# An Annotated Corpus based on Wikipedia

**Michael Strobl, Amine Trabelsi, Osmar Zaiane**
University of Alberta
Edmonton, AB
mstrobl@ualberta.ca, atrabels@ualberta.ca , zaiane@ualberta.ca

### Abstract

In this paper, we are discussing an approach in order to create a text corpus based on Wikipedia with exhaustive annotations of entity mentions. Editors on Wikipedia are only expected to add hyperlinks in order to help the reader to understand the content, but are discouraged to add links that do not add any benefit for understanding an article. Therefore, many mentions of popular entities (such as countries or popular events in history), previously linked articles as well as the article entity itself, are not linked. This results in a huge potential for additional annotations that can be used for downstream NLP tasks, such as Relation Extraction. We show that our annotations are useful for creating distantly supervised datasets for this task. Furthermore, we publish all code necessary to derive a corpus from a raw Wikipedia dump, so that it can be reproduced by everyone.

**Keywords:** Wikipedia, Knowledge Graphs, Relation Extraction

## 1. Introduction

Understanding factual knowledge from text data is a crucial skill towards mastering several high-level NLP tasks, such as Response Generation for conversational agents (e.g. (Logan et al., 2019)) or Question Answering (e.g. (Yih et al., 2015)). Elements of factual knowledge can include the following items:

- What are the entities in context?
- What is happening? Typically an event about how entities are interacting with or related to each other or what they are doing. (”John married Lisa”, ”John was born in New York City”)
- When is this happening? Events have a timestamp or a period of time is attached. (”John was born on November 11th, 1976”)

Apart from these three elements, text can also contain other numerical data, e.g. quantities with units, that express something of interest and are not just a sequence of distinct words as usually assumed in Language Modeling.
In order to achieve the task of extracting factual knowledge from text or conversations, several subtasks have to be mastered (either pipelined or in an end-to-end fashion):

1. Entity Recognition (ER) (e.g. (Lample et al., 2016)): Typically formulated as Named Entity Recognition (NER) task aiming to find all named entities in text. Different tagging schemes are possible, although the most common scheme is the CoNLL-2003 scheme (Sang and De Meulder, 2003) with 4 classes: Person, Organization, Location and Miscellaneous (entities that are related to entities from other classes, e.g. events).
2. Entity Linking (EL) (e.g. (Sil et al., 2018)): Linking known entities to unique identifiers in a Knowledge Graph (KG), such as Wikipedia.[1].
3. Co-reference Resolution (CR) (e.g. (Lee et al., 2018)): Finding mentions of entities in text that refer to named entities, such as *he*, *she*, *it* or *the company*, *the prince*.
4. Relation Extraction (RE) (e.g. (Takanobu et al., 2019)): Finding interactions between entities either using a fixed set of relations or words from text. For example, potential relations to extract include family relations, such as *spouse* or *issue*.
5. Creating a Knowledge Graph (KG): Assuming every entity has a unique identifier attached (either from a known KG or internally), a graph can be created using the KG identifiers and the extracted relations connecting them, using, for example, the Resource Description Framework (RDF)[2].

Afterwards, the created KG can be used for downstream tasks, such as fact-aware language modeling (Logan et al., 2019). However, corpora with labeled data including all of the aforementioned subtasks are hard to come by since manual annotation of entities (with KG identifiers, if available), co-references and their relations to each other is time-consuming and therefore only limited amounts of data exist. Typically, the higher level the task is, the less data is available.
Our goal in this paper is to create a large annotated corpus based on Wikipedia, which already contains millions of annotations (i.e. mentions in text, linked to their article name in Wikipedia) created by humans and we show that many more can be extracted with in order to increase the number of annotations per entity. This is crucial for tasks such as RE, in which more context for specific entities is desirable in order to create a KG with as much knowledge about an entity as possible. We are applying an extensive set of rules in order to create more annotations through finding co-references of already annotated entities as well as finding entities that are not linked or do not exist in Wikipedia. If conflicts appear (several entity candidates for a single mention), we use a modified version of existing

[1] `https://www.wikipedia.org/`

[2] `https://www.w3.org/RDF/`

neural-network-based Entity Linker for linking these mentions to its Wikipedia article.

The corpus created this way can later be used in order to extract datasets for RE using Distant Supervision (DS) (see, for example, (Bunescu and Mooney, 2007) and (Mintz et al., 2009)).

