# Old is Gold: Linguistic Driven Approach for Entity and Relation Linking of Short Text

**Ahmad Sakor**[1], **Isaiah Onando Mulang'**[2], **Kuldeep Singh**[2], **Saeedeh Shekarpour**[3], **Maria-Esther Vidal**[4], **Jens Lehmann**[2], and **Sören Auer**[4]

[1]L3S Research Center, Hannover, Germany
[2]Fraunhofer IAIS, Sankt Augustin, Germany
[3]University of Dayton, Dayton, USA
[4]TIB, Hannover, Germany
{sakor}@l3s.de,{maria.vidal,auer}@tib.eu
{isaiah.mulang.onando,kuldeep.singh}@iais.fraunhofer.de
{sshekarpour1}@udayton.edu
{jens.lehmann}@iais.fraunhofer.de

## Abstract

Short texts challenge NLP tasks such as named entity recognition, disambiguation, linking and relation inference because they do not provide sufficient context or are partially malformed (e.g. wrt. capitalization, long tail entities, implicit relations). In this work, we present the Falcon approach which effectively maps entities and relations within a short text to its mentions of a background knowledge graph. Falcon overcomes the challenges of short text using a light-weight linguistic approach relying on a background knowledge graph. Falcon performs joint entity and relation linking of a short text by leveraging several fundamental principles of English morphology (e.g. compounding, headword identification) and utilizes an extended knowledge graph created by merging entities and relations from various knowledge sources. It uses the context of entities for finding relations and does not require training data. Our empirical study using several standard benchmarks and datasets show that Falcon significantly outperforms state-of-the-art entity and relation linking for short text query inventories.

## 1 Introduction

Entity Linking (EL) task annotates surface forms in the text with the corresponding reference mentions in knowledge bases such as Wikipedia. It involves the two sub-tasks, i.e. Named Entity Recognition and Disambiguation (NER and NED) tasks. The state of the art contains considerable research body for EL from text to its Wikipedia mention (Cucerzan, 2007; Ferragina and Scaiella, 2010; Hoffart et al., 2011; Balog, 2018; Shen et al., 2015; Ferragina and Scaiella, 2010; Hoffart et al., 2014). With the emergence of Knowledge Graphs (KGs) which represent data in a higher structured and semantic format such as DBpedia (Auer et al., 2007), Freebase (Bollacker et al., 2008) and Wikidata (Vrandecic, 2012) that utilize Wikipedia as familiar knowledge source, retrieval-based applications such as question answering (QA) systems or keyword-based semantic search systems are empowered to provide more cognitive capabilities. Entity linking is a crucial component for a variety of applications built on knowledge graphs. For instance, an ideal NED tool on DBpedia recognizes the entities embedded in the question 'Who wrote the book The Pillars of The Earth?' and links them to the corresponding DBpedia entity (e.g. 'Pillars of The Earth' to `dbr:The_Pillars_of_the_Earth`)[1]. Another important NLP task is relation linking; it is about linking surface forms in text representing a relation to equivalent relations (predicates) of a KG. In our example question, an ideal relation linking (RL) tool links '`wrote`' to `dbo:author`[2]. There are existing approaches which address EL and RL tasks either jointly or independently (Miwa and Sasaki, 2014; Kirschnick et al., 2016; Wang et al., 2018; Dubey et al., 2018; Singh et al., 2017). However, they mostly fail in case of short text (e.g. question or key words based query) because the short text does not provide sufficient context which is essential for the disambiguation process. More importantly, a short text is often malformed meaning the text is incomplete,

[1]dbr is the prefix for http://dbpedia.org/resource/

[2]dbo is the prefix for http://dbpedia.org/ontology/

[1]First three authors have equal contribution.

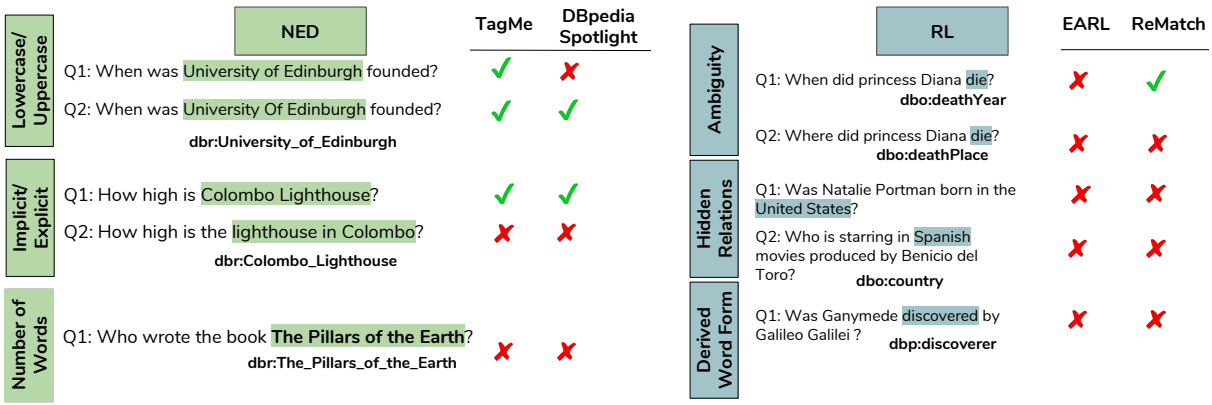


Figure 1: Performance of two EL and RL approaches on Specific Questions. TagMe and DBpedia Spotlight are the top-2 NED systems over the LC-QuAD QA dataset. However, considering short text questions, their behavior varies concerning question features, e.g., lowercase vs. uppercase, having implicit vs. explicit mappings, etc. Similar behavior has been observed for the top relation linking tools.

inexpressive, or implicit which is the case, particularly for relations in short sentences.

In this paper, we contribute to proposing a novel approach for jointly linking entities and relations within a short text into the entities and relations of DBpedia KG. This approach is robust to the challenges of short text, and moreover, it is efficient.

