Open Entity

Extraction, Classification and Matching of Entities from German Websites

Levin von Hollen
Marius Leka
Fabian Steputat
Daniel Niecke
Björn Werner
Tjark Krause
Oke Nissen

ABSTRACT

The web is a huge source of information. Organisations such as Google and Microsoft have developed systems for creating knowledge graphs, based on the information found on the web. Web content is crawled and the information is extracted and is classified into entities. However, their solutions are considered black boxes, which leads to a lack of transparency and accountability. Open Entity (OpEn) proposes a concept for a collaborative system that is designed as a whitebox instead and is therefore completely transparent about the methods, models and data it contains. For the extraction and classification, where the focus was on German organisations, traditional methods like SVM, CNN and RNN are investigated about their performance, as well as, newer methods like neural network. The evaluation has shown that traditional SVM can compete with neural networks and outperform other traditional methods such as CNN and RNN. Additionally, a method is presented that solves the name disambiguation on diagram data in an acceptable time.

1 INTRODUCTION

The web contains a large amount of information about entities (people, places, organisations, etc.). There is a trend to extract the attributes and relationships of entities to create knowledge graphs, for example Google's Knowledge Graph[1] or Microsoft's Bing Satori[2]. The sources of this information are publicly accessible sites such as LinkedIn, Wikipedia or Facebook [61]. This work investigates problems of current solutions and tries to improve them. The focus is on the creation of a knowledge graph for German organisations. For this purpose, state-of-the-art classifiers and extraction algorithms are evaluated and compared.

In addition, ethical considerations [12] and thus transparency and accountability seem to be an issue with existing solutions [62] (e.g. Google Knowledge Graph) when considering current trends.

To tackle the lack of transparency, we propose a collaborative system, that combines the extraction of information from websites and user feedback and allows central access to the knowledge base. Unlike existing solutions, it is transparent and accountable. Providing transparency and accountability is not simply done by providing access to the raw data since this is not enough to understand the data [30].

Building such a system creates multiple problems. First, we do not know which methods are used by existing solutions, since those can be considered as blackboxes [61]. This means that specifically for the extraction and classification of entities on the web, we need to find the best methods. Here arises another problem as a qualitative gold standard is required for these methods to train and evaluate them.

Another problem with the classification and extraction of entities is the identification of possible duplicates [28]. This is because it cannot be assumed that an entity can only be found on a single website. It also occurs that crawled and classified entities do not have enough describing attributes combined with very generic names. These entities must disappear from the database.

The heavy lifting of the information generation should be semi-automatic, to ensure a steady gain of information and reduce manual effort and maintenance. Additional sources like OpenStreetMap[3] or users should be incorporated to enhance the information. Therefore, the challenge is to provide a collaborative system that provides a structure that allows objective information selection from different sources while preventing abuse and corruption of the system.

In conclusion our challenges are the following:

(1) Creating a knowledge system that achieves transparency and accountability
(2) Creation of a gold standard for the extraction and classification methods.
(3) Extraction of entities from the web from multiple sources
(4) Classification of entities from the web with sparse information
(5) Matching of entities with erroneous information

To build such a system, we must meet all the challenges mentioned above. However, this paper focuses on the research questions arising from the challenges of entity classification, extraction and matching:

- Entity Classification and Extraction
  - Research Question 1.1: Are Neural Networks or Support Vector Machines more suitable for our entity extraction purposes?
  - Research Question 1.2: Do HTML-informations encoded in the text have a positive effect on the entity extraction results?

[1] https://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html
[2] https://blogs.bing.com/search/2013/03/21/understand-your-world-with-bing/
[3] https://www.openstreetmap.de/

– Research Question 1.3: Does the sentence length have an impact on the quality of the entity extraction results?

1.1 Approach

This section describes which approaches are used to address the previously described challenges. In addition, a broad overview is given for how transparency and accountability can be achieved in a knowledge system. An approach to the above described challenges is automatic extraction.

We focus on organisations and their attributes, like name, address, and opening hours. Additionally, we focus on German organisations and their websites. This makes the extraction process more complicated because the German language has some peculiarities, like more complex grammar in comparison to English.

To ensure that each organisation is represented by only one entity in the knowledge base, the data store needs to be checked for duplicates regularly and those have to be merged or purged.

In addition, in order to achieve a level of transparency and accountability, which is useful for the user, a system should not just provide the raw data. In fact, the raw data has to be extended with other information and explanations about the data to make it understandable to outside actors. This data is called provenance[4], which includes activities, entities and agents involved with the data. Every meaningful interaction with the data is documented in a lightweight way but should still be meaningful enough to be comprehensible. The system is designed as a whitebox in contrast to existing blackbox solutions, this means that the algorithms and data models are fully transparent.

Our proposed Open Entity (OpEn) concept for a knowledge system consists of four recurring stages that run in a continuous cycle, as shown in Figure 1.

