CHOLAN: A Modular Approach for Neural Entity Linking on Wikipedia and Wikidata

Manoj Prabhakar Kannan Ravi[1], **Kuldeep Singh**[3], **Isaiah Onando Mulang'**[2], **Saeedeh Shekarpour**[4], **Johannes Hoffart**[5], **and Jens Lehmann**[2]

[1]Hasso Plattner Institute, University of Potsdam, Potsdam, Germany
manoj.prabhakar@hpi.de
[2]Smart Data Analytics, University of Bonn, Bonn, Germany
{mulang,jens.lehmann}@cs.uni-bonn.de
[3]Zerotha Research and Cerence GmbH, Aachen, Germany
kuldeep.singh1@cerence.com
[4]University of Dayton, Dayton, USA
sshekarpour1@udayton.edu
[5]Goldman Sachs, Frankfurt, Germany
johannes.hoffart@gs.com

Abstract

In this paper, we propose CHOLAN, a modular approach to target end-to-end entity linking (EL) over knowledge bases. CHOLAN consists of a pipeline of two transformer-based models integrated sequentially to accomplish the EL task. The first transformer model identifies surface forms (entity mentions) in a given text. For each mention, a second transformer model is employed to classify the target entity among a predefined candidates list. The latter transformer is fed by an enriched context captured from the sentence (i.e. local context), and entity description gained from Wikipedia. Such external contexts have not been used in state of the art EL approaches. Our empirical study was conducted on two well-known knowledge bases (i.e., Wikidata and Wikipedia). The empirical results suggest that CHOLAN outperforms state-of-the-art approaches on standard datasets such as CoNLL-AIDA, MSNBC, AQUAINT, ACE2004, and T-REx.

1 Introduction

The explicit schema, graph-based structure, and interlinking nature of information represented in publicly available knowledge graphs (KGs) e.g., DBpedia (Auer et al., 2007), Freebase (Bollacker et al., 2007), Wikidata (Vrandecic, 2012) or knowledge bases (KBs) such as Wikipedia; introduce a new landscape of features, as well as structured knowledge and embeddings. Researchers have developed several techniques to align information available in unstructured text to the concepts of these KGs (Wu et al., 2019b; Broscheit, 2019).

End-to-end Entity Linking (hereafter EL) task follows this direction; such that, given a sentence EL first identifies the entity mention in the sentence, then maps these mentions to the most likely KG/KB entities. The EL comprises of a three-step process. With respect to the given example sentence *Soccer: Late Goals Give Japan win Over Syria*, the first step called mention detection (MD) identifies the surface forms *Japan* and *Syria*. The next step is candidate generation (CG) aiming to find a list of possible entity candidates in the KG/KB for each entity mention. For example, the candidates list for entity mention *Japan* consists in part of *Japan national football team, Japan (country), Japan (Band)* and for *Syria* is *Syria (Roman province), Syria national football team, Greater Syria*. Finally, the third step deals with the entity disambiguation (ED) which employs the co-reference and contextual features to discriminate the most likely entity from the candidates list e.g., *Japan national football team* and *Syria national football team* are correct entities.

Entity Linking approaches are broadly categorised into three categories. The initial attempts (Hoffart et al., 2011; Piccinno and Ferragina, 2014) solve MD and ED as independent sub-tasks of EL (i.e., a pipeline based system). However, these approaches exhibit a behaviour where errors propagate from MD to ED hence might downgrade the overall performance of the system. The second category has emerged in an attempt to mitigate these errors, where researchers focused on jointly modelling MD and ED, emphasising the importance of the mutual dependency of the two sub-tasks (Kolitsas et al., 2018). These two EL

Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, pages 504–514
April 19 - 23, 2021.

approaches depend on an intermediate candidate generation step and rely on a pre-computed list of entity candidates. For example, (Kolitsas et al., 2018) propose a joint MD and ED model and inherits the candidate list from (Ganea and Hofmann, 2017). The third approach combines the three sub-steps in a joint model and illustrates that each of those tasks is interdependent (Durrett and Klein, 2014; Broscheit, 2019).

The recent EL approaches focus on jointly modelling two or three subtasks (Sevgili et al., 2020). Furthermore, the NLP research community has extensively used transformers in end-to-end models for entity linking (Broscheit 2019, Peters et al. 2019, and Févry et al. 2020). Nevertheless, these works report less performance than (Kolitsas et al., 2018), which is a bi-LSTM based model. The observations regarding the limited performance of transformer-based models for the EL motivate our work, and in this paper, our focus is to understand the bottlenecks in the entity linking process. We argue that the less studied task in literature, i.e., candidate generation, has an essential role in the EL models' performance, which has not been a focus in the recently proposed transformer-based entity linking models.