**Research Objectives.** Existing approaches and systems for NER, NED, EL, and RL resort to machine learning and deep learning approaches that require a large training data (Cao et al., 2018; Mudgal et al., 2018). These approaches achieve high performance on data similar to seen data. For instance, Singh et al. (2018c) evaluated 20 NED tools for question answering over the DBpedia KG including TagMe (Ferragina and Scaiella, 2012), DBpedia Spotlight (Mendes et al., 2011), Babelfy (Moro et al., 2014), and several APIs released by industry including Ambiverse (Ambiverse, 2018), TextRazor (TextRazor, 2018), and Dandelion (Dati, 2018). Among all, TagMe reports the highest F-score (0.67) over the complex question answering dataset LC-QuAD (TagMe is one of the top performing tools with an F-score of 0.91 on the generic WikiDisamb30 dataset (Ferragina and Scaiella, 2012)). Please be noted that TagMe was explicitly released for short text. However, when the input text is from a domain different from the training domain, its performance significantly falls down. Regarding the performance of various RL approaches such as ReMatch (Mulang' et al., 2017), SIBKB (Singh et al., 2017) is still low concerning accuracy and run-time even if they are purposefully developed for a particular domain or task. This deficiency is due to disregarding the context of the entities (Singh et al., 2018c,b). Therefore, when aiming for annotating entities and relations of short text, it is important to develop an approach which a) is agnostic of the requirement of large training data and b) jointly links entities and relations to its KG equivalence.

**Approach.** We target the problem of joint entity and relation linking within short text using the DBpedia KG as background knowledge. We propose a novel approach that resorts to several fundamental principles of English morphology such as compounding (Bauer and Laurie, 1983), right-hand rule for headword identification (Williams, 1981) and utilizes an extended knowledge graph created by merging entities and relations from various knowledge sources. The approach focuses on capturing semantics underlying the input text by using the context of entities for finding relations and does not require any training data. Albeit simple, to the best of our knowledge, the combination of strategies and optimization of our approach is unique. Our evaluations show that it leads to substantial gains in recall, precision, and F-score on various benchmarks and domains.

**Resource.** Falcon is available as an open Web API[3], and its source code is released to ensure reproducibility. Another open source contribution is an extended knowledge graph which we built by merging information from several sources, e.g. DBpedia, Wikidata, Oxford dictionary, and Wordnet. These contributions are in our public Github[4].

The paper is structured as follows: the next sec-

[3] https://labs.tib.eu/falcon/
[4] https://github.com/AhmadSakor/falcon

tion motivates our work by illustrating several limitations of state of the art over short text. Section 3 detailed our approach and we present evaluation results in Section 4. We describe related literature in Section 5 and Section 6 concludes our findings.

## 2 Motivating Example

We motivate our work by analyzing the performance of state-of-the-art EL and RL tools regarding query inventories on the DBpedia KG. In the following, we categorize the observed limitations.

**Effect of Capitalization on EL tools** TagMe and DBpedia Spotlight are the best two performing EL systems for question answering over DBpedia (Singh et al., 2018c). Considering the question 'When was University of Edinburgh founded', where the entity `University of Edinburgh` has one word (i.e. 'of') starting with lowercase letters. TagMe can identify this entity and link to its corresponding DBpedia entity `dbr:University_of_Edinburgh` but DBpedia Spotlight fails. However, when all words in the entity label are in uppercase, both tools recognize and link entities correctly (cf. Figure 1).

**Effect of Implicit/Explicit Entities on EL tools** The vocabulary mismatch problem (Shekarpour et al., 2017) is common for text paraphrasing and significantly affects the performance of EL approaches. In Figure 1, both EL tools can correctly link the entity in the question 'How high is Colombo Lighthouse?' but fail when the question is rephrased to 'How high is the lighthouse in Colombo?' due to the vocabulary mismatch problem. In the first representation of the question, the entity label `Colombo Lighthouse` exactly matches to the DBpedia entity `dbr:Colombo_Lighthouse` which is not the case in the rephrased question (`dbr:Colombo_Lighthouse` is expected entity for `lighthouse in Colombo`).

**Effect of the Number of Words in an Entity Label on EL tools** Long tail entities were studied as a separate phenomenon such as in news (Esquivel et al., 2017). For question answering, an increasing number of words jeopardizes entity linking performance. In our motivating example, both EL tools can not link the entity present from the question 'Who wrote the book The Pillars of the Earth?' where the entity label ('The Pillars of the Earth') has five words (a question from LC-QuAD dataset (Trivedi et al., 2017)).

**Effect of Ambiguity of Question on RL tools** EARL (Dubey et al., 2018) and Rematch (Mulang' et al., 2017) are the two top performing relation linking tools for question answering over two different datasets QALD-5 (Unger et al., 2015) and LC-QuAD respectively. In Figure 1, for the question 'When did princess Diana die', Rematch correctly recognizes the relation `die` and links it to `dbo:deathYear`. However, when the question slightly changed to "Where did princess Diana die?" in which the expected relation is `dbo:deathPlace`, both tools fail to understand the ambiguity of the question intent and cannot provide the correct DBpedia IRIs.

**Effect of Hidden Relation in a Question on RL tools** Questions are typically relatively short and sometimes there is no natural language label for the relation. For example, to correctly answer the LC-QuAD question 'Was Natalie Portman born in the United States?' contains two relations: 1) the relational label `born` needs to be linked to `dbo:birthPlace` and 2) `dbo:country` is the hidden relation for which no relation surface form is present. A similar case can be observed in another question from the same dataset 'Who is starring in Spanish movies produced by Benicio del Toro?' where one of the expected relations is `dbo:country` for which no relation label is present. For both questions, EARL and ReMatch cannot identify hidden relations.

**Effect of Derived Word Form of Relation Label on RL tools** Consider the question 'Was Ganymede discovered by Galileo Galilei?' in which the relation label `discovered` is expected to link to the DBpedia ontology `dbo:discoverer`. The word `discoverer` is the derived word form of relation label `discovered`, and due to this, both tools fail to provide correct relation linking.