(1) Acquiring the Data: The data is primarily crawled from semi-structured websites on the Web. We crawl breadth-first starting from a seedlist, which was generated via using the Google Places API [24]. This is mainly discussed in the technical report Section 2 [44].
(2) Consuming the Data: Here the previously acquired raw data is consumed. This includes the extraction, classification and matching of entities. This is the main focus of our work and is discussed in Section 4 and 7.
(3) Collaborative Enriching: Here, actors such as users or software agents can extend the information or suggest changes. External sources are contacted to further improve data quality. The collaborative system is presented in the technical report.
(4) Providing the data: The collected and enriched information is published on the web, e.g. as linked open data or on request in JSON. Further details can be found in Section 3 of our technical report [44].

We achieve transparency and accountability of the system by providing full provenance. This means that OpEn tracks any change or activity to our information prepares the information for better understanding and makes all the information available to the public. This will be addressed in more detail in Section 3. The system stores all information as Linked Open Data (LOD) in a triple store, so all information can easily be reused and are available to the public.

In contrast to existing solutions, we want to do a more direct approach and directly store which data is the current one, which helps outside actors to understand the data. Also, in contrast, we want to be clear how the current valid information is chosen. In summary, we want to improve upon current solutions, with regards to usability, clarity and efficiency. To our knowledge, no system exists that has a combination of automatic extraction and classification processes with a focus on transparency and accountability.

The main contributions of our work can be summarized as follows:

(1) The structural Support Vector Machines (SVM) outperformed the BiLSTM-CNN-char-embeddings-CRF implementation (CNN). The BiLSTM-RNN-char-embeddings-CRF (RNN) came closer and even levelled with the SVM in some attributes.
(2) The results on research question two showed that the natural language methods could not comprehend the HTML structure encoded in the training data. The F1-score dropped in almost all the cases between sentences of the same length the more HTML tags were used.
(3) Large amount of RDF data can be processed by an entity matching process in acceptable time.
(4) Providing the used and created datasets.
(5) Collaborative system for knowledge graph creation which provides transparency and accountability.

2 RELATED WORK

First, related works relating to the whole OpEn system are described. Both, the aspect of cooperation and the aspect of transparency and accountability, are in the focus. Second, classification and extraction methods are discussed. Specifically, regular expressions, wrappers, support vector machines, deep learning and relation extraction are discussed in detail. Third, related work on matching and merging is described. The main areas introduced are entity matching, string similarity measures, identity in linked data and the SILK framework.

2.1 Collaborative entity extraction system

For our collaborative entity and extraction system, we took inspiration from the iKnoweb Framework [51]. Its main goal was integrating user input into the entity knowledge mining process to create qualitative data. But most of the heavy lifting should be done by the automatic processes so that only minimal user effort is needed. The system consists of three recurring stages. First, the data model is trained. Second, the entities are extracted and classified. Third, the user interacts with the data and improves it, which results in adjustments to the training data. We incorporate and extend this in regards to transparency and accountability for the Open Entity concept. In regards to the collaborative aspect, a system is needed, which encodes all collaborative actions performed on the data and allows versioning and revisioning of changes. There already exist solutions with a focus on a collaborative environment. One such system is STEVIE [9]. This system collects all contributions of the user for the attributes of entities and when a request is sent to the server, the actual entity with its attributes is calculated during the

