Falcon 2.0: An Entity and Relation Linking Tool over Wikidata

Ahmad Sakor
ahmad.sakor@tib.eu
L3S Research Center and TIB, University of Hannover
Hannover, Germany

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

Anery Patel
anery.patel@tib.eu
TIB, University of Hannover
Hannover, Germany

Maria-Esther Vidal
maria.vidal@tib.eu
L3S Research Center and TIB, University of Hannover
Hannover, Germany

ABSTRACT

The Natural Language Processing (NLP) community has significantly contributed to the solutions for entity and relation recognition from a natural language text, and possibly linking them to proper matches in Knowledge Graphs (KGs). Considering Wikidata as the background KG, there are still limited tools to link knowledge within the text to Wikidata. In this paper, we present Falcon 2.0, the first joint entity and relation linking tool over Wikidata. It receives a short natural language text in the English language and outputs a ranked list of entities and relations annotated with the proper candidates in Wikidata. The candidates are represented by their Internationalized Resource Identifier (IRI) in Wikidata. Falcon 2.0 resorts to the English language model for the recognition task (e.g., N-Gram tiling and N-Gram splitting), and then an optimization approach for the linking task. We have empirically studied the performance of Falcon 2.0 on Wikidata and concluded that it outperforms all the existing baselines. Falcon 2.0 is open source and can be reused by the community; all the required instructions of Falcon 2.0 are well-documented at our GitHub repository[1]. We also demonstrate an online API, which can be run without any technical expertise. Falcon 2.0 and its background knowledge bases are available as resources at https://labs.tib.eu/falcon/falcon2/.

CCS CONCEPTS

• **Information systems → Resource Description Framework (RDF)**; **Information extraction**.

KEYWORDS

NLP, Entity Linking, Relation Linking, Background Knowledge, English morphology, DBpedia, and Wikidata

ACM Reference Format:
Ahmad Sakor, Kuldeep Singh, Anery Patel, and Maria-Esther Vidal. 2020. Falcon 2.0: An Entity and Relation Linking Tool over Wikidata. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3340531.3412777

[1]https://github.com/SDM-TIB/falcon2.0

This work is licensed under a Creative Commons Attribution International 4.0 License.
CIKM '20, October 19–23, 2020, Virtual Event, Ireland
© 2020 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-6859-9/20/10.
https://doi.org/10.1145/3340531.3412777

1 INTRODUCTION

Entity Linking (EL)- also known as Named Entity Disambiguation (NED)- is a well-studied research domain for aligning unstructured text to its structured mentions in various knowledge repositories (e.g., Wikipedia, DBpedia [1], Freebase [4] or Wikidata [28]). Entity linking comprises two sub-tasks. The first task is Named Entity Recognition (NER), in which an approach aims to identify entity labels (or surface forms) in an input sentence. Entity disambiguation is the second sub-task of linking entity surface forms to semi-structured knowledge repositories. With the growing popularity of publicly available knowledge graphs (KGs), researchers have developed several approaches and tools for EL task over KGs. Some of these approaches implicitly perform NER and directly provide mentions of entity surface forms in the sentences to the KG (often referred to as end-to-end EL approaches) [7]. Other attempts (e.g., Yamanda et al. [30], DCA [32]) consider recognized surface forms of the entities as additional inputs besides the input sentence to perform entity linking. Irrespective of the input format and underlying technologies, the majority of the existing attempts [22] in the EL research are confined to well-structured KGs such as DBpedia or Freebase[2]. These KGs rely on a well-defined process to extract information directly from Wikipedia infoboxes. They do not provide direct access to the users to add/delete the entities or alter the KG facts. Wikidata, on the other hand, also allows users to edit Wikidata pages directly, add newer entities, and define new relations between the objects. Wikidata is hugely popular as a crowdsourced collection of knowledge. Since its launch in 2012, over 1 billion edits have been made by the users across the world[3].

Motivation, Approach, and Contributions. We motivate our work by the fact that despite the vast popularity of Wikidata, there are limited attempts to target entity and relation linking over Wikidata. For instance, there are over 20 entity linking tools/APIs for DBpedia [22, 26], which are available as APIs. To the best of our knowledge, there exists only one open-source API for Wikidata entity linking (i.e., OpenTapioca [7]). Furthermore, there is no tool over Wikidata for relation linking, i.e., linking predicate surface forms to their corresponding Wikidata mentions. In this paper, we focus on providing Falcon 2.0, a reusable resource API for joint

[2]it is now depreciated and no further updates are possible
[3]https://www.wikidata.org/wiki/Wikidata:Statistics

entity and relation linking over Wikidata. In our previous work, we proposed Falcon [23], a rule-based approach yet effective for entity and relation linking on short text (questions in this case) over DBpedia. In general, the Falcon approach has two novel concepts: 1) a linguistic-based approach that relies on several English morphology principles such as tokenization, and N-gram tiling; 2) a local knowledge base which serves as a source of background knowledge (BK). This knowledge base is a collection of entities from DBpedia. We **resort** to the Falcon approach for developing `Falcon 2.0`. Our aim here is to study whether or not the Falcon approach is agnostic to underlying KG; hence, we do not claim novelty in the underlying *linguistic-based approach* for `Falcon 2.0`. Further, we investigate the concerns related to robustness, emerging failures, and bottlenecks. We introduce `Falcon 2.0` based on the methodology employed in the first version. Our tool is the *first* joint entity and relation linking tool for Wikidata. Our novel contributions briefly lie in two aspects:

(1) **`Falcon 2.0`**: The first resource for joint entity and relation linking over Wikidata. `Falcon 2.0` relies on fundamental principles of English morphology (tokenization and compounding) and links entity and relation surface forms in a short sentence to its Wikidata mentions. `Falcon 2.0` is available as an online API and can be accessed at https://labs.tib.eu/falcon/falcon2/. Falcon 2.0 is also able to recognize entities in keywords such as Barack Obama, where there is no relation. We empirically evaluate `Falcon 2.0` on three datasets tailored for Wikidata. According to the observed results, `Falcon 2.0` significantly outperforms all the existing baselines. For the ease of use, we integrate the Falcon API[4] into Falcon 2.0. This option is available in case Wikipedia contains an equivalence entity (Wikidata is a superset of DBpedia) The `Falcon 2.0` API already has over half a million hits from February 2020 to the time of paper acceptance, which shows its gaining usability (excluding self-access of the API while performing the evaluation).