## 3 The Falcon Approach

The Falcon approach maps the surface forms within the short text into the textual representation of entities in KG. This mapping follows a particular strategy which is formalized in the following. Formally, a given short text is a set of tokens $\mathcal{T} = \{t_1, ..., t_n\}$. The set of entities in KG is the union of all KG resources $\mathcal{E} = C \cup P \cup I$ (where $C, P, I$ are respectively a set of classes, properties, and instances), and $L$ is the set of literals associated with entities. The task of entity linking is

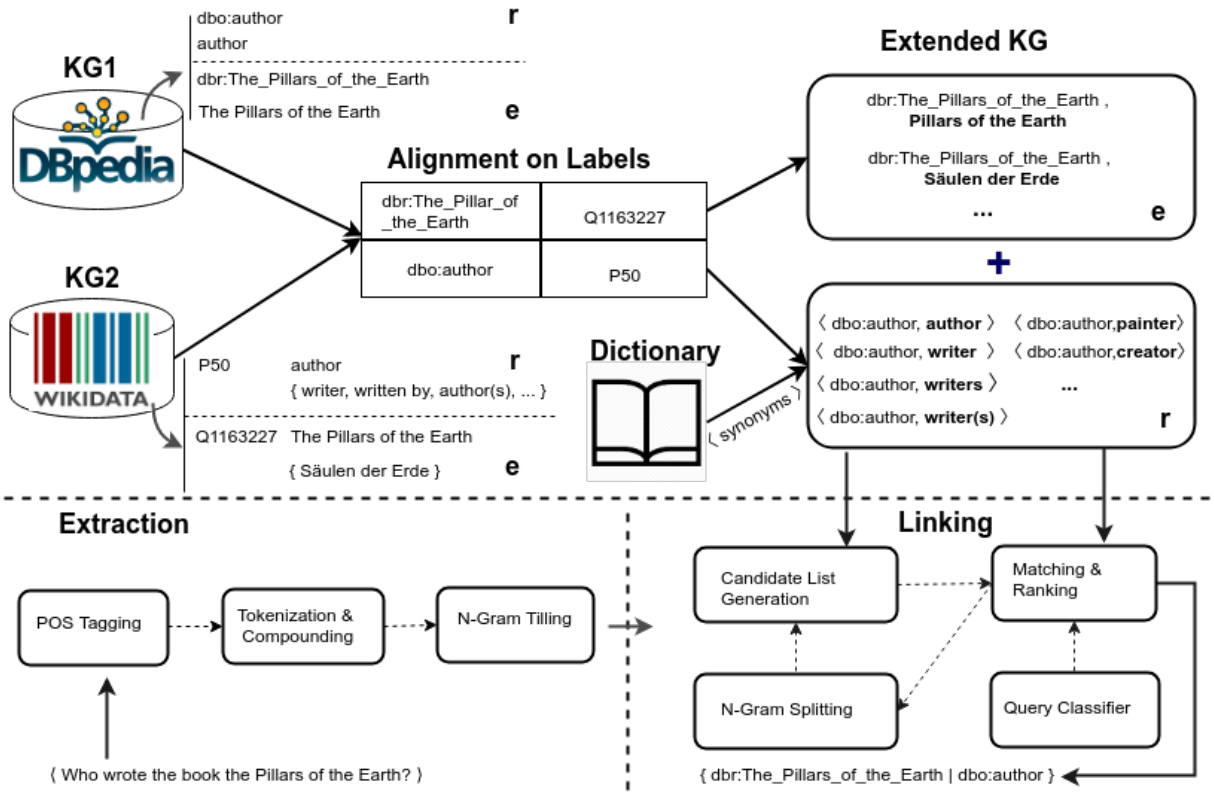


Figure 2: Overview of Falcon Approach. Falcon consists of two building blocks: 1) An extended knowledge graph which is built by merging information from various knowledge sources such as DBpedia, Wikidata, Oxford Dictionary, and WordNet. 2) Falcon architecture that has several modules focusing on surface form extraction and linking them to KG.

about mapping a subset of the input tokens denoted by $\mathcal{S} \in \mathcal{P}(\mathcal{T})$ (where $\mathcal{P}(\mathcal{T})$ is the power set of $\mathcal{T}$) to a set of entities denoted by $\mathcal{S}' \in \mathcal{P}(\mathcal{E})$ (where $\mathcal{P}(\mathcal{E})$ is the power set of $\mathcal{E}$), this mapping formally is represented as $\rho : \mathcal{S} \rightarrow \mathcal{S}'$. The Falcon approach deals with two optimization tasks as while it tries to maximize the number of tokens included in the set $\mathcal{S}$ (equation 1), it reduces the number of mapped entities in the set $\mathcal{S}'$ (eq. 2).

$$\gamma = \arg \max_{t_i \in \mathcal{A} | \mathcal{S} \in \mathcal{S}} \{\#t_i\} \quad (1)$$

$$\omega = \arg \min_{e_i \in \mathcal{A} | \mathcal{A} \in \mathcal{S}'} \{\#e_i\} \quad (2)$$

**Extended Knowledge Graph** The DBpedia KG contains over 5.6 million entities and 111 million facts (consisting of subject-predicate-object triples) which require overall 14.2GB storage (Auer et al., 2007). A major portion of this large information is not useful for EL/RL. Therefore we sliced DBpedia and extracted all the entity and relation labels to create a local KG. For example, the entity `Barack Obama`[5] in DBpedia has the natural language label 'Barack Obama' but DBpedia does not contain another representation of this label. However, the Wikidata KG is much richer and contains several aliases (or `known_as` labels) of Barack Obama such as Hussein Obama II, Barack Obama II , Obama, Barak Obama, President Obama, BHO and others[6]. We extended our local KG with this information from Wikidata. Similarly, for relation labels, the local KG is enriched with traditional linguistic resources such as Oxford dictionary (OED, 1989), and semantic dictionaries like WordNet (Miller, 1995a) to provide synonyms, derived word forms, etc. Use of background knowledge is common in question answering over DBpedia such as AskNow (Dubey et al., 2016) uses Wordnet to support relation linking. However, we also propose extending entity labels using Wikidata which is not yet used in literature. These two separate extended KGs with a total size of 1.4GB are used as an underlying

[5] `http://dbpedia.org/page/Barack_Obama`
[6] `https://www.wikidata.org/wiki/Q76`

source of knowledge and act as the core of our approach (cf. Figure 2).