This is a summary of our research contributions:

- Using a current Wikipedia dump[3], we provide code in order to create a large annotated corpus through extracting all articles and annotate additional mentions of entities including their co-references, which are typically not linked in Wikipedia. Whenever a new dump is released, a new updated corpus can be created.
- This corpus can be especially used for creating Distantly Supervised Relation Extraction datasets for a wide variety of relations.

It is possible to apply our approach to Wikipedia for all languages since only minimal language-dependent assumptions are made (only regarding frequent words starting words in English sentences). However, we show the creation of such a corpus based on the English version of Wikipedia.

The remainder of this article is organized as follows. Wikipedia and related statistics are presented in section 2 . In section 3 we present related work, including similar corpora based on the English Wikipedia. Section 4 presents the steps that are necessary to create an annotated corpus. In section 5 we present experimental results indicating why this corpus can be useful to the community and the article is concluded in section 6 , including future work.

## 2. Wikipedia

Wikipedia is a free encyclopedia that exists for 307 languages[4] of varying content size. Table 2 shows statistics about the 10 largest versions (the Swedish and Cebuano Wikipedias were largely created by a bot[5]). Although Wikipedia exists for 307 languages, only $\approx$ 5% of all Wikipedias contain more than 1,000,000 articles and $\approx$ 37% contain only 1,000 to 9,999 articles. Nevertheless, Wikipedia is a Knowledge Base of impressive size with almost 6 million articles in the English version. This leads to a huge potential for NLP research, e.g. as shown by (Ghaddar and Langlais, 2017) for NER.

Wikipedia contains many more pages than articles, such as redirect or disambiguation pages as seen in table 3. In the following, we explain certain, for our approach important, features of Wikipedia and the annotation scheme proposed for editors:

- Redirect pages: Wikipedia contains many redirect pages, e.g. *NYC* or *The City of New York* referring to the article of *New York City*. Editors can create these pages through adding alternative names for an article or they are created automatically, e.g. in case the name of an article changes and therefore broken links can be avoided through creating a redirect page.
- Disambiguation pages: Wikipedia contains many disambiguation pages, which are similar to redirect pages, except they deal with mentions that are knowingly referring to several different articles. For example, the disambiguation page of *New York*[6] refers to a whole list of articles including the city, state and many sports clubs located in New York City or the state of New York.
- Typically, entities are only linked once in an article when they are mentioned first. Subsequent mentions should not be linked anymore.[7] In addition to that, pages do not contain links to themselves, e.g. there is not link to the article of *Barack Obama* within itself, although he is mentioned in there many times.
- Links can consist of two parts: (1) The article name

| Language | Articles | Edits | Active editors |
|---|---|---|---|
| English | 5,971,985 | 922,228,758 | 137,092 |
| Cebuano | 5,378,807 | 29,550,021 | 156 |
| Swedish | 3,745,430 | 46,482,187 | 2,520 |
| German | 2,366,596 | 192,825,026 | 19,260 |
| French | 2,156,551 | 164,189,535 | 19,649 |
| Dutch | 1,983,049 | 54,935,619 | 4,075 |
| Russian | 1,579,561 | 103,113,453 | 11,361 |
| Italian | 1,565,874 | 108,671,261 | 8,564 |
| Spanish | 1,558,170 | 120,965,543 | 18,299 |
| Polish | 1,369,738 | 57,787,765 | 4,308 |

Table 1: Statistics of the 10 largest Wikipedias.

| No. articles | Languages |
|---|---|
| 1,000,000+ | 16 |
| 100,000+ | 46 |
| 10,000+ | 84 |
| 1,000+ | 114 |
| 100+ | 37 |
| 10+ | 0 |
| 1+ | 8 |
| 0 | 2 |

Table 2: Distribution of number of articles and language versions.

| Page type | No. |
|---|---|
| Redirects | 8,440,863 |
| Disambiguations (other) | 189,124 |
| Disambiguations (geo) | 38,507 |
| Disambiguations (human) | 59,988 |

Table 3: Number of important pages other than articles.