(2) **`Falcon 2.0 Background KG`**: We created a new background KG of `Falcon 2.0` with the Wikidata. We extracted 48,042,867 Wikidata entities from its public dump and aligned these entities with the aliases present in Wikidata. For example, Barack Obama is a Wikidata entity Wiki:Q76[5]. We created a mapping between the label (Barack Obama) of Wiki:Q76 with its aliases such as President Obama, Barack Hussein Obama, and Barry Obama and stored it in the background knowledge base. We implemented a similar alignment for 15,645 properties/relations of Wikidata. The background knowledge base is an indexed graph and can be queried. The resource is also present at a persistent URI for further reuse[6].

The rest of this paper is organized as follows: Section 2 reviews the state-of-the-art, and the following Section 3 describes our two resources and approach to build `Falcon 2.0`. Section 4 presents experiments to evaluate the performance of `Falcon 2.0`. Section 5 presents the importance and impact of this work for the research community. The availability and sustainability of resources is explained in Section 6, and its maintenance related discussion is presented in Section 7. We close with the conclusion in Section 8.

2 RELATED WORK

Several surveys provide a detailed overview of the advancements of the techniques employed in entity linking over KGs [2, 24]. Various reading lists [16], online forums[7] and Github repositories[8] track the progress in the domain of entity linking. Initial attempts in EL considered Wikipedia as an underlying knowledge source. The research field has matured and the SOTA nearly matches human-level performance [20]. With the advent of publicly available KGs such as DBpedia, Yago, and Freebase, the focus has shifted towards developing EL over knowledge graphs. The developments in Deep Learning have introduced a range of models that carry out both NER and NED as a single end-to-end step [11, 17]. NCEL [5] learns both local and global features from Wikipedia articles, hyperlinks, and entity links to derive joint embeddings of words and entities. These embeddings are used to train a deep Graph Convolutional Network (GCN) that integrates all the features through a Multi-layer Perceptron. The output is passed through a Sub-Graph Convolution Network, which finally resorts to a fully connected decoder. The decoder maps the output states to linked entities. The BI-LSTM+CRF model [15] formulates entity linking as a sequence learning task in which the entity mentions are a sequence whose length equals the series of the output entities. Albeit precise, deep learning approaches demand *high-quality* training annotations, which are not extensively available for Wikidata entity linking [6, 19].

There is concrete evidence in the literature that the machine learning-based models trained over generic datasets such as WikiDisamb30 [10], and CoNLL (YAGO) [14] do not perform well when applied to short texts. Singh et al. [26] evaluated more than 20 entity linking tools over DBpedia for short text (e.g., questions) and concluded that issues like capitalization of surface forms, implicit entities, and multi-word entities affect the performance of EL tools in a short input text. Sakor et al. [23] addresses specific challenges of short texts by applying a rule-based approach for EL over DBpedia. In addition to linking entities to DBpedia, Sakor et al. also provides DBpedia IRIs of the relations in a short text. EARL [3] is another tool that proposes a traveling salesman algorithm-based approach for joint entity and relation linking over DBpedia. To the best of our knowledge, EARL and Falcon are the only available tools that provide both entity and relation linking.

Entity linking over Wikidata is a relatively new domain. Cetoli et al. [6] propose a neural network-based approach for linking entities to Wikidata. The authors also align an existing Wikipedia corpus-based dataset to Wikidata. However, this work only targets entity disambiguation and assumes that the entities are already recognized in the sentences. Arjun [19] is the latest work for Wikidata entity linking. It uses an attention-based neural network for linking Wikidata entity labels. OpenTapioca [7] is another attempt that performs end-to-end entity linking over Wikidata; it is the closest to our work even though OpenTapioca does not provide Wikidata

[4] https://labs.tib.eu/falcon/
[5] https://www.wikidata.org/wiki/Q76
[6] https://doi.org/10.6084/m9.figshare.11362883
[7] http://nlpprogress.com/english/entity_linking.html
[8] https://github.com/sebastianruder/NLP-progress/blob/master/english/entity_linking.md

Ids of relations in a sentence. OpenTapioca is also available as an API and is utilized as our baseline. S-Mart [33] is a tree-based structured learning framework based on multiple additive regression trees for linking entities in a tweet. The model was later adapted for linking entities in the questions. VCG [27] is another attempt which is a unifying network that models contexts of variable granularity to extract features for an end to end entity linking. However, Falcon 2.0 is the *first* tool for joint entity and relation linking over Wikidata.

3 FALCON 2.0- A RESOURCE

In this section, we describe Falcon 2.0 in detail. First the architecture of Falcon 2.0 is depicted. Next, we discuss the BK used to match the surface forms in the text to the resource in a specific KG. In the paper's scope, we define "short text" as grammatically correct questions (up to 15 words).