[4]https://www.w3.org/TR/2013/NOTE-prov-overview-20130430/

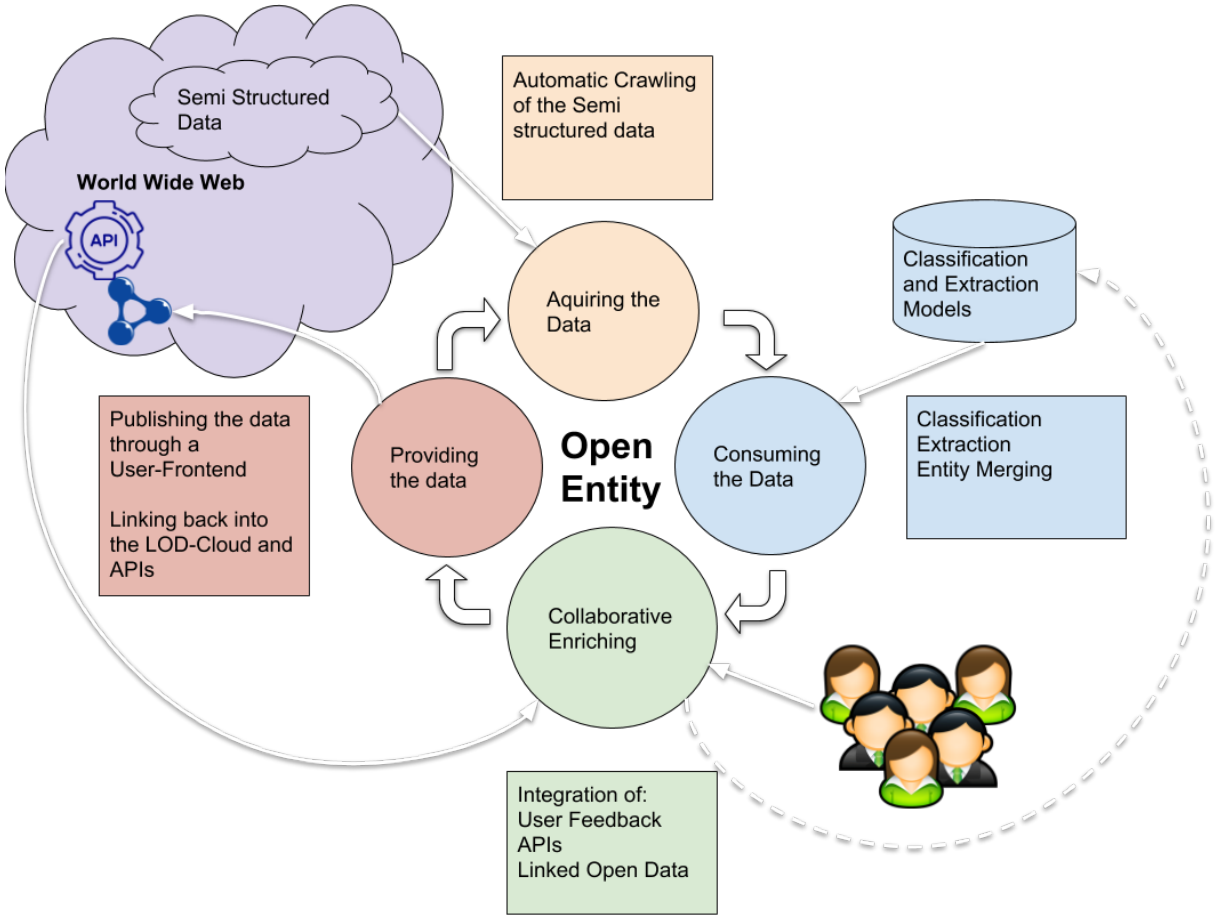

Figure 1: The Open Entity stages: (1) Acquiring data, (2) consuming the data (processing the data), (3) collaborative enriching to improve the data quality and (4) providing the data on the web

request. Therefore just by looking at the data, it is not clear which of the entities will be returned as a query result.

2.2 Classification and extraction methods

There are many approaches to overcome the difficulties in the area of Named Entity Recognition (NER) and even more in the area of document classification.

The most used techniques for Named Entity Recognition (NER) are handcrafted rules or supervised machine learning, such as Hidden Markov Models, Support Vector Machines or Conditional Random Fields, to automatically generate rules or sequence labelling algorithms [49].

The first approaches for entity extraction were handcrafted rule-based extractions with methods like Regular Expressions. They search either for pattern describing the embedding of entities in a text or rules describing the unique style of such names [54]. In the case of extracting information from webpages, these rules can also exploit the semi-structure of HTML documents, to find the desired entities. So-called Wrappers were designed to achieve this by searching along the DOM Structure [42].

Supervised learning methods, especially Hidden Markov Models, Support Vector Machines, and Conditional Random Fields are trained to capture and generalize the structure of entities embedded in a sentence. They provide sequence-based models to detect the entities in new sentences. To perform this task, a large annotated set of data is required to successfully find general rules [49]. In recent years Neural Networks were also adapted to solve sequence-based tasks like NER. They achieve a better generalization with less data [43].

The document classification describes the task of sorting texts into predefined categories based on their content. Unlike in the NER context, the data is not extracted from the text directly, instead, some latent information about the text is acquired, based on the content of the text.

Supervised Machine Learning. In the case of NER, the Support Vector Machine has to predict a structured output [39]. The prediction of a structured output is more complex than the assignment of a certain label or the prediction of real numbers as with regression tasks. An adapted version, the Structural Support Vector Machine, is used because it is more capable of suggesting sequences of data.

The drawbacks of long training times ascending super-linearly with the size of training data is moderated by new algorithms like the cutting plane method [39]. But it still remains a challenge to train large datasets like ours with limited resources.

Deep Learning. For solving NER tasks using a Neural Network it has to be able to take sequences as input and produce sequence predictions. The most popular Deep Learning sequence model is the Recurrent Neural Network (or RNN). It was improved several times e.g. with states utilizing long short-term memory (LSTM) units [29]. The NER task, also known under the term sequence labelling, further required a way of not only incorporating the past sequence elements into the prediction, but the future elements as well. Bi-directional RNNs were created to successfully tackle this demand, often in combination with LSTMs to be able to store this surrounding context [25, 58].

The impact Deep Learning had on the NLP research area, that led to various new state-of-the-art results, came from the ability to generate word embeddings using RNNs. These embeddings are trained on a big unsupervised text corpus and capture important semantic similarities between words. This meant that handcrafted word features such as gazetteers were no longer necessary. These learned embeddings can also be used for classical NLP methods and led to huge performance gains [21, 31].

To capture not only semantic similarities but also similarities in the string structure of words, character-based embeddings have been concatenated to the semantic word vectors which increased the scores once again and finally led to the beating of traditional CRF models [43, 46].

2.3 Document Classification

The best performing approaches for document classification tasks use machine learning models such as linear models like support vector machines (SVM), linear regression models, decision trees and more [34]. Lately, Deep Learning approaches outperform old state-of-the-art results, but this usually comes with the downside of significantly longer training times. Previous works [60] have shown that web page classification works very well given that a gold standard and a predefined set of categories is available.