In this paper, we **hypothesise** that the transformer models, though trained on a large corpus, may require additional task-specific contexts. Furthermore, inducing the context at the entity disambiguation step may positively impact the overall performance, which has not been utilised in the state of the art methods due to monolithic implementations (Kolitsas et al., 2018; Peters et al., 2019; Broscheit, 2019; Févry et al., 2020). Subsequently, we deviate from the joint modelling of two or three subtasks of the EL and revert to the methodology opted by earlier EL systems in 2011 (Hoffart et al., 2011), i.e. treat each subtask independently. As such, we study the research question: **RQ**: *what is the impact of each sub-task (aka component) on the overall outcome of the transformer-based entity linking approach?* We propose an intuitive novel approach named CHOLAN, comprising a modular architecture of two transformer models to solve MD and ED independently. In the first step, CHOLAN employs BERT (Devlin et al., 2019) model to identify mentions of the entities in an input sentence. The second step involves expanding each mention with a list of KB entity candidates. Finally, the entity mention, sentence (local context), an entity candidate, and entity Wikipedia description (entity context) are fed as input sequences in the second BERT based model to predict the correct KB entity (cf. Figure 1). We train MD and ED steps independently during training, and while testing, we run the CHOLAN pipeline end-to-end for predicting the KB entity. The following are the novel features of CHOLAN:

- The core focus of the approach is to flexibly induce external context and candidate lists in a transformer-based model to improve the EL performance. CHOLAN is independent of a particular candidate list and additional background context. We study four different configurations of CHOLAN to demonstrate the impact of candidate generation step and background knowledge (i.e. entity and sentential context) induced in the model. CHOLAN achieves a new state of the art performance on several datasets: T-REx (ElSahar et al., 2018) for Wikidata; AIDA-B, MSBC, AQUAINT, and ACE2004 for Wikipedia (Hoffart et al., 2011; Guo and Barbosa, 2018).
- CHOLAN is the first approach which is empirically demonstrated to be transferable across KBs having completely different underlying structure and schema i.e., on semi-structured Wikipedia and fully structured Wikidata.

The implementation is publicly available[1]. The paper is structured as follows: next section summarises the related work. Section 3 describes the problem statement and approach. Section 4 explains the experimental settings followed by results in 5. We conclude in Section 6.

2 Related Work

Mention Detection (MD): The first attempt to organise a named entity recognition (NER) task traced back to 1996 (Grishman and Sundheim, 1996). Since then, numerous attempts have been made ranging from conditional random fields (CRFs) with features constructed from dictionaries (Rocktäschel et al., 2013) or feature-inferring neural networks (Collobert and Weston, 2008).

[1] https://github.com/ManojPrabhakar/CHOLAN

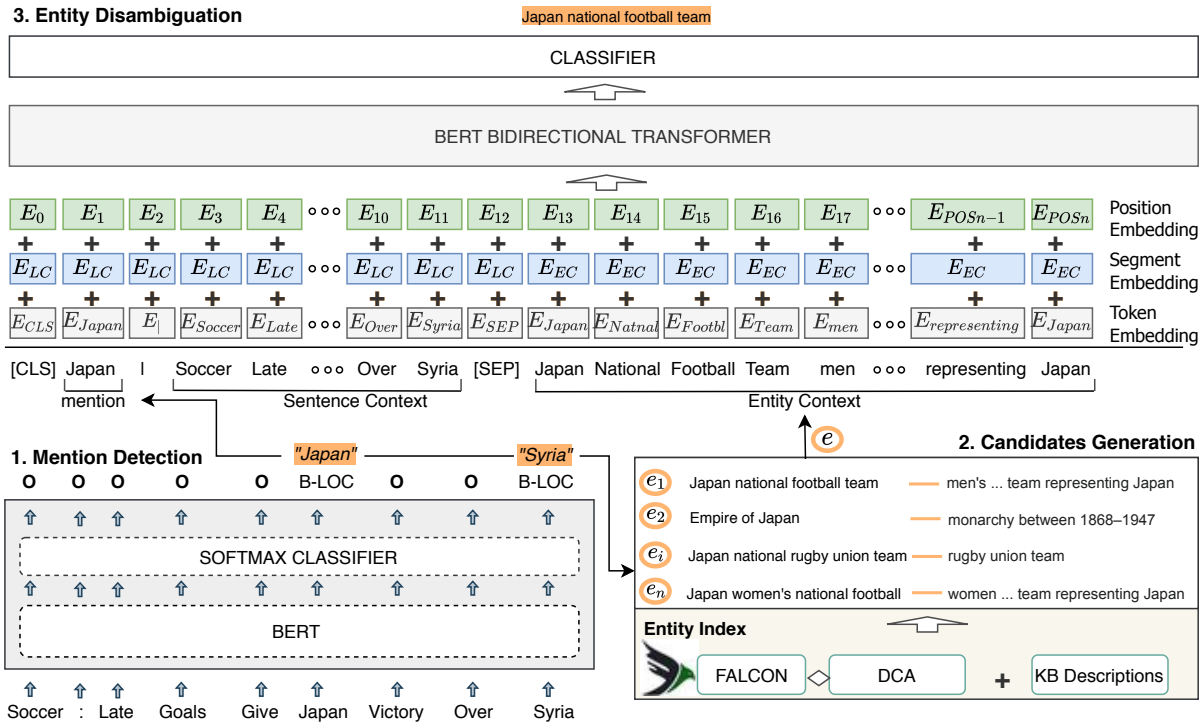

Figure 1: CHOLAN has three building blocks: i) BERT-based Mention Detection that identifies entity mentions in the text ii) Candidate Generation that retrieves a set of entities for the mention iii) Entity Disambiguation: employs BERT transformer model powered by background knowledge from KB and local sentential context.