3.1 Architecture

The Falcon 2.0 architecture is depicted in Figure 1. Falcon 2.0 receives short input texts and outputs a set of entities and relations extracted from the text; each entity and relation in the output is associated with a unique Internationalized Resource Identifier (IRI) in Wikidata. Falcon 2.0 resorts to BK and a catalog of rules for performing entity and relation linking. The BK combines Wikidata labels and their corresponding aliases. Additionally, it comprises alignments between nouns and entities in Wikidata. Alignments are stored in a text search engine, while the knowledge source is maintained in an RDF triple store accessible via a SPARQL endpoint. The rules that represent the English morphology are in a catalog; a forward chaining inference process is performed on top of the catalog during the extraction and linking tasks. Falcon 2.0 also comprises several modules that identify and link entities and relations to the Wikidata. These modules implement POS Tagging, Tokenization & Compounding, N-Gram Tiling, Candidate List Generation, Matching & Ranking, Query Classifier, and N-Gram Splitting and are reused from the implementation of Falcon.

3.2 Background Knowledge

Wikidata contains over 52 million entities and 3.9 billion facts (in the form of subject-predicate-object triples). Since Falcon 2.0 background knowledge only depends on labels, a significant portion of this extensive information is not useful for our approach. Hence, we only extract all the entity and relation labels to create a local background KG, A.K.A "alias background knowledge base.". For example, the entity United States of America[9] in Wikidata has the natural language label 'United States of America' and several other aliases (or known_as labels) of United States of America such as "the United States of America, America, U.S.A., the U.S., United States, etc.". We extended our background KG with this information from Wikidata. Similarly, for relation's labels, the background KG is enriched with known_as labels to provide synonyms and derived word forms. For example, the relation spouse [10] in Wikidata has the label spouse and the other known as labels are husband, wife, married to, wedded to, partner, etc. This variety of synonyms for each relation empowers Falcon 2.0 to match the surface form in the text to a relation in Wikidata. Figure 2 illustrates the process of building background knowledge.

[9]https://www.wikidata.org/wiki/Q30
[10]https://www.wikidata.org/wiki/Property:P26

3.3 Catalog of Rules

Falcon 2.0 is a rule-based approach. A catalog of rules is predefined to extract entities and relations from the text. The rules are based on the English morphological principles and borrowed from Sakor et al. [23]. For example, Falcon 2.0 excludes all verbs from the entities candidates list based on the rule verbs are not entities. For example, the N-Gram tiling module in the Falcon 2.0 architecture resorts to the rule: entities with only stopwords between them are one entity. Another example of such rule When -> date, Where -> place solves the ambiguity of matching the correct relation in case the short text is a question by looking at the questions headword. For example, give the two questions When did Princess Diana die? and Where did Princess Diana die?, the relation died can be the death place or the death year. The question headword (When/Where) is the only insight to solve the ambiguity here. When the question word is where, Falcon 2.0 matches only relations that have a place as a range of the relation.

3.4 Recognition

Extraction phase in Falcon 2.0 consists of three modules. POS tagging, tokenization & compounding, and N-Gram tiling. The input of this phase is a natural language text. The output of the phase is the list of surface forms related to entities or relations.

Part-of-speech (POS) Tagging receives a natural language text as an input. It tags each word in the text with its related tag, e.g., noun, verb, and adverb. This module differentiates between nouns and verbs to enable the application of the morphological rules from the catalog. The output of the module is a list of the pairs of (word, tag).

Tokenization & Compounding builds the tokens list by removing the stopwords from the input and splitting verbs from nouns. For example, if the input is What is the operating income for Qantas, the output of this module is a list of three tokens [operating, income, Qantas].

N-Gram Tilling module combines tokens with only stopwords between them relying on one of the rules from a catalog of rules. For example, if we consider the previous module's output as an input for the n-gram tilling module, operating and income tokens will be combined in one token. The output of the module is a list of two tokens [operating income, Qantas].

3.5 Linking

This phase consists of four modules: candidate list generation, matching & ranking, relevant rule selection, and n-gram splitting.

Candidate List Generation receives the output of the recognition phase. The module queries the text search engine for each token. Then, tokens will have an associated candidate list of resources. For example, the retrieved candidate list of the token operating income is [(P3362, operating income), (P2139, income), (P3362, operating profit)]; where the first element is the Wikidata predicate identifier and the second is the list of labels associated with the