**POS Tagging** In the first module illustrated in Figure 2, short input text annotated with POS tag information using spaCy (Honnibal and Johnson, 2015). This step is used primarily to identify verb and noun phrases in the sentence.

**Tokenization and Compounding** The next module creates tokens from the input sentence removing the stop words. In the first step, we break the sentence into potential tokens by removing all the stop words, and we use the stopword list provided by Fox (1990). For creating tokens, we also reuse basic compounding principle of English morphology. Compound words are lexeme that contains two or more stems (Bauer and Laurie, 1983). The words which do not have any stop words between them considered as one compound word during token formation. For example, in question "Who is the wife of Barack Obama?", Barack Obama is noun phrases which do not have any stop word between, they considered as a single compound word. Compounding allows us to reduce the total number of tokens.

**N-gram Tiling** Typically, approaches described in (Shekarpour et al., 2017, 2013) dealing with short text start with the shortest token (or N-gram) to search associated candidates in the knowledge graph. This approach is not effective when an entity has many words in its label as it creates several additional tokens. For example in question "Who wrote the book The Pillars of the Earth?", It may generate several little tokens such as book, Pillars, Earth and it will result in several potential candidates in KG. In contrast, Bill et al. (2002) applied an N-gram tiling algorithm in a question answering system to find the long answer in case of overlapping small answers. For example, answers "PQR" and "QRS" merged into single long answer "PQRS." This algorithm proceeds greedily until high scoring longest tilled N-gram found. We applied a similar approach to find the longest possible token for extracting the potential entity label. In the exemplary question " Who wrote the book The Pillars of the Earth?", The previous module generates tokens "wrote, book, Pillars, Earth." In N-gram tiling algorithm, we do not consider identified verbs of the sentence because in most cases a verb cannot be an entity label. Hence three tokens "book, Pillars, Earth" are merged as a single token. Also, verb token acts as a division point of the sentence in case of two entities, and we do not merge tokens from either side of the verb. In this process, the N-gram tiling algorithm starts with the first token from either side of the verb (which is a case of two entities in a sentence) and ends at the last non-stop word. The tiling algorithm also considers the stop words and provide the longest tilled N-gram. After N-gram tiling, we have two tokens: "wrote" and "book The Pillars of the Earth."

**Candidate List Generation** From the tokens, we create two list 1) `potential relation candidates` which contain verbs ("wrote") 2) `potential entity candidates` ("book The Pillars of the Earth"). We first search tokens of potential relation candidates in an extended KG of relations and get all the possible `DBpedia relation candidates`. Similar process has been repeated separately for `potential entity candidates` and all the `DBpedia entity candidates` are generated. For search, we use elastic search (elasticsearch, 2015) over indexed extended KG. The reason behind the use of elastic search is its effectiveness over indexed KGs as reported by Dubey et al. (2018). In few cases, it is also possible that there is no verb in a sentence (e.g. Who is the prime minister of USA?). Then, we keep the list `potential relation candidates` empty, and search all the tokens of `potential entity candidates` into extended KG of DBpedia relations because number of relations in DBpedia are very less and when tokens in `potential entity candidates` find any match, they are pushed to `potential relation candidates`.

**Candidate Ranking** To rank best DBpedia candidates, we utilize the fundamental principle of knowledge graph creation. In any knowledge graph, a sentence is represented as triple with <subject, predicate, object>. Therefore, we rank the candidates by creating a triple consisting of the relation and entity candidates from `DBpedia entity candidates` and `DBpedia relation candidates`, then check if these triples exist in the DBpedia KG. We do it by passing the triple to DBpedia SPARQL endpoint. This can be done by executing a simple Ask query against a KG endpoint which would return a boolean value indicative of the existence

of triple or otherwise of this triple. For each existing triple, we increase the weight of the entities and relations involved in the triple.

While ranking, we also consider question headwords (who, what, when, etc.) for question classification (Huang et al., 2009). Each relation in DBpedia has its domain and range associated with an entity such as person, place, date, etc. The headwords are used to determine the correct range and domain of the DBpedia relation. For example in the question "Who is starring in Spanish movies produced by Benicio del Toro?" there is a hidden relation `dbo:country` for which no surface form is present. While checking the domain of each token in relation and entity candidate lists, we can extract that word "Spanish" has the domain country; therefore, it is also an expected relation.

**N-Gram Splitting** In the previous module, if we do not get any triple in DBpedia for candidates present in `potential entity candidates` and `potential relation candidates`, we split the tokens (N-grams). To split the tokens, we again use the fundamentals of English morphology. The compound words in English have their headword always towards right side (Williams, 1981). Therefore, we start splitting tokens from "N-Gram tiling" module from the right side and pass these tokens to candidate generation module. This greedy algorithm stops when it finds triple(s) of DBpedia candidate list.