2.4 Matching

Entity matching describes the process of identifying duplicate records in data storages.
While there are a lot of publications dealing with the problem of Entity Matching they are mostly focused on relational databases or structures that are organized like relational databases [6, 16, 28, 41]. However, publications that focus on identifying duplicates in linked data structures often suggest reusing these techniques [6, 50, 63] sometimes extended by the use of structural similarities [4]. Differences to relational databases are a reduced query speed [7, 8] and the lack of a fixed database structure [27]. While there are already Frameworks for matching linked data items that are not domain-specific [50, 63] these aim at integrating or mapping data from different sources. However, our approach requires to find duplicates in one singular dataset. This requires some adjustments to be made (see [44])

2.5 Identity in Linked Data

One option to express that two *URIs* are aliases, which means that they refer to the same real object, is to use *owl:sameAs* statements [27, 64].
According to the law of Leibniz *x = y if, and only if, x has every property which y has, and y has every property which x has* [18]. However, under the given circumstances (missing information, typographical errors, etc.) it appears appropriate to concentrate on a subset of properties to determine identity. Furthermore, it cannot be demanded that all properties are exactly the same - it is enough that they are sufficiently the same [26].
An alternative to the use of *owl:sameAs* statements for representing equality could be the use of SKOS *Mapping Properties*. These indicate an inaccurate match and offer a finer granularity of matches (exactMatch, closeMatch, broadMatch) [48]. But we don't rate the level of accuracy, so a simple marking by owl:sameAs is sufficient. The use of *owl:sameAs* statements rises a lot of potential problems [26] which are irrelevant in our application scenario because we only use the statements temporarily and merge the connected objects in the next step and do not derive any further conclusions from them. Our target model does not contain any *owl:sameAs* statements. That is why we have chosen the least complex model for temporary marking that meets our requirements: *owl:sameAs*.

3 THE OPEN ENTITY CONCEPT

The Open Entity system presented in Section 1.1 describes, how a system must be structured in order to generate information from the public sources of the Internet. However, this section deals with the aspects and principles of Open Entity. The two main aspects of Open Entity are the following:

The Open-Aspect: Open stands for full transparency and general openness of the data around the OpEn system. This means that everyone is able to re-use, improve and enrich them in a collaborative way.

The Entity-Aspect: Entity stands for a system, that combines the power of modern extraction and classification of entities from semi-structured web content and secondary sources such as wiki pages and APIs such as Google Places.

The OpEn concept was developed taking into account the ACM principles of algorithmic transparency and accountability[5]:

(1) Awareness: Awareness of possible biases in the system to everyone who is involved.
(2) Access and Redress: Provide a mechanism to enable individuals adversely affected by algorithmically informed decisions to question and redress them.
(3) Accountability: Institutions should be held responsible for the results and decisions made by algorithms.
(4) Explanation: The system should provide explanations for the results of the used algorithms.
(5) Data Provenance: Providing the full provenance as well as an exploration of biases regarding the training data.

[5] https://www.acm.org/binaries/content/assets/public\protect\discretionary{\char\hyphenchar\font}{}{}policy/2017_usacm_statement_algorithms.pdf

Requirement	Achieved?	How does the concept solve this?	What needs to be done additionally
Awareness	partially	for working on the data:"full provenance"	for users: visual cues in the frontend
Access and Redress	yes	collaborative system	-
Accountability	yes	“full provenance”	-
Explanation	yes	“full provenance” with additional explanatory meta data	-
Data Provenance	yes	“full provenance” and providing full access to models and methods and their evaluations	-
Auditability	yes		
Validation Testing	partially	evaluation within paper	automatic testing mechanisms

Table 1: Overview about how the concept achieves the set requirements ACMs principles of algorithmic transparency and accountability Item 1.

(6) Auditability: Models, algorithms, data, and decisions should be recorded so that they can be audited in cases where harm is suspected.
(7) Validation and Testing: Institutions should use rigorous methods to validate their models and document those methods and results.

The above aspects and principles are the basis of the OpEn system. This leads to an objective and transparent platform. Transparency and accountability, and the mechanics of a collaborative system are explained in more detail in the technical report [44].

The requirements for *Accountability*, *Explanation*, *Data Provenance* and *Auditability* are directly met by providing the "full provenance". Accountability is achieved because of the provenance chain stores directly who is accountable for the information or for an activity regarding the information. Therefore, related issues can be directly addressed. The provided provenance is not just raw data but is instead structured in a meaningful way. In addition, the user interface of the system visualizes the provenance to help to understand and meet the requirement for Explanation. Auditability and Data Provenance are met since the provenance chain stores the data and decisions used and the models and algorithms are public as well as their evaluations.

Awareness must also be addressed by giving the user visual clues when the data is of low quality. Access and Redress are achieved by allowing edits to the data and to the algorithms. Both are achieved via the collaborative system described in Section 5 of the technical report [44]. OpEn provides a technical report [44] in addition to this research paper, which evaluates the proposed models and methods. This needs to be further extended via automatic testing mechanisms to meet the Validation and Testing requirement. The Table 1 summarizes the evaluation of the ACM principles of algorithmic transparency within our proposed concept.

4 CLASSIFICATION AND EXTRACTION OPTIMISATION OF ENTITIES

To extract the information, the websites are also analyzed for their semi-structured content, the HTML annotations, which create a tree structure that can contain additional information, that supports the NER task. We analyze the texts with rule-based methods like Regular Expressions, classical supervised methods like Support Vector Machines or Neural Nets. Our main task is to find out which company is behind the respective website, to find names, addresses, opening hours and telephone numbers and to classify the companies into the categories of the Thesaurus of Economic Sciences from the Leibniz Information Centre for Economics [19].