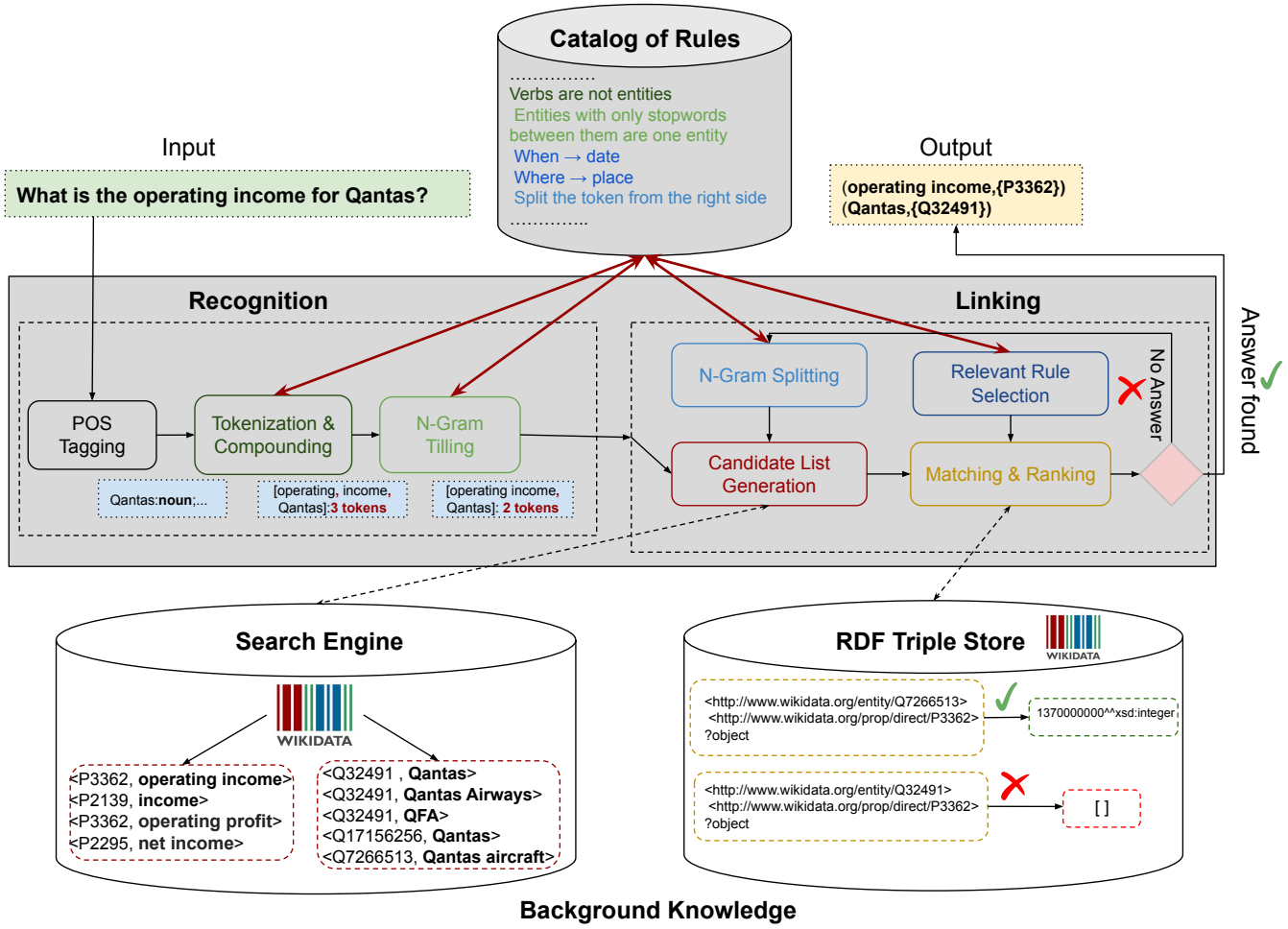

Figure 1: The `Falcon 2.0` Architecture. The boxes highlighted in Grey are reused from Falcon [23]. Grey boxes contain a linguistic pipeline for recognizing and linking entity and relation surface forms. The boxes in White are our addition to the Falcon pipeline to build a resource for the Wikidata entity and relation linking. The white boxes constitute what we refer to as BK specific to Wikidata. The text search engine contains the alignment of Wikidata entity/relation labels along with the entity and relation aliases. It is used for generating potential candidates for entity and relation linking. RDF triple store is a local copy of Wikidata triples containing all entities and predicates.

predicates which match the query "operating income."

Matching & Ranking ranks the candidate list received from the candidate list generation module and matches candidates' entities and relations. Since, in any KG, the facts are represented as triples, the matching and ranking module creates triples consisting of the entities and relationships from the candidates' list. Then, for each pair of entity and relation, the module checks if the triple exists in the RDF triple store (Wikidata). The checking is done by executing a simple ASK query over the RDF triple store. For each triple, the module increases the rank of the involved relations and entities. The output of the module is the ranked list of the candidates.

Relevant Rule Selection interacts with the matching & ranking module by suggesting increasing the ranks of some candidates relying on the catalog of rules. One of the suggestions is considering the question headword to clear the ambiguity between two relations based on the range of relationships in the KG.

N-Gram Splitting module is used if none of the triples tested in the matching & ranking modules exists in the triple store, i.e., the compounding the approach did in the tokenization & compounding module led to combining two separated entities. The module splits the tokens from the right side and passes the tokens again to the candidate list generation module. Splitting the tokens from the right side resorts to one of the fundamentals of the English morphology; the compound words in English have their headword always towards the right side [29].

Text Search Engine stores all the alignments of the labels. A simple querying technique [12] is used as the text search engine over background knowledge. It receives a token as an input and then returns all the related resources with labels similar to the received token.

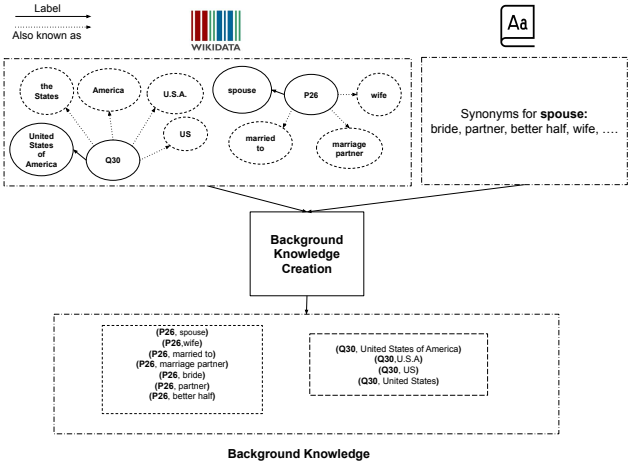

Figure 2: `Falcon 2.0` Background Knowledge is built by converting labels of entities and relations in Wikidata into pairs of alignments. It is a part of search engine (cf. Figure 1).

Figure 3: `Falcon 2.0` API Web Interface.

RDF Triple store is a local copy of the Wikidata endpoint. It consists of all the RDF triples of Wikidata labeled with the English language. An RDF triple store is used to check the existence of the triples passed from the **Matching & Ranking** module. The RDF triple store keeps around 3.9 billion triples.