## 4 Experimental Study

**Experiment Setup.** We used a local laptop machine, with eight cores and 16GB RAM running Ubuntu 18.04 for implementation. Falcon is deployed as public API on a server with 723GB RAM, 96 cores (Intel(R) Xeon(R) Platinum 8160 CPU with 2.10GHz) running Ubuntu 18.04. This API is used for calculating all the results. The EL systems have been evaluated on different settings in literature, therefore to provide a fair evaluation we utilize Gerbil (Usbeck et al., 2015), which is a benchmarking framework for EL systems and integrated Falcon API into the Gerbil architecture. We report macro precision (P), macro recall (R), and macro F-score[7] in the tables. Falcon average run time is 1.9 seconds per question. Gerbil does not benchmark RL systems; therefore, RL systems are benchmarked using Frankenstein platform (Singh et al., 2018a). Our code, Extended KG, and data is in Github.[8]

**Datasets.** We employ two distinct datasets: 1) the LC-QuAD (Trivedi et al., 2017) dataset comprises 5,000 complex questions for DBpedia (80 percent questions are with more than one entity and relation) where average question length is 12.29 words. 2) QALD-7 (Usbeck et al., 2017) is the most popular benchmarking dataset for QA over DBpedia comprising 215 questions. In QALD, the average question length is 7.41 words and over 50% of the questions include a single entity and relation. For our linguistic based approach, we randomly selected 100 questions each from SimpleQuestions dataset (Bordes et al., 2015) and complex questions[9] for the formation of rules.

| System | Dataset | P | R | F |
|---|---|---|---|---|
| *KEA (Waitelonis and Sack, 2016)* | QALD-7 | 0.06 | 0.06 | 0.06 |
| *EARL (Dubey et al., 2018)* | QALD-7 | 0.58 | 0.60 | 0.58 |
| *FOX (Speck and Ngomo, 2014)* | QALD-7 | 0.59 | 0.57 | 0.57 |
| *Babelfy (Moro et al., 2014)* | QALD-7 | 0.40 | 0.55 | 0.44 |
| *AIDA (Hoffart et al., 2011)* | QALD-7 | 0.61 | 0.58 | 0.59 |
| *DBpedia Spotlight (Mendes et al., 2011)* | QALD-7 | 0.68 | 0.72 | 0.69 |
| *TagMe (Ferragina and Scaiella, 2012)* | QALD-7 | 0.64 | 0.76 | 0.67 |
| **Falcon** | QALD-7 | **0.78** | **0.79** | **0.78** |
| *KEA (Waitelonis and Sack, 2016)* | LC-QuAD | 0.001 | 0.001 | 0.001 |
| *EARL (Dubey et al., 2018)* | LC-QuAD | 0.53 | 0.55 | 0.53 |
| *FOX (Speck and Ngomo, 2014)* | LC-QuAD | 0.53 | 0.51 | 0.51 |
| *Babelfy (Moro et al., 2014)* | LC-QuAD | 0.43 | 0.50 | 0.44 |
| *AIDA (Hoffart et al., 2011)* | LC-QuAD | 0.50 | 0.45 | 0.47 |
| *DBpedia Spotlight (Mendes et al., 2011)* | LC-QuAD | 0.60 | 0.65 | 0.61 |
| *TagMe (Ferragina and Scaiella, 2012)* | LC-QuAD | 0.65 | 0.77 | 0.68 |
| **Falcon** | LC-QuAD | **0.81** | **0.86** | **0.83** |
| *(Singh et al., 2018c)* | LC-QuAD3253 | 0.69 | 0.66 | 0.67 |
| **Falcon** | LC-QuAD3253 | **0.73** | **0.74** | **0.73** |

Table 1: Performance of the Falcon Framework compared to various entity linking tools.

**Baselines.** The state-of-the-art outperforming tools are TagMe and DBpedia Spotlight reported in (2018c). These two systems in addition to the systems already integrated in Gerbil i.e., KEA (Waitelonis and Sack, 2016), FOX (Speck and Ngomo, 2014), Babelfy (Moro et al., 2014), AIDA (Hoffart et al., 2011) are included in our benchmark. We also report the performance of EARL (Dubey et al., 2018) for entity linking as it jointly performs EL and RL. For relation linking, the recently released EARL system is our baseline. We evaluate NED and RL systems on the LC-QuAD3253 subset of the LC-QuAD dataset (containing 3,253 LC-QuAD questions) to compare the

[7] https://github.com/dice-group/gerbil/wiki/Precision,-Recall-and-F1-measure

[8] https://github.com/AhmadSakor/falcon

[9] http://qa.mpi-inf.mpg.de/comqa/

performance with the 20 NED and five RL systems evaluated by Singh et al. (2018c). Many of these 20 tools are APIs from industry (Ambiverse (Ambiverse, 2018), TextRazor (TextRazor, 2018), and Dandelion (Dati, 2018)) which use state of the art machine learning approaches.

| QA Component | Dataset | P | R | F |
|---|---|---|---|---|
| *SIBKB (Singh et al., 2017)* | QALD-7 | 0.29 | 0.31 | 0.30 |
| *ReMatch (Mulang' et al., 2017)* | QALD-7 | 0.31 | 0.34 | 0.33 |
| *EARL (Dubey et al., 2018)* | QALD-7 | 0.27 | 0.28 | 0.27 |
| **Falcon** | QALD-7 | **0.58** | **0.61** | **0.59** |
| *SIBKB (Singh et al., 2017)* | LC-QuAD | 0.13 | 0.15 | 0.14 |
| *ReMatch (Mulang' et al., 2017)* | LC-QuAD | 0.15 | 0.17 | 0.16 |
| *EARL (Dubey et al., 2018)* | LC-QuAD | 0.17 | 0.21 | 0.18 |
| **Falcon** | LC-QuAD | **0.42** | **0.44** | **0.43** |
| *(Singh et al., 2018c)* | LC-QuAD3253 | 0.25 | 0.22 | 0.23 |
| **Falcon** | LC-QuAD3253 | **0.56** | **0.57** | **0.56** |

Table 2: Performance of the Falcon Framework compared to various Relation Linking tools.

**Performance Evaluation** Table 1 summarizes Falcon's performance compared to state-of-the-art systems integrated in Gerbil. For the QALD and LC-QuAD datasets, Falcon significantly outperforms the baseline. Similar observations are made for relation linking, where the performance of Falcon is approximately twice as high as the next best competitor on all datasets (cf. Table 2).