In the beginning, the websites are preprocessed to fit the scheme of natural language used by the classifiers and extractors. All remaining parts of the code are filtered and encoding errors like leftover HTML-escapes are fixed. Additionally, all sites of a domain are concatenated to a large document as continuous text, if it helps the extractors find a structure over a larger text set. The preprocessing is individual for each classifier/extractor so that each classifier/extractor gets the data in the easiest digestible way for prediction ensuring better usage of the methods.

Classifying texts, even semi-structured ones as in the context of webpages works very well in most cases, but extracting information from texts, especially from semi or unstructured texts still bears a great challenge.

4.1 Classification

A useful information of an organisation to have is the category or sector that it belongs to. Since this is usually not written directly on the website, classifying the organisation based on the text content of the website seems logical. Even with semi-structured texts, the results are pretty satisfying and therefore this task was not seen as a major problem.

For supervised learning, a gold standard is necessary, where each organisation has a website and a category label. For this, we use the metadata from Google Places [45]. The Google Places metadata includes a ’types’ field with categories such as ’airport’, ’clothing store’, ’nightclub’ and many more for each organisation. These are used for training the classifier. In addition, we manually mapped those types to the STW Thesaurus for Economics [19] to put those types into a hierarchy. This is explained in more detail in the technical report [44].

Since the goal is to have a lightweight classification model or tool that can handle a lot of data and produces state-of-the-art results the popular fastText library [40] was chosen as the classification framework.

FastText [40] is an open-source library developed by Facebook [35]. FastText does not use Neural Networks but trains a linear classifier with many optimizations to enhance the prediction quality and training times, such as hierarchical softmax and bag of n-grams [40].

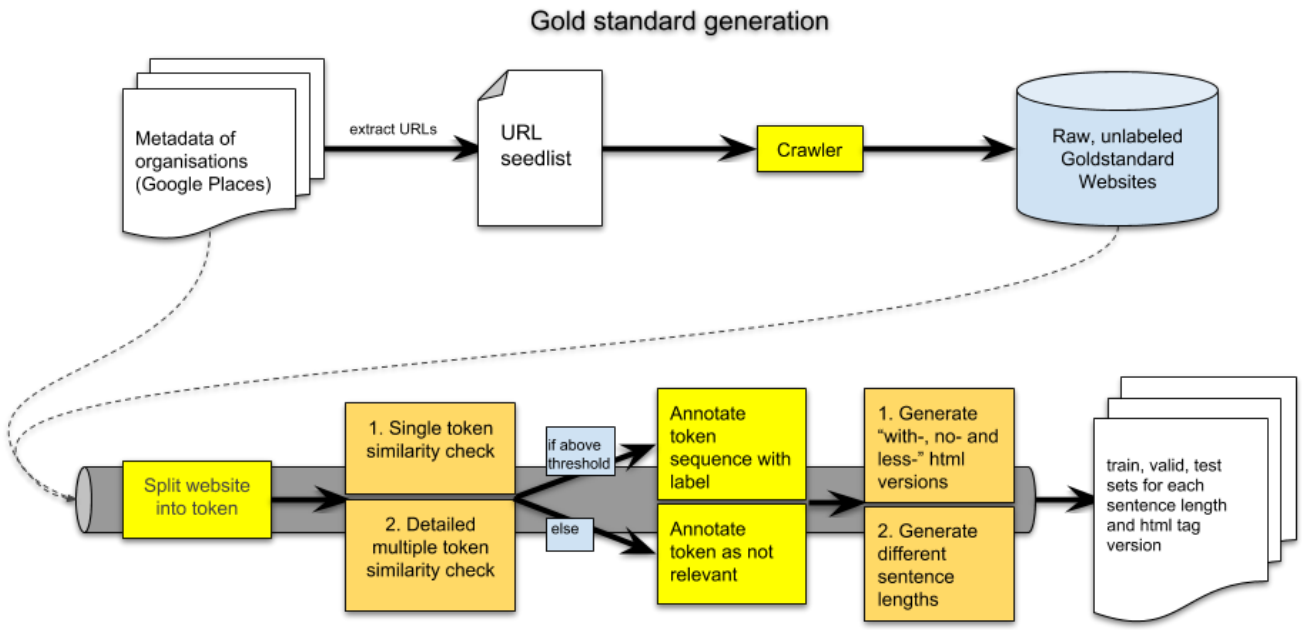

Figure 2: Overview of the automated gold standard generation process

4.2 Extraction

The most common type of information extraction is the so-called Named Entity Recognition task where the goal is to find and tag entities in a sequence of text input data [49, 59]. NER models usually learn by taking a lot of labeled sentences, usually of the same length, and then using the presence of words or of similar words using word vectors [47, 52] as well as the context, more precisely the presence of previous and subsequent tokens and predicted labels, to predict the labels for every token in an input sentence. Extracting the company's name can be more of a challenge since the company names can consist of artificial names, person names, random numbers and many more or less meaningful terms and generalize from a small number of names is usually not possible. For this type of attributes, other characteristics such as the number and location of occurrence, as well as the context of the entire page, etcetera, should be taken into account [53]

For the extraction we used three methods. A rule based approach with Regular Expressions, and two supervised machine learning approaches, a classical one with Support Vector Machines and an approach using Deep Learning.