4 EXPERIMENTAL STUDY

We study three research questions: RQ1) What is the performance of `Falcon 2.0` for entity linking over Wikidata? RQ2) What is the impact of Wikidata's specific background knowledge on the performance of a linguistic approach? RQ3) What is the performance of `Falcon 2.0` for relation linking over Wikidata?

Metrics. We report the performance using the standard metrics of Precision, Recall, and F-measure. Precision is the fraction of relevant resources among the retrieved resources. Recall is the fraction of relevant resources that have been retrieved over the total amount of relevant resources. F-Measure or F-Score is the harmonic mean of precision and recall.

Datasets. We rely on three different question answering datasets namely SimpleQuestion dataset for Wikidata [8], WebQSP-WD [27] and LC-QuAD 2.0 [9]. The SimpleQuestion dataset contains 5,622 test questions which are answerable using Wikidata as underlying KG. WebQSP-WD contains 1639 test questions, and LC-QUAD 2.0 contains 6046 test questions. SimpleQuestion and LC-QuaD 2.0 provide the annotated gold standard for entity and relations, whereas WebQSP-WD only provides annotated gold standard for entities. Hence, we evaluated entity linking performance on three datasets and relation linking performance on two datasets. Also, SimpleQuestion and WebQSP-WD contain questions with a single entity and relation, whereas LC-QuAD 2.0 contains mostly complex questions (i.e., more than one entity and relation).

Baselines. **OpenTapioca** [7]: is available as a web API; it provides Wikidata URIs for entities. We run OpenTapioca API on all the three datasets.

Variable Context Granularity model (VCG) [27]: is a unifying network that models contexts of variable granularity to extract features for mention detection and entity disambiguation. We were unable to reproduce VCG using the publicly available source code. Hence, we only report its performance on WebQSP-WD from the original paper [27] as we are unable to run the model on the other two datasets for entity linking. For the completion of the approach, we also report the other two baselines provided by the authors, namely **Heuristic Baseline** and **Simplified VCG**.

S-Mart [33]: was initially proposed to link entities in the tweets and later adapted for question answering. The system is not open source, and we adapt its result from [27] for WebQSP-WD dataset.

No Baseline for Relation Linking: To the best of our knowledge, there is no baseline for relation linking on Wikidata. One argument could be to run the existing DBpedia based relation linking tool on Wikidata and compare it with our performance. We contest this solely because Wikidata is extremely noisy. For example, in "What is the longest National Highway in the world?" the entity surface form "National Highway" matches four(4) different entities in Wikidata that share the same entity label (i.e., "National Highway"). In comparison, 2,055 other entities contain the full mention in their labels for the surface form "National Highway". However, in DBpedia, there exists only one unique label for "National Highway". Hence, any entity linking tool or relation linking tool tailored for DBpedia will face issues on Wikidata (cf. table 3). Therefore, instead of reporting the bias and under-performance, we did not evaluate their performance for a fair comparison. Hence, we report `Falcon 2.0` relation linking performance only to establish new baselines on two datasets: SimpleQuestion and LC-QuAD 2.0.

Experimental Details. `Falcon 2.0` is extremely lightweight from an implementation point of view. A laptop machine, with eight cores and 16GB RAM running Ubuntu 18.04 is used for implementing and evaluating `Falcon 2.0`. We deployed its web API on a server with 723GB RAM, 96 cores (Intel(R) Xeon(R) Platinum 8160CPU with 2.10GHz) running Ubuntu 18.04. This publicly available API is used to calculate the standard evaluation metrics, namely Precision, Recall, and F-score.

Table 1: Entity linking evaluation results on LC-QuAD 2.0 & SimpleQuestion datasets. Best values are in bold.

Approach	Dataset	P	R	F
OpenTapioca [7]	LC-QuAD 2.0	0.29	0.42	0.35
Falcon 2.0	LC-QuAD 2.0	**0.50**	**0.56**	**0.53**
OpenTapioca [7]	SimpleQuestion	0.01	0.02	0.01
Falcon 2.0	SimpleQuestion	**0.56**	**0.64**	**0.60**
OpenTapioca [7]	SimpleQuestion Uppercase Entities	0.16	0.28	0.20
Falcon 2.0	SimpleQuestion Uppercase Entities	**0.66**	**0.75**	**0.70**

Table 2: Entity linking evaluation results on the WEBQSP test dataset. Best values are in bold.

Approach	P	R	F
S-MART [33]	0.66	0.77	0.72
Heuristic baseline [27]	0.30	0.61	0.40
Simplified VCG [27]	**0.84**	0.62	0.71
VCG [27]	0.83	0.65	0.73
OpenTapioca [7]	0.01	0.02	0.02
Falcon 2.0	0.80	**0.84**	**0.82**

4.1 Experimental Results

Experimental Results 1. In the first experiment described in Table 1, we compare entity linking performance of Falcon 2.0 on SimpleQuestion and LC-QuAD 2.0 datasets. We first evaluate the performance on the SimpleQuestion dataset. Surprisingly, we observe that for the OpenTapioca baseline, the values are approximately 0.0 for Precision, Recall, and F-score. We analyzed the source of errors and found that out of 5,622 questions, only 246 have entity labels in uppercase letters. Opentapioca fails to recognize and link entity mentions written in lowercase letters. Case sensitivity is a common issue for entity linking tools over short text, as reported by Singh et al. [25, 26] in a detailed analysis. From the remaining 246 questions, only 70 are answered correctly by OpenTapioca. Given that OpenTapioca finds limitation in linking lowercase entity surface forms. We evaluated Falcon 2.0 and OpenTapioca on the 246 questions of SimpleQuestion to provide a fair evaluation for the baseline (reported as SimpleQuestion uppercase entities in table 1). OpenTapioca reports F-score 0.20 on this subset of SimpleQuestion. On the other hand, Falcon 2.0 reports F-score 0.70 on the same dataset (cf. Table 1). For LC-QuAD 2.0, OpenTapioca reports F-score 0.35 against Falcon 2.0 with F-score 0.53 reported in Table 1.