[3] https://dumps.wikimedia.org/enwiki/latest/

[4] As of 19 November 2019: https://meta.wikimedia.org/wiki/List_of_Wikipedias

[5] https://en.wikipedia.org/wiki/Lsjbot

[6] https://en.wikipedia.org/wiki/New_York

[7] https://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style/Linking

the link refers to (mandatory) and (2) an alias for that article since it is not always convenient to include the linked article's full name. This could look like the following link (following the linking scheme of Wikipedia): *[[Barack Obama|Obama]]*, resulting in a hyperlink with anchor text *Obama*, linking to article *Barack Obama*.

- Links, in general, should help the reader to understand the article and therefore should only be added if helpful (overlinking should be avoided). This also means that articles, most readers are familiar with, such as countries, locations, languages, etc., should not be linked.

Wikipedia's linking scheme aims for the best readability, but in order to be useful for NLP tasks, more annotations can be helpful and are possible, as we show in this article. Therefore, our approach aims for an exhaustively annotated corpus, based on Wikipedia. And by exhaustively annotated we mean that all mentions of entities, whether or not they have an article in Wikipedia, and their co-references are annotated.

## 3. Background and Related Work

There are two main lines of previous work that are important to our method: (1) Datasets with annotations similar to ours (mostly for NER) and (2) semi-automatically extracted datasets for Relation Extraction (RE) using Distant Supervision (DS). Both are described below.

### 3.1. Wikipedia Annotated Corpora for NER

In (Nothman et al., 2008) entities in Wikipedia are classified into one of the CoNLL-2003 classes, i.e. Person, Organization, Location or Miscellaneous, in order to create a large annotated corpus based on the English Wikipedia. Already linked entities are classified and additional links are identified through the use of 3 different rules:

1. The title of an article and all redirects are used to find more links of this article.
2. If article A is of type PER, the first and last word are considered as alternative title for article A.
3. The text of all links linking to article A is used as alternative title as well.

Their work is based on a 2008 Wikipedia dump. Co-references such as *he* or *she* or others that can be used to refer to certain entities are not considered, if not already in Wikipedia (typically not the case).

Ghaddar and Langlais (2017) created the WiNER corpus and follow a similar approach using a 2013 Wikipedia dump. Their annotation pipeline mainly follows the one of (Nothman et al., 2008), although conflicts (certain words may refer to multiple entities) are resolved through linking such a mention to the closest already linked article before or after. While easy to implement, this rule does not hold in general.

The resulting corpora of both systems are evaluated using common NER approaches and corpora and showed a slightly better result than training an NER system on other typically smaller datasets. Even though both datasets are publicly available, classifying entities into one of those 4 classes introduces errors and the original annotations are removed and cannot be obtained anymore. This removes valuable information, e.g. for creating distantly supervised RE corpora or training CR systems, since it is not clear which annotation refers to which Wikipedia article.

The closest to our approach is (Klang and Nugues, 2018), which uses an EL system using a pruning strategy based on link counts in order to keep the number of candidates per mention low (after running a mention detection algorithm) and PageRank (Brin and Page, 1998) combined with a neural network classifier to link mentions to their Wikipedia articles. Furthermore, articles with all linked mentions are indexed and visualized online.[8] The authors did not publish the code and the data is not publicly available (not downloadable), except through their website, therefore it is not possible to compare against this approach. In addition to that, co-references are not resolved.

Another similar but smaller dataset is the *Linked WikiText-2* dataset from (Logan et al., 2019), which is publicly available[9]. It consists of only 720 articles (600 train, 60 dev, 60 test) and was created using a neural EL system (Gupta et al., 2017) as well as a CR system (Manning et al., 2014) in order to create additional annotations (apart from the already given ones from the editors of the articles). However, using automatic tools introduce additional errors, especially since both tools are trained on non-annotated data, whereas Wikipedia can be considered as partially annotated data and therefore it would certainly be beneficial to consider these annotations, as we do and was done by (Nothman et al., 2008) and (Ghaddar and Langlais, 2017). The main issue with using an Entity Linker in such an unrestricted way is that it tries to link all mentions of entities, regardless whether they have an article in Wikipedia.

### 3.2. Distant Supervision

Another line of work, relevant to ours, is extracting datasets from large text corpora for RE using DS.

Due to the lack of large datasets with entities as well as their relations annotated, Mintz et al. (2009) proposed to link entities in text corpora to their corresponding entries in a KG, e.g. Freebase (Bollacker et al., 2008), and whenever two entities that are linked in the KG appear in the same sentence, this sentence is considered as expressing this relationship in some ways. They created a dataset with 10,000 instances and 102 Freebase relations using Wikipedia, although entities are tagged with an NER and existing annotations are ignored. The reported human-evaluated precision of the extracted instances is 67.7%. Using Wikipedia annotations may help here, instead of relying on an NER.