Recently, contextual embedding based models achieve state of the art for NER/MD task (Akbik et al., 2018; Devlin et al., 2019). We point to the survey by Yadav and Bethard (2018) for details about NER. Few early EL models have performed MD task independently (Ceccarelli et al., 2013; Cornolti et al., 2016).

Candidate Generation (CG): There are four prominent approaches for candidate generation. First is a direct matching of entity mentions with a pre-computed candidate set (Zwicklbauer et al., 2016). The second approach is the dictionary lookup, where a dictionary of the associated aliases of entity mentions is compiled from several knowledge base sources (e.g. Wikipedia, Wordnet) (Sevgili et al., 2020; Fang et al., 2019; Cao et al., 2017). The third approach is to generate entity candidates using empirical probabilistic entity-map $p(e|m)$. The $p(e|m)$ is a pre-calculated prior probability of correspondence between positive mentions and entities. A widely used entity map was built by (Ganea and Hofmann, 2017) from Wikipedia hyperlinks, Crosswikis (Spitkovsky and Chang, 2012) and YAGO (Hoffart et al., 2011) dictionaries. End-to-end EL approaches such as (Kolitsas et al., 2018; Cao et al., 2018) relies on the entity map built by Ganea and Hofmann. The next approach for generating the candidates is proposed by (Sakor et al., 2019). Authors build a local KG by expanding entity mentions using Wikidata and DBpedia entity labels and associated aliases. The local KG can be queried using BM25 ranking algorithm (Logeswaran et al., 2019). The modular architecture of CHOLAN gives us the flexibility to experiment with several ways of generating entity candidates. Hence, we reused candidate list proposed by (Ganea and Hofmann, 2017) and built a new CG approach based on (Sakor et al., 2019).

End to End EL: Few EL approaches accomplish MD and ED tasks jointly. (Nguyen et al., 2016) propose joint recognition and disambiguation of named-entity mentions using a graphical model and show that it improves EL. The work in (Kolitsas et al., 2018) also proposes a joint model for MD and ED. Authors use a bi-LSTM based model for mention detection and computes the similarity between the entity mention embedding and set of predefined entity candidates. The work in (Broscheit, 2019) employs BERT to jointly model three subtasks of the EL. Author employ an entity vocabulary of 700K top most frequent entities to train the model. Work in (Févry et al., 2020) uses a Transformer architecture with large scale pre-

training from Wikipedia links for EL. For CG, authors train the model to predict BIO-tagged mention boundaries to disambiguate among all entities. For Wikidata KG, Opentapioca is an entity linking approach which relies on a heuristic-based model for disambiguation of the mentions in a text to the Wikidata entities (Delpeuch, 2020). Arjun (Mulang et al., 2020) is the most similar to our approach CHOLAN and trains two independent neural models for MD and ED. It generates candidates on the fly using a Wikidata entity alias map. Arjun does not induce any context in the model.

3 Problem Statement and Approach

We formally define EL task as follows: given an input sequence of words W = $\{w_1, w_2, w_3, \ldots, w_n\}$, and a set of entities denoted by $\mathcal{E}$ from a KG/KB. The EL task aligns the text into a subset of entities represented as $\Theta : W \rightarrow \mathcal{E}'$ where $\mathcal{E}' \subset \mathcal{E}$. We formulate the EL task as a three step process in which the first step is the mention detection (MD). The MD is a function $\theta_1 : W \rightarrow \mathcal{M}$, where the set of mentions is denoted by $\mathcal{M} = (m_1, m_2, ..., m_k)$ $(k \leq n)$ and each mention m_x is a sequence of words starting from i to end position j: $m_x^{(i,j)} = (w_i, w_{i+1}, ..., w_j)$ $(0 < i, j \leq n)$. The next task is candidate generation where for each mention m_x a set of candidates $C(m_x)$= $\{e_1^x, ..., e_n^x | e_i^x \in \mathcal{E}\}$ is derived. Finally, the entity disambiguation (ED) task aims to map each mention $m_x \in \mathcal{M}$ to the most likely entity from its list of candidates. In our case, we model the ED task as a classification task and augment the input with extra signals as context. For every candidate entity $c_i \in C(m_x)$, the model estimates a probability p_i, thus the most likely entity is the one with the highest probability as $\gamma = \arg\max_{p_i}\{\mathcal{P}(p_i \mid m_x, c_i^x, W, C)\}$ where W and C are the input representations respectively for the given sentence (local context) and the context derived from KG/KB. As such the probability of score p_i is conditioned not only on m_x and c_i^x but also on W and C as contextual parameters.

3.1 CHOLAN Approach

The CHOLAN architecture comprises of three main modules as illustrated in Figure 1.