Experimental Results 2. We report performance of Falcon 2.0 on WebQSP-WD dataset in Table 2. Falcon 2.0 clearly outperforms all other baselines with highest F-score value 0.82. OpenTapioca demonstrates a low performance on this dataset as well. Experiment results 1 & 2 answer our first research question (RQ1).

Ablation Study for Entity Linking and Recommendations. For the second research question (RQ2), we evaluate the impact of Wikidata's specific background knowledge on the entity linking performance. We evaluated Falcon on the WebQSP-WD dataset against Falcon 2.0. We linked Falcon predicted DBpedia IRIs with corresponding Wikidata IDs using owl:sameAs. We can see in the Table 3 that Falcon 2.0 significantly outperforms Falcon despite using the same linguistic driven approach. The jump in Falcon 2.0 performance comes from Wikidata's specific local background knowledge, which we created by expanding Wikidata entities and relations with associated aliases. It also validates the novelty of Falcon 2.0 when compared to Falcon for the Wikidata entity linking.

We observe an indifferent phenomenon in our performance for three datasets, and the performance for Falcon 2.0 differs a lot per dataset. For instance, on WebQSP-WD, our F-score is 0.82, whereas, on LC-QuAD 2.0, the F-Score drops to 0.57. The first source of error is the dataset(s) itself. In both the datasets (SimpleQuestion and LC-QuAD 2.0), many questions are grammatically incorrect. To validate our claim more robustly, we asked two native English speakers to check the grammar of 200 random questions on LC-QuAD 2.0. Annotators reported that 42 out of 200 questions are grammatically incorrect. Many questions have erroneous spellings of the entity names. For example, "Who is the country for head of state of Mahmoud Abbas?" and "Tell me about position held of Malcolm Fraser and elected in?" are two grammatically incorrect questions in LC-QuAD 2.0. Similarly, many questions in the SimpleQuestion dataset are also grammatically incorrect. "where was hank cochran birthed" is one such example in the SimpleQuestion dataset. Falcon 2.0 resorts to fundamental principles of the English morphology and finds limitation in recognizing entities in many grammatically incorrect questions.

We also recognize that the performance of Falcon 2.0 on sentences with minimal context is limited. For example, in the question "when did annie open?" from the WebQSP-WD dataset, the sentential context is shallow. Also, more than one instance of "Annie" exists in Wikidata, such as Wiki:Q566892 (correct one) and Wiki:Q181734. Falcon 2.0 wrongly predicts the entity in this case. In another example, "which country is lamb from?", the correct entity is Wiki:Q6481017 with label "lamb" in Wikidata. However, Falcon 2.0 returns Wiki:13553878, which also has a label "lamb". In such cases, additional knowledge graph context shall prove to be useful. Approaches such as [32] introduced a concept of feeding "entity descriptions" as an additional context in an entity linking model over Wikipedia. Suppose the extra context in the form of entity description (1985 English drama film directed by Colin Gregg) for the entity Wiki:13553878 is provided. In that case, a model may correctly predict the correct entity "lamb." Based on our observations, we propose the following recommendations for the community to improve the entity linking task over Wikidata:

- Wikidata has inherited challenges of vandalism and noisy entities due to crowd-authored entities [13]. We expect the research community to come up with more robust short text datasets for the Wikidata entity linking without spelling and grammatical errors.
- Rule-based approaches come with its limitations when the sentential context is minimal. However, such methods are beneficial for the nonavailability of training data. We recommend a two-step process to target questions with minimal sentential context: 1) work towards a clean and large Wikidata dataset for entity linking of short text. This will allow more robust machine learning approaches to evolve 2) use of entity descriptions from knowledge graphs to improve the linking process (same as [32]).

Table 3: Entity Linking Performance of Falcon vs Falcon 2.0 on WEBQSP-WD. Best values are in bold.

Approach	P	R	F
Falcon [23]	0.47	0.45	0.46
Falcon 2.0	**0.80**	**0.84**	**0.82**

Table 4: Relation linking evaluation results on LC-QuAD 2.0 & SimpleQuestion datasets.

Approach	Dataset	P	R	F
Falcon 2.0	LC-QuAD 2.0	**0.44**	**0.37**	**0.40**
Falcon 2.0	SimpleQuestion	**0.35**	**0.44**	**0.39**

Experimental Results 3: In the third experiment (for RQ3), we evaluate the relation linking performance of Falcon 2.0. We are not aware of any other model for relation linking over Wikidata. Table 4 summarizes relation linking performance. With this, we established new baselines over two datasets for relation linking on Wikidata.

Ablation Study for Relation Linking and Recommendations. Falcon reported an F-score of 0.43 on LC-QuAD over DBpedia in [23] whereas Falcon 2.0 reports a comparable relation linking F-score 0.40 on LC-QuAD 2.0 for Wikidata (cf. Table 4). The wrong identification of the entities does affect the relation linking performance, and it is the major source of error in our case for relation linking. Table 5 summarizes a sample case study for relation linking on five LC-QuAD 2.0 questions. We observe that the relations present in the questions are highly uncommon and nonstandard, and it is a peculiar property of Wikidata. Falcon 2.0 finds limitations in linking such relations. We recommend the following:

- Wikidata challenges relation linking approaches by posing a new challenge: user-created nonstandard relations such as in Table 5. A rule-based approach like ours faces a clear limitation in linking such relations. Linking user-created relations in crowd-authored Wikidata is an open question for the research community.