Riedel et al. (2010), follow a similar approach, except that their assumption is slightly different. Instead of assuming every sentence expresses a certain relation, it is assumed that, considering a set of sentences mentioning two specific

[8] `http://vilde.cs.lth.se:9001/en-hedwig/` for the English Wikipedia.

[9] `https://rloganiv.github.io/linked-wikitext-2`

entities, at least one of them expresses the relation of interest. They created an RE corpus based on the New York Times Annotated Corpus[10]. However, they run a manual inspection on a dataset based on Wikipedia as well as the New York Times corpus comparing the number of violations of the distant supervision assumption on three relations and found that it is a lot less often violated in Wikipedia ($\approx 31\%$ vs. $\approx 13\%$). This indicates that Wikipedia can indeed provide DS data for RE systems and high-quality annotations presumably lead to better extractions. As we show in section 5 , our approach is capable of extracting many more relevant sentences than only using Wikipedia without additional annotations.

## 4. Method

In this section we present our method in order to annotate as many mentions as possible in Wikipedia articles.
In order to illustrate the problem our approach aims to solve, consider the following sentence from *Tony Hawk*'s Wikipedia article[11] (original annotations in blue):

> Tony Hawk was born on May 12, 1968 in San Diego, California to Nancy and Frank Peter Rupert Hawk, and was raised in San Diego.

However, apart from the entity ”San Diego, California”, a few other non-annotated mentions of entities appear just in this sentence: (1) *Tony Hawk*, the entity of the current article, (2) his parents *Nancy Hawk* and *Frank Peter Rupert Hawk* (both currently not in Wikipedia), as well as (3) another mention of *San Diego, California*. Therefore, if correctly annotated, this sentence includes 5 mentions of entities, although only a single one is already annotated.
In general, editors should avoid to link mentions if they refer to entities that were already linked before or if they refer to very popular entities, and linking them would not contribute to understanding the meaning of a sentence[12]. However, our approach aims to exhaustively annotate all mentions in Wikipedia in order to create an annotated text corpus that can be more useful in downstream tasks, than solely relying on already existing links or using an NER and EL to find and link more mentions, which introduces unnecessary errors.
In order to achieve this task, it can be broken down into several subtasks, including dictionary generations for linking mentions to articles, initial annotations, co-reference resolution and candidate conflict resolution, which is described below.

### 4.1. Dictionary Creation

We are focusing only on Named Entities (similar to (Ghaddar and Langlais, 2017)) and therefore we only keep entities and their articles that typically start with a capital letter (considering the most frequent anchor text of incoming links of an article).

The first step of our approach is to create the following dictionaries that help to do the initial Mention Detection (MD) and EL, considering the hyperlinks that are added by the editors of the article:

- Redirects dictionary: Wikipedia contains many redirect pages, e.g. *NYC*[13] or *The City of New York*[14] referring to the article of *New York City*. These redirects are useful alternative titles that, presumably, solely refer to a certain entity (otherwise it would not be a redirect page, it could be a disambiguation page instead if several candidate articles are possible). Redirect pages can be either created by an editor to provide alternative titles or they are created automatically in case the title of an article changes.[15]

- Alias dictionary: This is another dictionary containing alternative names for articles, created through collecting anchor texts of hyperlinks referring to an article, e.g. *U.S.* and *USA* are both included in the alias dictionary since they appear in other Wikipedia articles linking to the article of the *United States*. Overlaps with the redirects dictionary are possible, but typically the alias dictionary contains more ambiguous aliases, e.g. *New York* referring to the articles *New York City*, *New York (state)* and many more. We only keep aliases that start with a capital letter, since only these can refer to Named Entities, and we ignore alias-entity links that only appear once in Wikipedia, since these are often not meaningful and can introduce unnecessary errors.