3.1.1 Mention Detection (MD)

We adapt the vanilla BERT (Devlin et al., 2019) model for the task of entity mention detection in an unstructured text. For each input sentence, we append the special tokens [CLS] and [SEP] to the beginning and end of the sentence, respectively. This is then used as input to the model which learns a representation of the tokens in the sentence. We then introduce a (logistic regression based) classification layer on top of the BERT model to determine named entity tags for each token following the BIO format (Sang and Meulder, 2003). Our BERT† model is initialised using publicly available weights from the pretrained BERT$_{BASE}$ model and is fine-tuned to the specific dataset for detecting a mention m_i. Please note that BERT$_{BASE}$ model is the latest approach which successfully outperformed in various NLP tasks, including MD. Thus, we reuse this model for the completion of our approach.

$$m_i = BERT^\dagger(w_i) \quad (1)$$

3.1.2 Candidate Generation (CG)

One of the critical focus of CHOLAN is to understand the bottleneck at the CG step. Hence, we reuse the DCA candidate list and propose a novel candidate list to understand the candidate generation impact on overall EL performance.

DCA Candidates: (Yang et al., 2019) adapts the probabilistic entity-map $p(e|m)$ created by (Ganea and Hofmann, 2017) (cf. section 2) to calculate the prior probabilities of candidate entities for a given mention. In the probabilistic entity-map, each entity mention has 30 potential entity candidates. Yang and colleagues also provide associated Wikipedia description of each entity. In CHOLAN, we *reuse* candidate set $C(m)$ provided by (Yang et al., 2019) and further consider associated Wikipedia entity descriptions.

Falcon Candidates: (Sakor et al., 2019) created a local index of KG items from Wikidata entities expanded with entity aliases. For example, in Wikidata the entity Q33[2] has the label "Finland". Sakor and colleagues expanded the entity label with other aliases from Wikidata such as "Finlande", "Finnia", "Land of Thousand Lakes", "Suomi", and "Suomen tasavalta". We adopt this local KG index to generate entity candidates per entity mention in the employed datasets. The local KG has a querying mechanism using BM25† algorithm (cf. equation (2)) and ranked by the calculated score. We build a predefined candidate set using the top 30 Wikidata entity candidates in

[2] https://www.wikidata.org/wiki/Q33

$C\_Falcon(m)$ for each entity mention. We enrich the candidates set obtained from Wikidata by the correspondence from Wikipedia. We also add the first paragraph of Wikipedia as entity descriptions (only if Wikidata entity has corresponding Wikipedia page) to the hyperlinks. By selecting two different candidate list, our idea is to understand the impact of candidate generation step on end-to-end entity linking performance.

$$e_i = BM25^{\dagger}(m_i) \quad (2)$$

3.1.3 Entity Disambiguation (ED)

In order to use the power of the transformers, we propose "WikiBERT" to perform the ED task. In WikiBERT, our novel methodological contribution is the induction of local sentential context and global entity context at the ED step in a transformer model, which has not been used in the recent EL models. WikiBERT is derived from the vanilla BERT$_{BASE}$ model and fine-tuned on the two EL datasets (CoNLL-AIDA and T-REx). We view the ED task as sequence classification task. The input to our model is a combination of two sequences. The first sequence S_1 concatenates the entity mention $m \in \mathcal{M}$ and sentence $\mathcal{W}$ where the sentence acts as a local context. The second sequence S_2 is a concatenation of entity candidate $e \in C(m)/C\_Falcon(m)$(obtained from Equation 2) and its corresponding Wikipedia description (entity context ct_i). The two sequences are paired together with special start and separator tokens: ([CLS] S_1 [SEP] S_2 [SEP]). The sequences are fed into the model which in turn learns the input representations according to the architecture of BERT (Devlin et al., 2019). Any given token (local context word, entity mention, or entity context words) is a summation of the three embeddings :

i. *Token embedding*: refers to the embedding of the corresponding token. We make note here on specific tokens that comprises the input representations for our model more specialised as compared to other fine-tuning tasks. The entity mention tokens appended at the beginning of S_1 and separated from the sentence context tokens by a single vertical token bar |, likewise, for the entity context sequence S_2, we prepend the entity title tokens from the KB before adding the descriptions.

ii. *Segment embedding*: each of the sequences receive a single representation such that the segment embedding for the local context E_{LC} refers to the representation for S_1 whereas E_{EC} is the representation of S_2

iii. *Position embedding*: represents the position of the token in an input sequence. A token appearing at the i-th position in the input sequence is represented with E_i

To train the model, we use the negative sampling approach similar to Yamada and Shindo (2019). The candidate list is generated for each identified mention. The desired entity candidate item is labelled as one, and the rest of the incorrect candidate items (from candidate list) are labelled as zero for a given mention. This process iterates over all the identified mentions using Equation 1.

The training process fine-tunes BERT using the contextual input from sentence and Wikipedia resulting into the WikiBERT model (Equation (3)). The model predicts the relatedness of the two sequences by classifying it as either positive or negative.

$$e_i = WikiBERT(m_i, e_i, ct_i) \quad (3)$$

4 Experimental Setup

4.1 Datasets

For Wikidata EL, we rely on T-REx dataset (ElSahar et al., 2018). We adapt the subset of T-REx used by Mulang et al. (2020) for a fair evaluation setting. The dataset contains 983,257 sentences (786,605 in training and 196,652 in the test set) accommodating 3,133,778 instances of surface forms which are linked to 85,628 distinct Wikidata entities. T-REx does not have a separate validation set to fine-tune the hyperparameters. Therefore, we further divide the train set into a 90:10 ratio for training and validation.