5 IMPACT

In August 2019, Wikidata became the first Wikimedia project that crossed 1 billion edits, and over 20,000 active Wikidata editors[11]. A large subset of the information extraction community has extensively relied on its research around DBpedia and Wikidata targeting different research problems such as KG completion, question answering, entity linking, and data quality assessments [18, 21, 34]. Furthermore, entity and relation linking tasks have been studied well beyond information extraction research, especially NLP and Semantic Web. Despite Wikidata being hugely popular, there are limited resources for reusing and aligning unstructured text to Wikidata mentions. However, when it comes to a short text, the performance of existing baselines are limited. We believe the availability of Falcon 2.0 as a web API along with open source access to its code will provide researchers an easy and reusable way to annotate unstructured text against Wikidata. We also believe that a rule-based approach, such as ours that does not require any training data, is beneficial for low resource languages (considering Wikidata is multilingual [12]).

6 ADOPTION AND REUSABILITY

Falcon 2.0 is open source. The source code is available in our public GitHub: https://github.com/SDM-TIB/Falcon2.0 for reusability and reproducibility. Falcon 2.0 is easily accessible via a simple CURL request or using our web interface. Detailed instructions are provided on our GitHub. It is currently available for the English language. However, there is no assumption in the approach or while building the background knowledge base that restricts its adaptation or extensibility to other languages. The background knowledge of Falcon 2.0 is available for the community and can be easily reused to generate candidates for entity linking [31] or in question answering approaches such as [35]. The background knowledge consists of 48,042,867 alignments of Wikidata entities and 15,645 alignments for Wikidata predicates. MIT License allows for the free distribution and re-usage of Falcon 2.0. We hope the research community and industry practitioners will use Falcon 2.0 resources for various usages such as linking entities and relations to Wikidata, annotating an unstructured text, developing new low language resources, and others.

7 MAINTENANCE AND SUSTAINABILITY

Falcon 2.0 is a publicly available resource offering of the Scientific Data Management(SDM) group at TIB, Hannover[13]. TIB is one of the largest libraries for Science and Technology in the world [14]. It actively promotes open access to scientific artifacts, e.g., research data, scientific literature, non-textual material, and software. Similar to other publicly maintained repositories of SDM, Falcon 2.0 will be preserved and regularly updated to fix bugs and include new features[15]. The Falcon 2.0 API will be sustained on the TIB servers to allow for unrestricted free access.

8 CONCLUSION AND FUTURE WORK

We presented the resource Falcon 2.0, a rule-based entity and relation linking tool able to recognize entities & relations in a short text and link them to the existing knowledge graph, e.g., DBpedia and Wikidata. Although there are various approaches for entity & relation linking to DBpedia, Falcon 2.0 is one of the few tools targeting Wikidata. Thus, given the number of generic and domain-specific facts that compose Wikidata, Falcon 2.0 has the potential to impact researchers and practitioners that resort to NLP tools for transforming semi-structured data into structured facts. Falcon 2.0 is open source. The API is publicly accessible and maintained in the servers of the TIB labs. Falcon 2.0 has been empirically evaluated on three benchmarks, and the outcomes suggest that it is able to overcome the state of the arts. Albeit promising, the experimental results can be improved. In the future, we plan to continue researching novel techniques that enable adjusting the catalog of

[11] https://www.wikidata.org/wiki/Wikidata:Statistics

[12] https://www.wikidata.org/wiki/Help:Wikimedia_language_codes/lists/all

[13] https://www.tib.eu/en/research-development/scientific-data-management/

[14] https://www.tib.eu/en/tib/profile/

[15] https://github.com/SDM-TIB

Table 5: Sample Questions from LC-QuAD 2.0 datset. The table shows five sample questions and associated gold standard relations. These sentences do not include standard sentential relations in the English language. Considering Wikidata is largely authored by the crowd, the crowd often creates such uncommon relations. `Falcon 2.0` finds limitation in linking such relations, and most results are empty.

Question	Gold Standard IDs	Gold Standard Labels	Predicted IDs	Predicted Labels
Which is the global-warming potential of dichlorodifluoromethane?	P2565	global warming potential	[]	_
What is the AMCA Radiocommunications Licence ID for Qantas?	P2472	ACMA Radiocommunications Client Number	P275	copyright license
What is ITIS TSN for Sphyraena?	P815	ITIS TSN	[]	_
What is the ARICNS for Fomalhaut?	P999	ARICNS	[]	_
Which is CIQUAL 2017 ID for cheddar?	P4696	CIQUAL2017 ID	[]	_

rules and alignments to the changes in Wikidata. We further plan to mitigate errors caused by a rule-based approach using machine learning approaches to aim towards a hybrid approach.

9 ACKNOWLEDGMENTS

This work has received funding from the EU H2020 Project No. 727658 (IASIS).

REFERENCES

[1] 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 *ISWC*. 722–735.

[2] Krisztian Balog. 2018. *Entity-oriented search*. Springer Open.

[3] Debayan Banerjee, Mohnish Dubey, Debanjan Chaudhuri, and Jens Lehmann. [n.d.]. Joint Entity and Relation Linking using EARL. ([n. d.]).

[4] Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *ACM SIGMOD*. 1247–1250.

[5] Yixin Cao, Lei Hou, Juanzi Li, and Zhiyuan Liu. 2018. Neural Collective Entity Linking. arXiv:1811.08603 http://arxiv.org/abs/1811.08603

[6] Alberto Cetoli, Stefano Bragaglia, Andrew D O'Harney, Marc Sloan, and Mohammad Akbari. 2019. A Neural Approach to Entity Linking on Wikidata. In *European Conference on Information Retrieval*. Springer, 78–86.