- Disambiguation page dictionaries: Wikipedia contains many disambiguation pages, which are similar to redirect pages, except they deal with mentions that knowingly refer to several different articles, e.g. the disambiguation page *New York* refers to a whole list of articles including the city, state and many sports clubs located in New York City or the state of New York. Often, these disambiguation pages have a certain type attached, mainly *human*[16] for persons or *geo*[17] for geopolitical entities. In case the page contains several types of entities, such as ”New York”, it typically does not fall under one of these two categories. However, if it does, it is useful information used by our approach, in case a person is mentioned matching a page in this dictionary is mentioned, but from the article it is not clear who it is and it might not even be a person with an article in Wikipedia.

- We are ignoring stub articles[18], which are articles that

[10] https://catalog.ldc.upenn.edu/LDC2008T19
[11] https://en.wikipedia.org/wiki/Tony_Hawk
[12] https://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style/Linking

[13] https://en.wikipedia.org/w/index.php?title=NYC&redirect=no
[14] https://en.wikipedia.org/w/index.php?title=The_City_of_New_York&redirect=no
[15] https://en.wikipedia.org/wiki/Wikipedia:Redirect
[16] https://en.wikipedia.org/wiki/Template:Human_name_disambiguation
[17] https://en.wikipedia.org/wiki/Template:Place_name_disambiguation
[18] https://en.wikipedia.org/wiki/Wikipedia:Stub

are very short and do not contain a lot of information. Usually, these are articles that are just started, but none of the editors has taken a closer look into it, and therefore the expected quality could be lower than the more popular and longer articles.

- Frequent sentence starter words: We collected a list of frequent sentence starter words that should not be considered for starting a mention at the beginning of a sentence, similarly to (Nothman et al., 2008). We used the PUNKT sentence tokenizer (Kiss and Strunk, 2006) from the NLTK Python package (Bird et al., 2009) in order to tokenize each article and collect the most frequent sentence starter words. We ended up with a list of 1760 words that, if starting a sentence and are not part of a multi-word mention, should not be considered as entity mention that should be linked to an article.
- We compiled a list of persons from the Yago KG (Mahdisoltani et al., 2013) in order to figure out whether an article refers to a person, in which case the first and last word of the name can be also considered as alternative name for that person, again, as done in (Nothman et al., 2008).
- Often very popular entities, such as countries or certain events (”World War II”) are mentioned in text without being linked. In order to link these mentions with high confidence we kept a dictionary of the 10,000 most popular articles (regarding the number of incoming hyperlinks found in all of Wikipedia) that can be linked to mentions, without being linked in the text at all.
- Wikipedia contains many articles about given names. We collect all articles for this dictionary through looking for the categories ”Given names”, ”Masculine given names” or ”Feminine given names”.

## 4.2. Direct Mention Annotations

We apply a relatively extensive set of rules in order to annotate mentions in Wikipedia articles, compared to just 3 or 4 as done in (Ghaddar and Langlais, 2017) or (Nothman et al., 2008). Apart from keeping the links corresponding to articles mostly starting with capital letters, we are detecting new potential mentions of articles using the following rules (applied to text between already annotated mentions):

1. The first line of an article often contains mentions of the article entity in **bold**, representing alternative names and are therefore annotated with the article entity.
2. At any time throughout processing an article, we are keeping an alias dictionary of alternative names for each linked article up until this point. This includes all aliases in the alias dictionary, all redirects and first and last word of an article in case it is a person. Since each article should be linked once, when it was mentioned first, by the author, these alternative names can be searched for throughout the text coming after an article was seen. If a match was found, the found alias can be annotated with the matching article.
3. We are searching for acronyms using a regular expression, for example, strings such as ”Aaaaaa Baaaaa Cccccc (ABC)”, linking the acronym to the matching string appearing before the brackets, which was linked to its article before.
4. In general, all words starting with a capital letter are detected as mentions, if not falling under the frequent sentence starter rule.
5. Pairs of mentions detected that way are combined if they are right next to each other or, if combined, are part of the alias dictionary or consist of the following words in between the pair of mentions: *de*, *of*, *von*, *van*.
6. In many cases, the corresponding article for a mention was not linked in the text before (or cannot even be found in Wikipedia) and therefore the following rules are applied in these cases:
   - If the mention matches an alias of the current articles’ main entity and does not exactly match any other entities, it is linked to it.
   - If the article matches one of the 10,000 most popular entities in Wikipedia, the mention is linked to this article.
   - If it matches a disambiguation page and one of the previously linked articles appears in this page, the mention is linked to this article.
   - If the mention matches an alias from the general alias dictionary, it is linked to the most frequently linked entity given the mention.
7. We also apply rules in case there are conflicts (more than one potential candidate for a mention using previous rules):
   - If all candidates correspond to persons (sometimes people with the same first or last names appear within the same article), the person that was linked with the current mention more often, is used as annotation.
   - If a mention matches an alias of the current articles’ main entity and more entities in the current alias dictionary, these are discarded in case the corresponding articles do not match the mention exactly.
   - Otherwise, in some cases conflicts cannot be solved this way and EL has to be used (see section 4.3.).
8. If no entity can be found these ways, the mention is annotated as *unknown entity* or, in case a disambiguation page matches, this page is used and it sometimes contains the information that the mention corresponds to a person or geopolitical entity.