For EL over Wikipedia, we adapt standard dataset CoNLL-AIDA proposed by (Hoffart et al., 2011) for the training. The dataset contains 18,448 linked mentions in 946 documents, a test set of 4,485 mentions in 231 documents, and a validation set of 4,791 mentions in 216 documents. For testing, we use AIDA-B (test) dataset from (Hoffart et al., 2011) and MSNBC, AQUAINT, ACE2004 datasets from (Guo and Barbosa, 2018).

4.2 Models for Comparison

4.2.1 Baselines over Wikidata

We now briefly explain Wikidata baselines.

1. OpenTapioca (Delpeuch, 2020): is a heuristic-based end-to-end approach that depends on topic similarity and mapping coherence for linking

Wikidata entity in an input text.
2. Arjun (Mulang et al., 2020): is a pipeline of two attentive neural networks employed for MD and ED. Arjun is the SotA, and we take baseline values from Arjun's paper.

4.2.2 Baselines over Wikipedia

1. (Hoffart et al., 2011): build a weighted graph of entity mentions and candidate entities. Then, the model computes a dense subgraph that predicts the best joint mention-entity mapping.
2. DBpedia Spotlight (Mendes et al., 2011) proposes a probabilistic model and relies on the context of the text to link the entities.
3. KEA (Steinmetz and Sack, 2013) employs a linguistic pipeline coupled with metadata generated from several Web sources. The candidates are ranked using a heuristic approach.
4. Babelfy (Moro et al., 2014) is a graph-based approach that uses loose identification of candidate meanings coupled with the densest subgraph heuristic to link the entities.
5. Piccinno and Ferragina (2014): to solve entity linking, authors focus on mentions recognition and annotations pruning to propose a voting algorithm for entity candidates using PageRank.
6. Kolitsas et al. (2018) train MD and ED task jointly using word and character-level embeddings. The model reuses candidate set from (Ganea and Hofmann, 2017) and generates a global voting score to rank the entity candidates.
7. Peters et al. (2019) induce multiple KBs into a large pretrained BERT model with a knowledge attention mechanism.
8. Broscheit (2019) trains MD, CG, ED task jointly using a BERT-based model. Besides, an entity vocabulary containing 700K most frequent entities in English Wikipedia was utilised.
9. Févry et al. (2020) consider large scale pretraining from Wikipedia links as the context for a transformer model to predict KB entities.
In Wikipedia-based experiments, we report values from (Févry et al., 2020) and (Kolitsas et al., 2018) for AIDA-B test set. On MSNBC (MSB), AQUAINT (AQ), and ACE2004 (ACE) test datasets, only (Kolitsas et al., 2018), DBpedia Spotlight (Mendes et al., 2011), KEA (Steinmetz and Sack, 2013), and Babelfy (Moro et al., 2014) report the values and we compare against them.

Hyper-parameters	**Value**
Epochs	4
Batch size	8
Learning rate	$2e^{-5}$
Learning rate decay	linear
Adam β_1	0.9
Adam β_2	0.999
dropout	0.1
Loss Function	Cross-Entropy
Classifier	Softmax

Table 1: Hyper-parameters during fine-tuning.

4.3 CHOLAN Configurations

We configure CHOLAN model applying various candidate generation approaches detailed below.
CHOLAN-Wikidata: we train the model using T-REx dataset and employ $C\_Falcon(m)$ candidate set. The ED model (WikiBERT) is fed with the sentential context but not with entity description as not all Wikidata entities have a corresponding Wikipedia entity.
CHOLAN-Wiki+FC: is trained on CoNLL-AIDA (Hoffart et al., 2011). For CG step, we employ Falcon candidate set $C\_Falcon(m)$. Here, the ED model (WikiBERT) is only fed with the sentential context.
CHOLAN-Wiki+DCA: We train the MD and ED models on CoNLL-AIDA. The CG step involves DCA candidate set $C(m)$. During ED step (WikiBERT), Wikipedia descriptions associated with each entity is fed along with sentential context.
CHOLAN: inherits **CHOLAN-Wiki+FC** but in addition, Wikipedia entity description is induced into the ED model (WikiBERT).

4.4 Metrics and Hyper-parameters

On Wikidata-based experiments, we employ standard metrics of accuracy i.e., precision (P), recall (R), and F-score (F) same as (Mulang et al., 2020). For Wikipedia-based datasets, we use Micro-F1 score in strong matching setting (Kolitsas et al., 2018). The strong matching needs exactly predicting the gold mention (i.e. target entity mention) boundaries and its corresponding entity annotation in the KB. To compare the recalls of two CG approaches, we report the performance on gold recall. Gold recall is the percentage of entity mentions for which the candidate set contain the ground truth entity (Yao et al., 2019).
We have implemented all our models in PyTorch[3]

[3] https://pytorch.org/

and optimized using Adam (Kingma and Ba, 2015). We used the pre-trained BERT models from the Transformers library (Wolf et al., 2019). We ran all the experiments on a single GeForce GTX 1080 Ti GPU with 11GB size. Table 1 outlines the hyper-parameters used in the fine-tuning on both the datasets. We followed the standard settings suggested by (Devlin et al., 2019). The average run time is 9.31 hours/epoch for CHOLAN and without description, it was 7.23 hours/epoch.