Rule Based Extraction. is used as a baseline for the extraction of telephone numbers and e-mail addresses.

The structure of these attributes is captured in two simple Regular Expressions. We created some Regular Expressions for entities with a clear structure (like email addresses, phone numbers and opening hours). To adapt already existing patterns to the German peculiarities, like the structure of the telephone numbers or the peculiarities with the spelling (e.g. 5 numbers preselection, space or forward slash, 6 numbers), would be connected with similar expenditure as to rewrite them. Entities such as company names have no reliable indicator in German like capitalization in English. In the context of the web, the entities often lack a comprehensible context (like 'of' or words that often appear in a company context, like 'is CEO of' or 'acquires').

In an optimal case these latent patterns, which capture the essential structure an entity can be found, in would be detectable with a pattern generator of some kind. At this point the patterns have to be so sophisticated to match the latent pattern, that a human barely understands them, so we rather focused on a fully supervised learning approach. We do not implement such rule deduction systems, as they have proven themselves in many other papers [2, 10, 17, 42, 65].

Supervised Machine Learning. aims to find latent patterns with the desired attributes can be extracted. We have created a semi-automatic gold standard with training data, as described in Section 5.

The model is adjusted to the gold standard data until its predictions are as good as possible, regarding a loss function or maximal training epochs. With this training data, the machine learning algorithm tries to find a latent structure to identify data similar to the given results. To do this, they use characteristics of the results (for example, a name is uppercase) or characteristics of surrounding words (for example, phrases or words that often appear before an entity).

Structural Support Vector Machines. We used Structural Support Vector Machines as a supervised learning method to achieve that because it is a tested method for entity extraction. We use the SVM

implementation of MITIE[6], which relies on eigenword spectral word embeddings[14] and a cutting-plane training algorithm [39].

Deep Learning. A state-of-the-art Deep Learning NER model is the bidirectional LSTM with character and word embeddings. which is often combined with classical CRFs (Conditional Random Fields) in the last scoring step [32, 43, 46].

The overall architecture is pretty simple. First, the character embeddings are learned from the training data for each word using RNNs that remember information about previous steps in a sequence, so called LSTM networks. The resulting character vectors of words are then concatenated to word embeddings which usually come pretrained from a large unsupervised corpus. These word embeddings capture similarities between similar words and thus help to learn by meaning rather than by certain words. The concatenated vector is then put through another set of LSTM layers.

It is called bi-LSTM because the sequence of word vectors is not only put through the layers in the forward order, but also in the backwards order to factor the context in both directions for each word. This step will result in a probability vector for each token. Feeding a sequence of these score vectors to a traditional linear-chain CRF results in the sequence of labels with the highest probability.

For the Deep Learning Named Entity Recognition application the BiLSTM-CNN-CRF Implementation from Nils Reimers and Iryna Gurevych [56], as well as, the Sequence Tagging implementation by Guillaume Genthial [43, 46] is used.

Both are based on the described Deep Learning architecture, the implementation of Nils Reimers and Iryna additionally focuses on high training speeds. We used these model implementation mostly on default parameters except for some mini batch adjustments. In addition, German word embeddings with a dimensionality of 100 are used [55].

4.3 Ensemble

After all methods have annotated possible labels on a website or extracted found attributes, only one label or attribute per category is selected. The goal is to create an entity with the highest trust values from the extracted attributes and labels. If several classifiers/extractors have results for the same category (e.g. both found a company name) they are treated like a simple ensemble, that works with weighted bags [15]. The weights dynamically proceed from the confidence values of the classifiers and adjust with every decision. If the confidence values of different methods are not comparable they are made comparable by normalizing them before they are used to decide which result is more reliable and take that result. If there is no confidence for a classifier or extractor, it receives a fixed confidence that reflects the overall reliability of this method compared to the other methods. For instance, the Regular Expressions deliver no confidence values for the strings that matched with their pattern, but the evaluation suggests that only about 50% of the suggestions from the Regular Expression patterns for phone numbers and email addresses are correct (Table 7, Table 8). So we choose a fixed confidence value that reflects this and enables a comparison with the other extractors. So a classifier or extractor with a fixed confidence can be overruled by another classifier or extractor if the result has a high confidence value. When classifiers and extractors only produce results that are not reliable they get overruled by the ones with fixed confidence values to ensure better reliability overall.

[6]https://github.com/mit-nlp/MITIE

5 GOLD STANDARD

To successfully train NER models a valid and large gold standard is the key aspect to achieve a high rate of correct entity label predictions. Therefore, many resources must be invested in the gold standard generation process.

In the context of extracting information such as addresses and phone numbers from websites, it is known, that the structure and locations of entities vary greatly between websites. Therefore, a large training corpus is the key to learn as much variation as possible. Furthermore, the gold standard labels have to be as precise as possible. Having miss-labelled entities will worsen the prediction in the end.