### 4.3. Candidate Conflict Resolution

Even after applying the rules explained in section 4.2., it is still possible to end up with multi-candidate mentions, which the following sentence (from Wikipedia[19]) illustrates:

> Leicestershire are in the second division of the County Championship and in Group C of the Pro40 one day league."

*Leicestershire* in this sentence refers to the *Leicestershire County Cricket Club* although an alternative candidate, which is exactly matching, would be *Leicestershire*[20], which was mentioned in a previous sentence within the same article.

In order to resolve these conflicts we use the Entity Linker from (Gupta et al., 2017), as used for the *Linked Wikilinks-2* dataset as well. The authors made the code and the used models publicly available[21]. However, we only use the Linker for multi-candidate mentions and not to find and link all mentions (which leads to errors if a mention refers to an entity that is absent in Wikipedia since the system always finds a link) and we modified it in a way that it only considers our candidate set for linking, not the candidate set from its own alias dictionary, since this set would include many more articles that are presumably not relevant.

### 4.4. Co-reference Resolution

Co-references that are not named entities (do not start with a capital letter), such as *he* or *she* for humans, *the station* for *Gare du Nord*[22] or *the company* for *General Electric*[23], should be linked as well. Our approach to achieve this is explained below.

As mentioned before, CR systems work increasingly well, although their performance is still far behind the performance of, for example, NER tools (see (Akbik et al., 2019) and (Lee et al., 2018)). We experimented with the state-of-the-art system from (Lee et al., 2018), but there are two main issues with it: (1) For long articles it takes several seconds to process (AMD Ryzen 3700x, 64gb, Nvidia GeForce RTX 2070) and is therefore way too slow to annotate approximately 3,000,000 articles within a reasonable amount of time. (2) The model was trained in a fully open way, i.e. it has to find all mentions of entities and create one cluster per distinct entity, which is a very hard task. Whereas in our setting, many mentions are already annotated and a system only has to figure out whether there are more mentions (non-named entities) of the annotated entities in the article. Therefore, we decided to use a simple rule-based system.

Our system for CR considers a small set of co-references, depending on the type of entity. In order to find the type of an entity, we use the Yago KG (Mahdisoltani et al., 2013) in order to retrieve all types of each entity. Yago is a system that links Wikipedia articles to a set of types, which consists of Wikipedia categories, as well as words from WordNet (Miller, 1995). In case, an entity is not of type "person", all WordNet types are considered as co-references, with "the" as prefix, e.g. *the station* (Gare du Nord) or *the company* (General Electric). For persons, *he*, *she*, *her*, *him* and *his* are considered as co-references. Also sometimes the article name includes the type of an entity, which can be considered as a co-reference as well, e.g. the type of *Audrey (band)*[24] is *band*. This results in an initial dictionary with zero or more co-references per article in Wikipedia.

In order to find out which of the co-references in the initial dictionary is actually used, we simple searched and counted each co-reference for each article. For example, we found that *General Electric* has type *company* in Yago and *the company* appears 19 times in its article. If a co-reference appeared more than a user-defined threshold, it was accepted. For persons, we looked for *he* and *she*, and if one of them appeared more than a threshold, the one appearing more often was accepted. This includes *his*, *him* and *her* as well, depending on the previous decision. We found setting this threshold to 2 worked reasonably well. This resulted in a dictionary of at least one co-reference for 825,100 entities in Wikipedia.

Using this dictionary, we can add more annotations to our corpus using the following procedure for each article:

1. Processing an article sequentially and whenever an already annotated entity mention appears, all its co-references are added to the current co-reference dictionary.