[7] Antonin Delpeuch. 2019. OpenTapioca: Lightweight Entity Linking for Wikidata. *arXiv preprint arXiv:1904.09131* (2019).

[8] Dennis Diefenbach, Thomas Tanon, Kamal Singh, and Pierre Maret. 2017. Question answering benchmarks for wikidata.

[9] Mohnish Dubey, Debayan Banerjee, Abdelrahman Abdelkawi, and Jens Lehmann. 2019. Lc-quad 2.0: A large dataset for complex question answering over wikidata and dbpedia. In *International Semantic Web Conference*. Springer, 69–78.

[10] Paolo Ferragina and Ugo Scaiella. 2010. TAGME: on-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of the 19th ACM Conference on Information and Knowledge Management, CIKM 2010, Toronto, Ontario, Canada, October 26-30, 2010*. 1625–1628.

[11] 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, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*. 2619–2629.

[12] Clinton Gormley and Zachary Tong. 2015. *Elasticsearch: The Definitive Guide: A Distributed Real-Time Search and Analytics Engine*. " O'Reilly Media, Inc.".

[13] Stefan Heindorf, Martin Potthast, Benno Stein, and Gregor Engels. 2016. Vandalism detection in wikidata. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. 327–336.

[14] 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 *EMNLP 2011*. 782–792.

[15] Emrah Inan and Oguz Dikenelli. 2018. A Sequence Learning Method for Domain-Specific Entity Linking. In *Proceedings of the Seventh Named Entities Workshop* (Melbourne, Australia). Association for Computational Linguistics, 14–21. http://aclweb.org/anthology/W18-2403

[16] Heng Ji. 2019. *Entity Discovery and Linking and Wikification Reading List*. http://nlp.cs.rpi.edu/kbp/2014/elreading.html

[17] 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*. 519–529.

[18] Changsung Moon, Paul Jones, and Nagiza F Samatova. 2017. Learning entity type embeddings for knowledge graph completion. In *Proceedings of the 2017 ACM on conference on information and knowledge management*. 2215–2218.

[19] 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. *In WISE (to appear)* (2020).

[20] Jonathan Raphael Raiman and Olivier Michel Raiman. 2018. DeepType: multilingual entity linking by neural type system evolution. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[21] Ridho Reinanda, Edgar Meij, and Maarten de Rijke. 2016. Document Filtering for Long-tail Entities. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016*. ACM, 771–780. https://doi.org/10.1145/2983323.2983728

[22] Michael Röder, Ricardo Usbeck, and Axel-Cyrille Ngonga Ngomo. 2018. Gerbil–benchmarking named entity recognition and linking consistently. *Semantic Web* 9, 5 (2018), 605–625.

[23] 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 NAACL HLT (Long Papers)*. 2336–2346.

[24] W. Shen, J. Wang, and J. Han. 2015. Entity Linking with a Knowledge Base: Issues, Techniques, and Solutions. *IEEE Transactions on Knowledge and Data Engineering* 27, 2 (2015), 443–460.

[25] Kuldeep Singh, Ioanna Lytra, Arun Sethupat Radhakrishna, Saeedeh Shekarpour, Maria-Esther Vidal, and Jens Lehmann. 2018. No One is Perfect: Analysing the Performance of Question Answering Components over the DBpedia Knowledge Graph. *arXiv:1809.10044* (2018).

[26] Kuldeep Singh, Arun Sethupat Radhakrishna, Andreas Both, Saeedeh Shekarpour, Ioanna Lytra, Ricardo Usbeck, Akhilesh Vyas, Akmal Khikmatullaev, Dharmen Punjani, Christoph Lange, Maria-Esther Vidal, Jens Lehmann, and Sören Auer. 2018. Why Reinvent the Wheel: Let's Build Question Answering Systems Together. In *Web Conference*. 1247–1256.

[27] Daniil Sorokin and Iryna Gurevych. 2018. Mixing Context Granularities for Improved Entity Linking on Question Answering Data across Entity Categories. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*. 65–75.

[28] 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)*. ACM, 1063–1064. https://doi.org/10.1145/2187980.2188242

[29] Edwin Williams. 1981. On the notions" Lexically related" and" Head of a word". *Linguistic inquiry* 12, 2 (1981), 245–274.

[30] Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2016. Joint Learning of the Embedding of Words and Entities for Named Entity Disambiguation. In *CoNLL 2016*, Yoav Goldberg and Stefan Riezler (Eds.). ACL, 250–259.

[31] Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2016. Joint Learning of the Embedding of Words and Entities for Named Entity Disambiguation. *CoRR* abs/1601.01343 (2016).

[32] 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 *EMNLP-IJCNLP 2019*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). 271–281.

[33] Yi Yang and Ming-Wei Chang. 2015. S-MART: Novel Tree-based Structured Learning Algorithms Applied to Tweet Entity Linking. In *ACL- IJCNLP (Volume 1: Long Papers)*. 504–513.

[34] Zi Yang, Elmer Garduño, Yan Fang, Avner Maiberg, Collin McCormack, and Eric Nyberg. 2013. Building optimal information systems automatically: configuration space exploration for biomedical information systems. In *22nd ACM CIKM'13, San Francisco, USA*. ACM, 1421–1430.

[35] Xinbo Zhang and Lei Zou. 2018. IMPROVE-QA: An Interactive Mechanism for RDF Question/Answering Systems. In *Proceedings of the 2018 International Conference on Management of Data, SIGMOD Conference 2018, Houston, TX, USA, June 10-15, 2018*. 1753–1756. https://doi.org/10.1145/3183713.3193555