The values of \( \Phi^{\text{max}} \) for entity: Q1967298 in figure 2, is given as: [National Highway description highway system in Australia [SEP] National Highway label National Highway [SEP] National Highway date modified 31 May 2019 [SEP]]

5 EVALUATION

Datasets: The first dataset is Wikidata-Disamb30 [3], which aligns Wiki-Disamb30 [17] to Wikidata entities, and adds closely matching entities as negative samples to every entity in the dataset. It consists of 200,000 Train and 20,000 Test samples. We also consider the ISTEX dataset introduced by [5], extracted from scientific publications and contains 1000 author-affiliation strings from research articles aligned to Wikidata. For generalizing the impact of KG context, we choose standard Wikipedia dataset: AIDA-CoNLL [9]. We aligned its Wikipedia entities to corresponding Wikidata mentions to fetch the KG triples. Datasets are accompanied with a pre-computed candidate list.

Baselines: We compare our results with three types of baselines. First is [3], which experimented with numerous configurations of KG context on Long Short Term Memory (LSTM) networks and reported an ablation of these configurations. These models were augmented with a massive amount of 1&2-hop KG triples. We also
run the model on the ISTEX dataset to enable performance comparison. We create a second set of baselines by employing the vanilla transformer models of RoBERTa and XLNet (i.e., transformers without KG context) on Wikidata-Disamb and ISTEX. We fine-tuned vanilla models on Wikidata-Disamb training set. For AIDA-CoNLL, we chose [12] as our underlying model which is the second peer reviewed SOTA on this dataset. Authors used Wikipedia descriptions as a context for candidate entities, and we replaced this context with our proposed 1-hop KG triple context fetched from Wikidata triples of corresponding Wikipedia entities. We verbalized the fetched triples, as described in our approach.

Model Parameters: We chose two state of the art transformer architectures: RoBERTa [13], and XLNet [16] and fine-tune them using Wikidata-Disamb30 training set. We report P,R,F values following the baseline of Wikidata-Disamb and ISTEX dataset. For each vanilla Transformer architecture, we add a classification head. The maximum sequence length for the inputs in both models is viewed vanilla models on Wikidata-Disamb training set. For AIDA-CoNLL, however, the transformer models task-specific model. This RNN model for the data, we publicly release code, datasets, training details, and results for reusability and reproducibility. On AIDA-CoNLL, We use open source implementation of [12] for feeding the KG context.

Figure 1: Overall Approach: Φ refers to the ordered set of triples from the KG for a candidate entity while φmax ∈ Φ, is the maximum number of triples that fits in the sequence length. For brevity: N → 'National', H → 'Highway', desc → 'description'.

Figure 2: KG context: Top three 1-hop triples from Wikidata for the two entities with same label: National Highway.

Model Parameters: We chose two state of the art transformer architectures: RoBERTa [13], and XLNet [16] and fine-tune them using Wikidata-Disamb30 training set. We report P,R,F values following the baseline of Wikidata-Disamb and ISTEX dataset. For each vanilla Transformer architecture, we add a classification head. The maximum sequence length for the inputs in both models is fixed at 512 tokens, and we use this to limit the amount of KG context to feed. We publicly release code, datasets, training details, and results for reusability and reproducibility. On AIDA-CoNLL, We use open source implementation of [12] for feeding the KG context and report In-KB accuracy as prior work(s).

Results and Discussion: Table 1 shows the results from evaluating our approach against the baselines on the Wikidata-Disamb30 and table 2 indicates the performance of the models on the ISTEX dataset. The results in table 2 obtained by running the same model trained on the Wikidata-Disamb30 dataset (also for baseline) with context, but during testing, no more context is provided. Based on our results, we postulate that although the Transformer based language models are trained on huge corpus and possess context for the data, they show limited performance even against the RNN model. This RNN model [3] uses GloVe embeddings together with task-specific context (cf. Table 1). However, the transformer models outperform the baseline models when fed with our proposed KG context. For instance, (cf. Table 1), RoBERTa, with a 1-hop context, can correctly link entities in additional 1127 sample sentences in the test set (flipped from incorrect to the correct predictions) compared to its vanilla setting. These samples have 997 unique Wikidata IDs. These results also indicate that the transformer models achieve better precision compared to recall; this is clear in table 2 and interpreted as follows: our model is more likely to classify an entity as the correct entity only when it is true (few false positives). For brevity, the detailed analysis of each experimental setup and corresponding data can be found in our Github.