5 Results

We study the following research question:*what is the impact of each sub-task (aka component) on the overall outcome of the transformer-based entity linking approach?* We further investigate a sub-research question: how do the external context and the candidate generation step impact the overall performance of CHOLAN? Our every experiment systematically studies the research questions in different settings.

Model	P	R	F
Delpeuch 2020	40.7	**82.9**	57.9
Mulang et al. 2020	71.4	71.2	71.3
CHOLAN-Wikidata	**75**	76	**75.4**

Table 2: Comparison on T-REx test set for Wikidata EL. Best values in bold.

5.1 Results on Wikidata dataset

Table 2 summarises CHOLAN performance on T-REx dataset. CHOLAN-Wikidata configuration outperforms the baselines. We dig deeper into our reported values. We observe that for MD task, our F-score is 94.3 (compared to 77 F-score of Arjun (Mulang et al., 2020)). However, the gold recall for CG step is 81.2. We generate the entity candidates using an information retrieval approach (BM25† algorithm) to get the top 30 candidates based on the confidence score. The Wikidata KG is challenging, and many labels share the same name. It contributes to a large loss in the F-score for the CG step. For instance, the entity mention "National Highway" matches exactly with four Wikidata ID labels while 2,055 other entities contain the full mention in their labels. Please note that we did not perform retraining of (Kolitsas et al., 2018) (SOTA on Wikipedia EL) on the T-REx dataset since we determined that the model is tightly coupled and relies on pre-computed Wikipedia candidate list from (Ganea and Hofmann, 2017).

5.1.1 Ablation Study on Wikidata

We study the impact of local context on the performance of CHOLAN. Therefore, we exclude the sentence as input in the ED step at training and testing time. Hence, the inputs to the ED model are only entity mention and the entity candidates gained from the CG step. We observe that the performance drops when the local sentential context is not fed (cf. Table 3). It justifies our choice to feed the model by the sentence during the ED task.

Model	P	R	F
CHOLAN-Wikidata	**75**	**76**	**75.4**
CHOLAN-Wikidata (WLC†)	72	73.5	72.7

Table 3: The ablation study on T-REx test set for Wikidata EL. Best values in bold. WLC† denotes model without local context. When the local sentential context is excluded from ED, the performance drops.

5.2 Results on Wikipedia datasets

Table 4 reports the performance of CHOLAN's configurations on AIDA-B test set. The first configuration is "CHOLAN-Wiki+ FC" in which MD and ED models are trained using CoNLL-AIDA. We notice a clear jump in the performance. We then replaced the Falcon candidate list $C\_Falcon(m)$ with DCA candidates $C(m)$ resulting into "CHOLAN-Wiki+ DCA". In DCA candidates, the description of entities is attached. The performance is increased when an additional background knowledge as an entity description is fed. Our next configuration is CHOLAN where we attached Wikipedia entity descriptions in Falcon candidate list $C\_Falcon(m)$ (as a modification of "CHOLAN-Wiki+ FC"). This setting outperforms all the existing baselines and previous CHOLAN configurations. Our experiments illustrate the impact of CG step and background knowledge on end-to-end EL performance. The improvement of CHOLAN continues to the other three test datasets where the jump is significantly higher compared to the baselines (cf. Table 5). Reported values in Table 5 also approves transferability of CHOLAN when we apply cross-domain experiments.

5.2.1 Ablation Study on Wikipedia

We conducted three ablation studies to understand the behaviour of CHOLAN's configurations over Wikipedia datasets. The first study

Model	Micro F1
Hoffart et al. 2011	72.8
Mendes et al. 2011	57.8
Steinmetz and Sack 2013	42.3
Moro et al. 2014	48.5
Piccinno and Ferragina 2014	73
Kolitsas et al. 2018	82.4
Peters et al. 2019	73.7
Broscheit 2019	79.3
Févry et al. 2020	76.7
CHOLAN-Wiki+ FC	75.1
CHOLAN-Wiki+ DCA	77.5
CHOLAN	**83.1**

Table 4: Comparison on *AIDA-B*. Best value in bold and previous SOTA value is underlined.