2. The text in between two entities (or start/end of article) are tagged using this co-reference dictionary, making sure that only previously mentioned entities can be used.

At any time, there can only be one entity attached to a certain co-reference, which effectively results in using the article matching a co-reference that appeared most recently in the previous text. Mentions that do not have a Wikipedia article, but were still annotated are classified into male of female human or something else using the gender guesser package[25]. We classify a mention as male, if there are no female-classified words in the mention and vice-versa for female mentions. Otherwise, the mention is not considered for annotation.

## 5. Evaluation

Evaluating the quality of such annotations is difficult since to the best of our knowledge a similarly annotated corpus with manual exhaustive annotations does not exist and therefore it is not possible to directly compare. However, in this section we will show statistics collected from such a corpus and the usefulness of a dataset created by our approach for the task of creating datasets for RE using DS.

[19] `https://en.wikipedia.org/wiki/Leicestershire_County_Cricket_Club`

[20] `https://en.wikipedia.org/wiki/Leicestershire`

[21] `https://github.com/nitishgupta/neural-el`

[22] `https://en.wikipedia.org/wiki/Gare_du_Nord`

[23] `https://en.wikipedia.org/wiki/General_Electric`

[24] `https://en.wikipedia.org/wiki/Audrey_(band)`

[25] https://pypi.org/project/gender-guesser/

| | Basic annotations | All annotations |
|---|---|---|
| Articles | 2,952,439 | 2,952,439 |
| Found mentions | 64,654,332 | 265,498,363 |
| Per sentence | 0.38 | 1.56 |
| Per article | 21.90 | 89.93 |
| Article entity per article | 0.93 | 12.7 |

Table 4: Annotation statistics for the *Basic* and *All annotations* corpora in absolute numbers (*Articles* and *Found mentions*) as well as the average number of annotations per article, per sentence and average number of article entities per article.

| | |
|---|---|
| Entity pairs in Dbpedia | 6,651,996 |
| Relevant triples in Dbpedia | 7,207,740 |
| Dbpedia relations | 12691 |

Table 5: Statistics from DBpedia with entity pairs, triples and relations relevant for RE datasets.

### 5.1. Annotation statistics

Using our approach, we created two corpora that can be directly compared to each other: (1) A Wikipedia-based corpus including all articles that typically start with a capital letter (i.e. containing the same articles as the ones our approach extracts), with only annotated entities by editors, including **bold** mentions of the article entity in the first line. (2) The corpus based on our approach. Both are denoted as *Basic annotations* and *All annotations*, respectively.

Table 4 shows statistics we collected based on these two datasets. While containing the same number of articles, the number of annotations we found increased by $\approx 411\%$. Specifically the number of article entities (i.e. mentions of the entity an article is about, which is not linked at all in Wikipedia; denoted as *Article entity per article*) increased even further.

This supports the large potential for entity mention annotations in Wikipedia.

### 5.2. Distant supervision datasets

One of the main challenges of RE approaches is to acquire annotated datasets (entities and relations). Therefore, we show how to extract such datasets for a whole set of relations using DS.

We used DBpedia (Bizer et al., 2009), a project aiming to extract structured information from Wikipedia infoboxes as subject-relation-object triples, to extract pairs of entities (each DBpedia entity directly corresponds to a Wikipedia entity) that are linked in DBpedia. Table 5 shows statistics from the collected data. We found slightly more relevant triples than entity pairs, since it is possible for the same entity pair to participate in multiple relations.

For datasets created with DS the assumption is (see (Mintz et al., 2009)) for each entity pair and relation that is used to link the pair in the Knowledge Base, at least one (ideally more) of the extracted sentences containing this pair expresses the relation of interest. Therefore, the more unique entity pairs with matching sentences per relation as well as the more sentences per pair (hopefully) expressing this relation, the better. Table 6 shows how many relations, sentences and unique pairs we can extract from both corpora. Due to the large amount of annotations in the *All annotations* corpus, we can increase the number of extracted sentences containing relevant entity pairs by a factor of $\approx 3$. We can extract sentences for more relation and also the number of unique pairs per relation increases from 113 to 162.