A common way of generating a gold standard is by manually labelling raw texts by a group of experts. Since having a lot of experts labelling many texts is very time-consuming, crowdsourcing this process through tools like Amazon Mechanical Turk [1] is often used instead. The cutback in quality is often worth the savings in time and money.

However, in our case, it was possible to use an automatic labelling process due to the availability of metadata for the organisations. The basic process was to find the metadata value e.g. the address of the organisation on the crawled organisation webpages and mark or annotate them with the label, in this example the 'address' or 'ADR' label. An overview of the gold standard creation is given in Figure 2 below.

Public Metadata sources

For acquiring the organisation's metadata, Google Places [45] was used which provides information regarding the organisation's name, website, address, phone number, opening hours and a category, that was mainly used for the organisation webpage classification model.

Other alternatives for getting the organisation metadata included the Facebook Graph [36], YELP [37], Open Street Map [20] and the German Company Register [22]. Since Google Places provides a big and up to date collection of business and organisation data and is free to use, it was chosen to be the source for the metadata. It is also worth mentioning that the German Company Register does not provide a free API [33].

Annotation process

With the metadata available, all that's left to do is tagging the specific attribute values inside the crawled HTML pages and fed that to the NER model. Things are not that simple since an exact match is not a good solution. Addresses, for example, can have a different order of tokens, websites might abbreviate certain tokens such as streets, names and many more which would not be considered if using exact match searches and opening hours are formatted in a lot of different ways which have to be taken care of. Due to the variance of how certain attribute values, such as addresses and phone numbers, are formatted and displayed, an exact match method would not be ideal and miss many attributes. Therefore,

a better process of identifying the metadata values on websites is needed, which is described below.

First, the website and metadata values are tokenized. Then, each token from the website is compared against each metadata value token using the Levenshtein distance metric and a minimum ratio. If that minimum is achieved, a detailed similarity check is made. This is done by consecutively adding up to 25 following token and then looking at the combined similarity with the gold standard value and taking the sequence with the maximum one if it surpasses a threshold which was found to produce satisfying results. The order of token is important for attributes like name and phone number but not directly for address tokens. That is why different similarity measures are used to incorporate this.

Lastly, the labelled data has been cut to three different sentence length of 25, 200 and the full website capped at 3,000 tokens. For every sentence length there are also three different HTML tag versions. One that includes every HTML tag, the second one only includes the major HTML tags such as divs and p tags and the last version has all the HTML tag filtered out.

6 EVALUATION

The evaluation of our research questions consists of measuring the reliability of our gold standard and the performance of our classification and extraction methods.

6.1 Gold standard

We make out two steps for evaluating the gold standard.

Discovery Ratio: The Discovery Ratio describes the number of discovered attribute values on the websites divided by the total number of attributes available from Google. However, this metric bears some deceptions since it is not known if the Google gold standard attribute value matches the attribute value on the website (topicality) or whether it is present on the website in the first place. Since German websites were the focus of study you can at least assume that almost every website should have a site notice comprising a name, address and a phone number, which is required by law [13].

Ratified Ratio: The next metric used to evaluate the gold standard is the Ratified Ratio metric. It expresses whether the annotated token sequence is, in fact, the attribute type it should be. For example, an organisation could have the name 'banana' and the algorithm annotates every token called 'banana' with the company tag, even if it's not a company in the context of the sentence. Since this leads to many false annotations and might confuse the NER algorithm it should be as high as possible. Unfortunately, it is not an easy measure to compile since you need human intelligence to judge each annotation of the algorithm. Since our team is small and lacks the required resource this metric is not evaluated.

6.2 Extraction

The annotated data is split into three sets. Two of them are used for training and the last one is used for evaluation purposes. Since it is not used to train the model, there are no problems due to overfitting. Three commonly used metrics are calculated to evaluate the extraction performance: precision, recall and the F1-score.

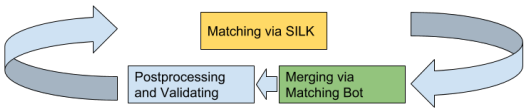

Figure 3: Process overview

Furthermore two additional parameters are to be evaluated. The first parameter describes the presence of HTML tags in the training data. Three variants of the parameter are used in order to evaluate the effect of having different levels of HTML tags in the training sets: *No HTML*, *with HTML* and *less HTML*. The *no HTML* version is completely free of HTML tags, while the *with HTML* version features all tags. The *less HTML* variant allows only divs and p tags to provide some structure information.

The second parameter controls the sentence length of the training data. Having larger sentences means the model can learn features from a bigger context, however, it might also favour overfitting and leads to longer training times since the size of training data is a lot larger. To evaluate the effect of different sentence lengths, three variants are compared to each other: Using 25 token, 200 token and using whole webpages capped at 3,000 tokens to train the models.

6.3 Classification

To evaluate the classification model, precision and recall at k are calculated using a built-in fastText method. Google provides multiple labels for organisations, thus, making it a multi-label classification task. Since not all organisations have multiple labels the P@1 and R@1 are the important scores to consider.

7 ENTITY MATCHING

To ensure that each organisation is represented by only one entity with one root node the data store needs to be checked for duplicates regularly.