Model	MSB	AQ	ACE
Mendes et al. 2011	40.6	45.2	60.5
Steinmetz and Sack 2013	30.9	35.9	40.3
Moro et al. 2014	39.7	35.8	17.8
Kolitsas et al. 2018	72.4	40.4	68.3
CHOLAN-Wiki+ FC	77.8	70	85.7
CHOLAN-Wiki+ DCA	78.3	75.9	71.3
CHOLAN	**83.4**	**76.8**	**86.8**

Table 5: The micro F1 scores are listed from the comparative study over three datasets (out of domain). The model is trained over CoNLL-AIDA dataset. Best value in bold and previous SOTA value is underlined.

is to calculate the Gold recall values for various datasets. CHOLAN uses the candidates from $C\_Falcon(m)$ candidate set for each entity mention. While generating the candidate set from local KG of (Sakor et al., 2019) we observe a drop in the Gold recall as reported in Table 6. CG plays a crucial role in trading off precision and recall. We conclude that more robust CG approaches likely impact overall performance. The second ablation study is about to calculate the performance of our configurations for ED step, i.e., running WikiBERT in isolation. Here, we assume that all entities are truly recognised; thus, our focus of the study is the ED model. We report the impact of various candidate generation approaches on the ED model in Table 7. The significant jump in the performance from "CHOLAN-Wiki+FC Vs CHOLAN" contributes to the additional background knowledge provided in CHOLAN as entity candidate descriptions. The third ablation study tests the impact of sentential context fed into two configurations on a Wikipedia dataset. Table 8 reports the achieved performance after excluding sentence as the additional context. Obviously, the performance decreases. The model shows similar behaviour on T-REx in Table 3. These observations confirm our hypothesis as the ED model is enhanced using additional contexts.

Model	AIDA-B	MSB	AQ	ACE
Falcon Candidates	94	93.8	85.3	97.3
DCA Candidates	98.3	98.5	94.2	90.6

Table 6: Gold Recall for Candidate Generation techniques over Wikipedia test datasets.

Model	Micro F1
Kolitsas et al. 2018	83.8
CHOLAN-Wiki+ FC	78.4
CHOLAN-Wiki+ DCA	79.1
CHOLAN	**85.7**

Table 7: Comparison on *AIDA-B* for ED. Best score in bold and previous SOTA value is underlined.

Model	Micro F1
CHOLAN-Wiki+ DCA	77.5
CHOLAN-Wiki+ DCA (WLC†)	71.2
CHOLAN	**83.1**
CHOLAN (WLC†)	79.6

Table 8: Ablation study on *AIDA-B*. We observe that when local sentential context is removed from ED step, the performance drops. Best values in bold. WLC† denotes model without local context.

6 Conclusions

In the last two years, the NLP research community has extensively tried transformer-based models for the EL task. However, the performance remained lower than Kolitsas et al. (2018). This paper combines the traditional software engineering principle of modular architecture with the context-induced transformers to effectively solve the EL task. Our reason to deviate from an end-to-end architecture was to provide full flexibility to our system in terms of candidate generation list, underlying KG, and induction of the context at the ED step. We attribute CHOLAN's outperformance to the following reasons: 1) the modular architecture, which brings flexibility and interoperability as CHOLAN can treat each task independently. Kolitsas et al. (2018) reports that shifting towards joint modelling of MD and ED tasks helps mitigate error propagation from MD to ED. However, the performance of BERT$_{BASE}$ for the MD task is significantly high (92.3 on AIDA-B and 94.3 F1-

score on T-REX calculated by us) remarkably reducing the errors in MD. CHOLAN leverages this capability in the MD subtask, placing more focus on CG and ED tasks. 2) The flexibility in architecture further permits us to induce sentence and entity descriptions as additional contexts. Furthermore, using candidate list in plug and play manner has resulted in a significant increase in the performance. In earlier transformer approaches, the implementation is monolithic and context is not utilised. There are scopes for improvement in our approach. Wu et al. (2019a) introduces a novel CG method that retrieves candidates in a dense space defined by a bi-encoder and can be used as alternate CG approach. We aim for scaling CHOLAN to multilingual entity linking as a viable next step.

References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. 2007. DBpedia: A Nucleus for a Web of Open Data. In *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007.*

Kurt D. Bollacker, Robert P. Cook, and Patrick Tufts. 2007. Freebase: A Shared Database of Structured General Human Knowledge. In *AAAI 2007*.

Samuel Broscheit. 2019. Investigating entity knowledge in bert with simple neural end-to-end entity linking. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 677–685.

Yixin Cao, Lei Hou, Juanzi Li, and Zhiyuan Liu. 2018. Neural collective entity linking. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 675–686.

Yixin Cao, Lifu Huang, Heng Ji, Xu Chen, and Juanzi Li. 2017. Bridge text and knowledge by learning multi-prototype entity mention embedding. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1623–1633.

Diego Ceccarelli, Claudio Lucchese, Salvatore Orlando, Raffaele Perego, and Salvatore Trani. 2013. Dexter: an open source framework for entity linking. In *Proceedings of the sixth international workshop on Exploiting semantic annotations in information retrieval*, pages 17–20.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167.

Marco Cornolti, Paolo Ferragina, Massimiliano Ciaramita, Stefan Rüd, and Hinrich Schütze. 2016. A piggyback system for joint entity mention detection and linking in web queries. In *Proceedings of the 25th International Conference on World Wide Web*, pages 567–578.