Table 7 shows 14 relations and the number of sentences found for each relation in both corpora with matching entity pairs. For all of these relations this number can be largely increased using our *All annotations* corpus, creating datasets for RE using DS.

| | Basic annotations | All annotations |
|---|---|---|
| Relations | 5734 | 7398 |
| Sentences | 2,223,167 | 6,321,166 |
| Unique pairs (avg per relation) | 113 | 162 |

Table 6: Extraction potential from both datasets using distant supervision.

| Relation/No. sentences | Basic annotations | All annotations |
|---|---|---|
| location | 77,244 | 190,073 |
| issue | 28,243 | 84,389 |
| birthPlace | 13,336 | 82,637 |
| predecessor | 24,604 | 79,114 |
| league | 41,958 | 71,791 |
| country | 21,267 | 69,376 |
| deathPlace | 8,122 | 50,247 |
| capital | 23,900 | 44,864 |
| headquarters | 7,367 | 42,900 |
| coachTeam | 11,314 | 35,798 |
| spouse | 5,405 | 32,464 |
| founder | 11,350 | 31,156 |
| husband | 1,868 | 21,668 |
| wife | 1,815 | 20,350 |

Table 7: Selection of DBpedia relations and no. sentences found in both corpora.

## 6. Conclusion

We created an approach in order to exhaustively annotate mentions of entities in Wikipedia, resulting in a large corpus of almost 3,000,000 articles with many more annotations than the original Wikipedia dump contains. The code is publicly available and can be applied to the latest Wikipedia dump, which is free to download, by everyone. Furthermore, we showed how this can be useful for Relation Extraction datasets based on such a corpus, DBpedia and Distant Supervision. Again, many more sentences can be extracted for more relations than using a Wikipedia-based corpus without additional annotations.

So far we were only concerned with creating a corpus using the English version of Wikipedia. However, Wikipedia is available for 307 languages and although the number of articles per language varies a lot, we believe that our approach can be used for other versions as well in order to create similar corpora, especially since we are only using minimal language-dependent resources. We leave this for future work.

## 7. Bibliographical References

Akbik, A., Bergmann, T., and Vollgraf, R. (2019). Pooled contextualized embeddings for named entity recognition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 724–728.

Bird, S., Klein, E., and Loper, E. (2009). *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., and Hellmann, S. (2009). Dbpedia-a crystallization point for the web of data. *Web Semantics: science, services and agents on the world wide web*, 7(3):154–165.

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008). Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. AcM.

Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117.

Bunescu, R. and Mooney, R. (2007). Learning to extract relations from the web using minimal supervision. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 576–583.

Ghaddar, A. and Langlais, P. (2017). Winer: A wikipedia annotated corpus for named entity recognition. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 413–422.

Gupta, N., Singh, S., and Roth, D. (2017). Entity linking via joint encoding of types, descriptions, and context. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2681–2690.

Kiss, T. and Strunk, J. (2006). Unsupervised multilingual sentence boundary detection. *Computational Linguistics*, 32(4):485–525.

Klang, M. and Nugues, P. (2018). Linking, searching, and visualizing entities in wikipedia. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. (2016). Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Lee, K., He, L., and Zettlemoyer, L. (2018). Higher-order coreference resolution with coarse-to-fine inference. *arXiv preprint arXiv:1804.05392*.

Logan, R., Liu, N. F., Peters, M. E., Gardner, M., and Singh, S. (2019). Barack's wife hillary: Using knowledge graphs for fact-aware language modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5962–5971.

Mahdisoltani, F., Biega, J., and Suchanek, F. M. (2013). Yago3: A knowledge base from multilingual wikipedias.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., and McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Mintz, M., Bills, S., Snow, R., and Jurafsky, D. (2009). Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics.

Nothman, J., Curran, J. R., and Murphy, T. (2008). Transforming wikipedia into named entity training data. In *Proceedings of the Australasian Language Technology Association Workshop 2008*, pages 124–132.

Riedel, S., Yao, L., and McCallum, A. (2010). Modeling relations and their mentions without labeled text. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 148–163. Springer.

Sang, E. F. and De Meulder, F. (2003). Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.

Sil, A., Kundu, G., Florian, R., and Hamza, W. (2018). Neural cross-lingual entity linking. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Takanobu, R., Zhang, T., Liu, J., and Huang, M. (2019). A hierarchical framework for relation extraction with reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7072–7079.

Yih, S. W.-t., Chang, M.-W., He, X., and Gao, J. (2015). Semantic parsing via staged query graph generation: Question answering with knowledge base.