**Success cases of Falcon:** Falcon overcomes several major issues of short text such as capitalization of surface forms, derived word forms of relation labels and successfully handles long tail entities. For entity linking, we achieve slightly better performance on LC-QuAD than QALD. This is due to the fact that LC-QuAD questions mostly contain more than one entity and relation and thus provide more context to understand the short text. Also, major failure cases of state-of-the-art EL systems over these datasets are due to the short length and limitation to exploit the context. For example the question 'Give me the count of all people who ascended a peak in California.' (`dbr:California` is correct entity), TagMe provides two entities: `dbr:California` (for surface form California) and `dbr:Give_In_to_Me` (for "Give me"). Fundamental principles such as compounding and N-gram tiling have positive impact on the Falcon performance and we can correctly annotate several long tail entities and entities containing compound words. For example, Falcon correctly annotates question from LC-QuAD: 'Name the military unit whose garrison is Arlington County, Virginia and command structure is United States Department of Defense' where expected entities are `dbr:Arlington_County,_Virginia` and `dbr:United_States_Department_of_Defense`. Also, extended local KG has provided several interpretation of entities and their derived forms. The extended KG act as source of background knowledge during the linking process and provide extra information about entities. Generally, other entity linking tools directly map surface forms to the underlying KG using several novel techniques. However, this concept of enriching a local extended KG is not exploited in the literature and it has positively impacted the performance of the Falcon.

For relation linking, taking the context of the entities into account improved the overall performance of the Falcon. In our example question 'Who wrote the book The Pillars of the Earth?', EARL, SIBKB and Rematch aim for directly mapping `wrote` to DBpedia which results in several wrong relations such as `dbo:writer`, `dbo:creator` but when Falcon considers entity references of the question to verify which triples exist with the given entity `dbr:The_Pillars_of_the_Earth`, Falcon determines the correct relation `dbo:author`. It is important to note that existing relation linking tools completely ignore the context of the entities. Secondly, Falcon uses a fundamental principle of creating an RDF knowledge graph. While ranking the candidates in the Candidate List Ranking step, Falcon verifies the presence of the correct triple containing entity and associated relation in the KG. It has been done by cross-checking all the combinations of `potential entity candidates` and `potential relation candidates` as triple using an ASK query. Three concepts (utilization of entity context, ranking the candidates based on the presence of triple in the KG, and use of extended KG) have collectively resulted into a significant jump over other relation linking tools as observed in the Table 2.

**Failure cases of Falcon:** There are few EL cases where Falcon fails. For example, in question 'How many writers worked on the album Main Course?', the expected entity is `dbr:Main_Course`. However, Falcon returns `dbr:Critters_2:_The_Main_Course`. This is caused by compounding and the resulting

token for this question was 'album Main Course'. For the same question Falcon correctly links the relations. We further analyzed failure cases of Falcon for RL. We found that more than half of the questions which were unanswered have implicit relations. For example, for the question 'In what city is the Heineken brewery?' with the two relations `dbo:locationCity` and `dbo:manufacturer`, Falcon returns `dbo:city` as relation. There are few types of questions ('Count all the scientologists.') for which Falcon fails both for EL and RL tasks. This question is relatively short and requires reasoning to provide correct entities and relations (`dbr:Scientology` and `dbo:religion`).

## 5 Related Work

A wide range of tools and research work exist in the area of NER and NED (please see (Balog, 2018; Shen et al., 2015) for a detailed survey). Mostly, research in this domain targets news corpus, documents and Wikipedia abstract having long sentences. Such systems have been trained and benchmarked for NER/NED performance over several related datasets such as ACE2004, IITB, AIDA/CoNLL, Wiki-Disamb30, Spotlight Corpus, etc (Usbeck et al., 2015). It is important to note that most of these approaches use state of the art machine learning techniques and require a large amount of training data. However, when these tools applied to short text in a new domain such as question answering (QA) or key word based search, the performance is limited. (Singh et al., 2018c; Derczynski et al., 2015). Considering short text, the tool TagMe (Ferragina and Scaiella, 2010) is one of the popular works in this area, and uses a dictionary of entity surface forms extracted from Wikipedia to detect entity mentions in the parsed input text. These mentions passed through a voting scheme that computes the score for each mention-entity pair as the sum of votes given by candidate entities of all other mentions in the text (Ferragina and Scaiella, 2010), finally a pruning step filters out less relevant annotations. However, TagMe considered sentence length 30 for referring it as short text; in contrast for Falcon we target relatively more shorter text such as questions where average length is much less than 30 words (e.g., average question length in LC-QuAD dataset is 12.29 (Trivedi et al., 2017)). Following the popularity of KGs, scholars have shifted focus to use KGs such as DBpedia (Auer et al., 2007), Freebase (Bollacker et al., 2008) and Wikidata (Vrandecic, 2012) for the NED task. DBpedia Spotlight (Mendes et al., 2011) is one such tool that performs NED on DBpedia. After an initial step of entity spotting, DBpedia Spotlight uses contextual information to resolve the surface forms of an entity to corresponding DBpedia resources. DBpedia Spotlight has also been reused in question answering systems (Dubey et al., 2016). Relation extraction from a sentence have been long-standing research field (Zelenko et al., 2003; Bunescu and Mooney, 2005; Banko and Etzioni, 2008; Zhu et al., 2009; Fundel et al., 2007). However, linking relation label to its KG mention as independent approach is a relatively new field of research. Mulang' et al. (Mulang' et al., 2017) had the first attempt in this direction and developed Rematch. ReMatch characterizes both the properties in a KG and the relations in a question as comparable triples, then leverages both synonyms and semantic similarity measures based on graph distances from the lexical knowledge base - Wordnet (Miller, 1995b). SIBKB (Singh et al., 2017) approach for relation linking uses PATTY to derive word embeddings for a bipartite semantically indexed knowledge base which assist in RL, likewise also in full QA systems such as AskNow (Dubey et al., 2016) where PATTY is deployed as an underlying source of relation patterns. Since NER/D and RE/L are parallel tasks and the occurrence of a named entity is often accompanied by relations, recent research has attempted to perform NED and RL as a joined process. EARL (Dubey et al., 2018) is a tool for joined NED and RL that relies on Generalized Travelling Salesman Problem to find the right path between entities in the question. Several techniques exist in the literature for the collective entity and relation extraction in a text (Miwa and Sasaki, 2014; Kirschnick et al., 2016; Wang et al., 2018) but we are not aware of any other approach besides EARL that perform joint entity and relation linking to a KG.