Antonin Delpeuch. 2020. Opentapioca: Lightweight entity linking for wikidata. *The 1st Wikidata Workshop co-located with International Semantic Web Conference 2020 (to appear)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Greg Durrett and Dan Klein. 2014. A joint model for entity analysis: Coreference, typing, and linking. *Transactions of the association for computational linguistics*, 2:477–490.

Hady ElSahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon S. Hare, Frédérique Laforest, and Elena Simperl. 2018. T-rex: A large scale alignment of natural language with knowledge base triples. In *LREC*.

Zheng Fang, Yanan Cao, Qian Li, Dongjie Zhang, Zhenyu Zhang, and Yanbing Liu. 2019. Joint entity linking with deep reinforcement learning. In *The World Wide Web Conference*, pages 438–447.

Thibault Févry, Nicholas FitzGerald, Livio Baldini Soares, and Tom Kwiatkowski. 2020. Empirical evaluation of pretraining strategies for supervised entity linking. In *Automated Knowledge Base Construction*.

Octavian-Eugen Ganea and Thomas Hofmann. 2017. Deep joint entity disambiguation with local neural attention. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2619–2629.

Ralph Grishman and Beth M Sundheim. 1996. Message understanding conference-6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*.

Zhaochen Guo and Denilson Barbosa. 2018. Robust named entity disambiguation with random walks. *Semantic Web*, 9(4):459–479.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 782–792.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Nikolaos Kolitsas, Octavian-Eugen Ganea, and Thomas Hofmann. 2018. End-to-end neural entity linking. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 519–529.

Lajanugen Logeswaran, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Jacob Devlin, and Honglak Lee. 2019. Zero-shot entity linking by reading entity descriptions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3449–3460.

Pablo N. Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. 2011. DBpedia spotlight: shedding light on the web of documents. In *I-SEMANTICS*.

Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014. Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, 2:231–244.

Isaiah Onando Mulang, Kuldeep Singh, Akhilesh Vyas, Saeedeh Shekarpour, Ahmad Sakor, Maria Esther Vidal, Soren Auer, and Jens Lehmann. 2020. Encoding knowledge graph entity aliases in an attentive neural networks for wikidata entity linking. *Web Information System and Engineering*.

Dat Ba Nguyen, Martin Theobald, and Gerhard Weikum. 2016. J-nerd: joint named entity recognition and disambiguation with rich linguistic features. *Transactions of the Association for Computational Linguistics*, 4:215–229.

Matthew E Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith. 2019. Knowledge enhanced contextual word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 43–54.

Francesco Piccinno and Paolo Ferragina. 2014. From tagme to wat: a new entity annotator. In *Proceedings of the first international workshop on Entity recognition & disambiguation*, pages 55–62.

Tim Rocktäschel, Torsten Huber, Michael Weidlich, and Ulf Leser. 2013. Wbi-ner: The impact of domain-specific features on the performance of identifying and classifying mentions of drugs. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 356–363.

Ahmad Sakor, Isaiah Onando Mulang, Kuldeep Singh, Saeedeh Shekarpour, Maria-Esther Vidal, Jens Lehmann, and Sören Auer. 2019. Old is gold: Linguistic driven approach for entity and relation linking of short text. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019,Volume 1 (Long and Short Papers)*, pages 2336–2346. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *CoRR*, cs.CL/0306050.

Ozge Sevgili, Artem Shelmanov, Mikhail Arkhipov, Alexander Panchenko, and Chris Biemann. 2020. Neural entity linking: A survey of models based on deep learning.

Valentin I Spitkovsky and Angel X Chang. 2012. A cross-lingual dictionary for english wikipedia concepts.

Nadine Steinmetz and Harald Sack. 2013. Semantic multimedia information retrieval based on contextual descriptions. In *Extended Semantic Web Conference*, pages 382–396. Springer.

Denny Vrandecic. 2012. Wikidata: a new platform for collaborative data collection. In *Proceedings of the 21st World Wide Web Conference, WWW 2012, Lyon, France, April 16-20, 2012 (Companion Volume)*, pages 1063–1064.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771v5.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2019a. Zero-shot entity linking with dense entity retrieval. *arXiv preprint arXiv:1911.03814*.

Shanchan Wu, Kai Fan, and Qiong Zhang. 2019b. Improving distantly supervised relation extraction with neural noise converter and conditional optimal selector. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019*, pages 7273–7280. AAAI Press.

Vikas Yadav and Steven Bethard. 2018. A survey on recent advances in named entity recognition from deep learning models. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2145–2158.

Ikuya Yamada and Hiroyuki Shindo. 2019. Pretraining of deep contextualized embeddings of words and entities for named entity disambiguation. *arXiv preprint arXiv:1909.00426*.

Xiyuan Yang, Xiaotao Gu, Sheng Lin, Siliang Tang, Yueting Zhuang, Fei Wu, Zhigang Chen, Guoping Hu, and Xiang Ren. 2019. Learning dynamic context augmentation for global entity linking. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 271–281.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Kg-bert: Bert for knowledge graph completion. *arXiv preprint arXiv:1909.03193*.

Stefan Zwicklbauer, Christin Seifert, and Michael Granitzer. 2016. Robust and collective entity disambiguation through semantic embeddings. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 425–434.