Concerning RQ2, our results indicate that including triples from higher hop counts either exhibit an inverse impact on the performance or have minimal effect on overall model behavior (cf. Table 1 RoBERTa vs. XLNet 2-hop values). This signals that the further away we drift from the head entity, the noisier the signal provided by the context added. As such, we did not extend evaluation to higher triple hops. However, we can observe that XLNet shows a more stable behavior in cases when the excess context is provided as it can preserve already learned information. It is in contrast to RoBERTa, which loses necessary signals in an attempt to learn.

Table 1: Comparison of our model against baselines on the Wikidata-Disamb dataset. Best results in dark bold

<table>
<thead>
<tr>
<th>Model</th>
<th>Prec</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM + RNN-triplets [3]</td>
<td>90.1</td>
<td>92.0</td>
<td>91.1</td>
</tr>
<tr>
<td>LSTM+RNN-triplets+ATN[3]</td>
<td>90.2</td>
<td>93.0</td>
<td>91.6</td>
</tr>
<tr>
<td>RoBERTa - without KG context</td>
<td>89.09</td>
<td>84.67</td>
<td>86.23</td>
</tr>
<tr>
<td>RoBERTa-without KG context</td>
<td>89.32</td>
<td>87.62</td>
<td>88.46</td>
</tr>
<tr>
<td>Our Contextual models</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RoBERTa + 1-hop KG Context</td>
<td>91.48</td>
<td>93.23</td>
<td>92.35</td>
</tr>
<tr>
<td>RoBERTa + 1&amp;2-hop KG Context</td>
<td>89.88</td>
<td>87.64</td>
<td>88.75</td>
</tr>
<tr>
<td>XLNet + 1-hop KG Context</td>
<td>91.55</td>
<td>93.14</td>
<td>93.34</td>
</tr>
<tr>
<td>XLNet + 1&amp;2-hop KG Context</td>
<td>91.93</td>
<td>92.36</td>
<td>92.14</td>
</tr>
</tbody>
</table>

Table 2: Our models against baseline on ISTEX dataset

<table>
<thead>
<tr>
<th>Model</th>
<th>Prec</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM + RNN of triplets + ATN [3]</td>
<td>86.32</td>
<td>96.38</td>
<td>90.97</td>
</tr>
<tr>
<td>RoBERTa + 1-hop Triples (Ours)</td>
<td>91.70</td>
<td>91.98</td>
<td>91.84</td>
</tr>
<tr>
<td>XLNet + 1-hop KG Context (Ours)</td>
<td>96.39</td>
<td>89.11</td>
<td>92.61</td>
</tr>
</tbody>
</table>

from the extra context. The amount of data fed as the context in our models is minimal (up to 15 1-hop triples). In contrast, the best performing model from work in [3], was fed up to 1500 1+2-hop triples. Our best performance can then be attributed to the quality of textual context learned by the transformers as well as the optimal choice of KG-triples context.

**Generalizing KG Context:** We induced 1-hop KG context in DCA-SL model [12] for candidate entities. The replacement of the unstructured Wikipedia description with structured KG triple context containing entity aliases, entity types, consolidated entity description, etc. has a positive impact on the performance. Our proposed change (DCA-SL + Triples) outperforms the baselines for Wikipedia entity disambiguation(cf. Table 3). Please note, out of 207,544 total entities of AIDA-CoNLL datasets, 7591 entities have no corresponding Wikidata IDs. Even if we do not feed the KG context for 7591 entities, the performance increases. It validates our third research question (RQ3), and we conclude KG triple context can be standardized for the NED task for Wikipedia.

### 6 CONCLUSION

In this paper, we study three closely related research questions. We demonstrate that pretrained Transformer models, although powerful, are limited to capturing context available purely on the texts concerning the original training corpus. We observe that an extra task-specific KG context improved the performance. However, there is a limit to the number of triples as the context that can improve performance. We note that 2-hop triples resulted in negative or little impact on transformer performance. Our triple context can be generalized (for Wikipedia) and observes a positive effect on the NED model for Wikipedia, leading into a new SOTA for AIDA-CoNLL dataset. For the future work, it would be interesting to understand which triples negatively impact the context and how to select the “optimal choice of KG-triples context,” considering we rely on the triple in the same order of the SPARQL endpoint returned results. As a viable next step, we plan to study independent effect of various KG attributes (entity properties such as aliases, descriptions, Instance-of, etc.) on NED models’ performance.

### 7 ACKNOWLEDGMENTS

This work is co-funded by the Federal Ministry of Education and Research’s (BMBF) Software Campus initiative under the Answer-King Project.

### REFERENCES