## 6 Conclusion

In this article we presented Falcon, an approach for linking Named Entities (EL) and Relations (RL) in short text to corresponding Knowledge Graph entities. The Falcon approach adopts two novel concepts. First we demonstrated how a

fused KG comprising several complimentary semantic and linguistic resources can be employed as background knowledge. Secondly, we devised a linguistic understanding based method for processing the text, that leverages the extended background KG for EL/RL. Our comprehensive empirical evaluations provide evidence that the approach outperforms the state-of-the-art on several benchmarks. Although, we evaluate our approach on DBpedia, there is no specific assumption in our work on the structure or schema of the underlying knowledge graph, and our method should be equally applicable and can be extended to any other knowledge graph. Additionally, Falcon is offered as an online tool as well as an API.

Our approach provides considerable benefits over machine learning based approaches for short text. While Falcon achieves better results, it does not require training data and is easily adaptable to new domains. This work has highlighted the importance of background knowledge available in fused KGs as well as the linguistic understanding of the text. The linguistic methods (e.g. compounding) employed in Falcon can made more robust by using dependency parsing information. In future, we plan to explore the option of augmenting Falcon with deep learning methods for further improvement in performance specially in entity and relation extraction module.

## Acknowledgments

This work has received funding from the EU H2020 Project No. 727658 (IASIS) and partially funded from Fraunhofer IAIS KDDS project No. 500747.

## References

Ambiverse. 2018. *Ambiverse GmbH*.

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.*, pages 722–735. Springer.

Krisztian Balog. 2018. *Entity Linking*, pages 147–188. Springer International Publishing, Cham.

Michele Banko and Oren Etzioni. 2008. The tradeoffs between open and traditional relation extraction. In *Proceedings of ACL-08: HLT*, pages 28–36, Columbus, Ohio. Association for Computational Linguistics.

Laurie Bauer and Bauer Laurie. 1983. *English word-formation*. Cambridge university press.

Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*, pages 1247–1250. ACM.

Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. 2015. Large-scale Simple Question Answering with Memory Networks. *CoRR*, abs/1506.02075.

Eric Brill, Susan T. Dumais, and Michele Banko. 2002. An analysis of the askmsr question-answering system. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing, EMNLP 2002, Philadelphia, PA, USA, July 6-7, 2002*.

Razvan C. Bunescu and Raymond J. Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 724–731, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yixin Cao, Lei Hou, Juanzi Li, and Zhiyuan Liu. 2018. Neural collective entity linking. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 675–686.

Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on wikipedia data. In *EMNLP-CoNLL 2007, Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, June 28-30, 2007, Prague, Czech Republic*, pages 708–716.

Spazio Dati. 2018. *Dandelion Ltd.*

Leon Derczynski, Diana Maynard, Giuseppe Rizzo, Marieke van Erp, Genevieve Gorrell, Raphaël Troncy, Johann Petrak, and Kalina Bontcheva. 2015. Analysis of named entity recognition and linking for tweets. *Inf. Process. Manage.*, 51(2):32–49.

Mohnish Dubey, Debayan Banerjee, Debanjan Chaudhuri, and Jens Lehmann. 2018. EARL: joint entity and relation linking for question answering over knowledge graphs. In *The Semantic Web - ISWC 2018 - 17th International Semantic Web Conference, Monterey, CA, USA, October 8-12, 2018, Proceedings, Part I*, pages 108–126.

Mohnish Dubey, Sourish Dasgupta, Ankit Sharma, Konrad Höffner, and Jens Lehmann. 2016. AskNow: A Framework for Natural Language Query Formalization in SPARQL. In *The Semantic Web. Latest Advances and New Domains - 13th International Conference, ESWC 2016, Heraklion, Crete, Greece, May 29 - June 2, 2016, Proceedings*, pages 300–316. Springer.

elasticsearch. 2015. elasticsearch/elasticsearch.

José Esquivel, Dyaa Albakour, Miguel Martinez-Alvarez, David Corney, and Samir Moussa. 2017. On the long-tail entities in news. In *Advances in Information Retrieval - 39th European Conference on IR Research, ECIR 2017, Aberdeen, UK, April 8-13, 2017, Proceedings*, pages 691–697.

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*, pages 1625–1628. ACM.

Paolo Ferragina and Ugo Scaiella. 2012. Fast and Accurate Annotation of Short Texts with Wikipedia Pages. *IEEE Software*, 29(1):70–75.

Christopher J. Fox. 1990. A stop list for general text. *SIGIR Forum*, 24(1-2):19–35.

Katrin Fundel, Robert Küffner, and Ralf Zimmer. 2007. Relex—relation extraction using dependency parse trees. *Bioinformatics*, 23(3):365–371.

Johannes Hoffart, Yasemin Altun, and Gerhard Weikum. 2014. Discovering emerging entities with ambiguous names. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, pages 385–396. ACM.

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. ACL.

Matthew Honnibal and Mark Johnson. 2015. An improved non-monotonic transition system for dependency parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1373–1378.

Zhiheng Huang, Marcus Thint, and Asli Çelikyilmaz. 2009. Investigation of question classifier in question answering. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009, 6-7 August 2009, Singapore, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 543–550.

Johannes Kirschnick, Holmer Hemsen, and Volker Markl. 2016. JEDI: joint entity and relation detection using type inference. In *Proceedings of ACL-2016 System Demonstrations, Berlin, Germany, August 7-12, 2016*, pages 61–66.

Pablo N. Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. 2011. DBpedia spotlight: shedding light on the web of documents. In *Proceedings the 7th International Conference on Semantic Systems, I-SEMANTICS 2011, Graz, Austria, September 7-9, 2011*, pages 1–8. ACM.

G. Miller. 1995a. WORDNET: A lexical database for English. *Communications of the ACM*, 38(11).

George A. Miller. 1995b. WordNet: A Lexical Database for English. *Commun. ACM*, 38(11):39–41.

Makoto Miwa and Yutaka Sasaki. 2014. Modeling joint entity and relation extraction with table representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1858–1869.

Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014. Entity Linking meets Word Sense Disambiguation: a Unified Approach. *TACL*, 2:231–244.

Sidharth Mudgal, Han Li, Theodoros Rekatsinas, AnHai Doan, Youngchoon Park, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, and Vijay Raghavendra. 2018. Deep learning for entity matching: A design space exploration. In *Proceedings of the 2018 International Conference on Management of Data, SIGMOD Conference 2018, Houston, TX, USA, June 10-15, 2018*, pages 19–34.

Isaiah Onando Mulang', Kuldeep Singh, and Fabrizio Orlandi. 2017. Matching natural language relations to knowledge graph properties for question answering. In *Proceedings of the 13th International Conference on Semantic Systems, SEMANTICS 2017, Amsterdam, The Netherlands, September 11-14, 2017*, pages 89–96.

OED. 1989. *Oxford English Dictionary*, second edition. Oxford University Press.

Saeedeh Shekarpour, Edgard Marx, Sören Auer, and Amit P. Sheth. 2017. RQUERY: Rewriting Natural Language Queries on Knowledge Graphs to Alleviate the Vocabulary Mismatch Problem. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 3936–3943.

Saeedeh Shekarpour, Axel-Cyrille Ngonga Ngomo, and Sören Auer. 2013. Question answering on interlinked data. In *22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013*, pages 1145–1156.

Wei Shen, Jianyong Wang, and Jiawei Han. 2015. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Trans. Knowl. Data Eng.*, 27(2):443–460.

Kuldeep Singh, Andreas Both, Arun Sethupat Radhakrishna, and Saeedeh Shekarpour. 2018a. Frankenstein: A Platform Enabling Reuse of Question Answering Components. In *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, pages 624–638. Springer.

Kuldeep Singh, Ioanna Lytra, Arun Sethupat Radhakrishna, Saeedeh Shekarpour, Maria-Esther Vidal, and Jens Lehmann. 2018b. No one is perfect: Analysing the performance of question answering components over the dbpedia knowledge graph. *CoRR*, abs/1809.10044.

Kuldeep Singh, Isaiah Onando Mulang', Ioanna Lytra, Mohamad Yaser Jaradeh, Ahmad Sakor, Maria-Esther Vidal, Christoph Lange, and Sören Auer. 2017. Capturing knowledge in semantically-typed relational patterns to enhance relation linking. In *Proceedings of the Knowledge Capture Conference, K-CAP 2017, Austin, TX, USA, December 4-6, 2017*, pages 31:1–31:8.

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. 2018c. Why Reinvent the Wheel: Let's Build Question Answering Systems Together. In *Proceedings of the 2018 World Wide Web Conference, WWW 2018, Lyon, France, April 23-27, 2018*, pages 1247–1256. ACM.

René Speck and Axel-Cyrille Ngonga Ngomo. 2014. Ensemble learning for named entity recognition. In *The Semantic Web - ISWC 2014 - 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I*, pages 519–534.

TextRazor. 2018. *TextRazor Ltd.*

Priyansh Trivedi, Gaurav Maheshwari, Mohnish Dubey, and Jens Lehmann. 2017. LC-QuAD: A Corpus for Complex Question Answering over Knowledge Graphs. In *The Semantic Web - ISWC 2017 - 16th International Semantic Web Conference, Vienna, Austria, October 21-25, 2017, Proceedings, Part II*, pages 210–218. Springer.

Christina Unger, Corina Forascu, Vanessa López, Axel-Cyrille Ngonga Ngomo, Elena Cabrio, Philipp Cimiano, and Sebastian Walter. 2015. Question Answering over Linked Data (QALD-5). In *Working Notes of CLEF 2015 - Conference and Labs of the Evaluation forum, Toulouse, France, September 8-11, 2015*. CEUR-WS.org.

Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Bastian Haarmann, Anastasia Krithara, Michael Röder, and Giulio Napolitano. 2017. 7th open challenge on question answering over linked data (QALD-7). In *Semantic Web Challenges - 4th SemWebEval Challenge at ESWC 2017, Portoroz, Slovenia, May 28 - June 1, 2017, Revised Selected Papers*, pages 59–69.

Ricardo Usbeck, Michael Röder, Axel-Cyrille Ngonga Ngomo, Ciro Baron, Andreas Both, Martin Brümmer, Diego Ceccarelli, Marco Cornolti, Didier Cherix, Bernd Eickmann, Paolo Ferragina, Christiane Lemke, Andrea Moro, Roberto Navigli, Francesco Piccinno, Giuseppe Rizzo, Harald Sack, René Speck, Raphaël Troncy, Jörg Waitelonis, and Lars Wesemann. 2015. GERBIL: general entity annotator benchmarking framework. In *Proceedings of the 24th International Conference on World Wide Web, WWW 2015, Florence, Italy, May 18-22, 2015*, pages 1133–1143.

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. ACM.

Jörg Waitelonis and Harald Sack. 2016. Named entity linking in #tweets with KEA. In *Proceedings of the 6th Workshop on 'Making Sense of Microposts' co-located with the 25th International World Wide Web Conference (WWW 2016), Montréal, Canada, April 11, 2016.*, pages 61–63.

Shaolei Wang, Yue Zhang, Wanxiang Che, and Ting Liu. 2018. Joint extraction of entities and relations based on a novel graph scheme. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden.*, pages 4461–4467.

Edwin Williams. 1981. On the notions" lexically related" and" head of a word". *Linguistic inquiry*, 12(2):245–274.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *J. Mach. Learn. Res.*, 3:1083–1106.

Jun Zhu, Zaiqing Nie, Xiaojiang Liu, Bo Zhang, and Ji-Rong Wen. 2009. Statsnowball: A statistical approach to extracting entity relationships. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 101–110. ACM.