While the problem of finding duplicate records is extensively reviewed [6], the problem of matching data in a linked data context is not only less reviewed but also highly depended on the used data model and certain environment variables.

To solve this problem we present an integrated and efficient matching and an adoptable merging-purging process (based on the established SILK Framework for matching). An evaluation of this process is not possible at this time, as no gold standard exists and the final source data was only available shortly before the project was completed.

All objects/organisations are stored as graphs in the Triple Store. Each entity has a unique root node. To merge all graphs that refer to the same object into one graph, the first step must be to identify all datasets that refer to the same real element [16]. Heath et al. [27] listed the most common methods for this purpose: Comparing labels using similarity metrics, Common Key Matching, Graph Matching.

However, since there are no unique identification keys and all objects are stored in the same structure, so only the labels can be compared. Since the match in one attribute is not sufficient to make

Attribute	Type	Algorithm	Weight	Required
hasName	suggested	Levenshtein	5	yes
hasName	accepted	Levenshtein	10	no
hasMainPage	suggested	Jaccard	4	no
hasMainPage	accepted	Jaccard	8	no
hasSector	suggested	Equality	2	yes
hasSector	accepted	Equality	4	no
hasAdress	suggested	Levenshtein	3	no
hasAdress	accepted	Levenshtein	6	no

Table 2: Listing of all attributes and algorithms which are used for comparison with the respective weights. Required values must exist for an object to be taken into account.

	Name	Adress	Telephone
Found	55807	46124	17711
Google	187574	187584	170292
Discovery Ratio	0.2975	0.2459	0.1040

Table 3: Gold standard statistics

		1	2	3
fastText	P@	0.6412	0.3638	0.2556
	R@	0.6412	0.7275	0.7668

Table 4: Classification results

a decision, several must be compared. The more labels per object to be queried and compared, the longer the processing of an object takes [3] (the calculation of the editing distance can have a complexity of $O(|X_1| * |X_2|)$ in the worst case(When $|X_1|$ is the length of the first string and $|X_2|$ the length of the second string [16])). The objective must, therefore, be to compare as few attributes as possible and still make a reliable decision. In other words, a trade-off between performance and quality must be waged.

It should also be considered that the attributes used for differentiation should be those that are frequently present and correct. Additionally more distinctive Attributes have to have a higher weight than less distinctive.

In addition, there is a special feature of the underlying data model that stores information about the trustworthiness of attribute values. It seems logical to put more emphasis (and therefore weight) on trustworthy attributes.

An estimate of trustworthiness is assigned to each attribute. Depending on this estimate, each attribute value is sorted into one of three categories (*accepted*, *suggested*, *stashed*). Values marked as *accepted* are the most trusted ones and values marked as *stashed* have fallen below a minimum trust level. Therefore, values marked as *stashed* are not used in the decision-making process.

We have decided to use the following attributes to make a judgement on equality: Name, MainPage, Sector, Adress.

As a previous comparison of distance metrics on census data (consisting of first name, last name, house number and street name) has shown, Levenshtein distance metric (based on average accuracy and maximum F1 score) is best suited to match these characteristics [5, 11] (we have opted for the scaled variant in order to be able to set definite thresholds). We also assume that URLs belonging to different organisations can have common elements (e.g. *mcdonalds.de/restaurant/kiel-holstenbruecke-3* and *mcdonalds.de/restaurant/kiel-gutenbergstrasse-82a* obviously do not describe the same organisation comparing them with (scaled) Levenshtein results in a similarity score of 0.69 using Jaccard results in 0.44). Therefore, we decided to use a token-based metric for the comparison.

Since the sector names are standardized, they can be tested for simple equality. These considerations resulted in the following model:

Another point to note is the problem of merging. If two objects are merged, the result may result in further matches with other objects that did not fit so far. A possible solution to this problem would be to merge the entries immediately and then compare them again with all other objects [3]. Instead, we decided on a two-phase approach in which all matches are first noted and then resolved in a second phase. This has the advantage that well-established techniques can be used for matching. Furthermore, merging, which is expensive in terms of time [7]. But merging can be carried out in parallel when all objects involved are known from the start. If the described case occurs, in which a merge results in further similarities, these are discovered and processed during the next matching iteration.

Trying to compare each object (each Organisation) to each other would result in the computation of the Cartesian product of all the objects [38] in the triple store. Hence using this naive approach for comparison is not a practical option.

Elemagermid et al. [16] suggested (among other possible options) to use *blocking* to reduce the number of required comparisons. Therefore we decided to use the *blocking*-algorithm already implemented in Silk [38]. The creators of *SILK* state that the matching has a complexity of $O(|S| + |T|)$ where $|S|$ is the number of entities in the source dataset and $|T|$ is the number of entities in the target dataset [63] or in our case $O(2 * |S|)$ (since the source is the target and vice versa).

It should be pointed out that established technologies often deal with the integration of data from different sources [50, 63]. We, on the other hand, want to compare data from one source with itself. Self-explanatory is that an object is always the same to itself, which is noted accordingly. To remove these links a renewed, one-time search of the store is required ($O(n)$). This search could be avoided by introducing the condition that the URIs should not be the same, but this is always fulfilled, except when an object is compared to itself. Therefore it seems more practical and less resource consuming to allow the self-referential links to be set and remove them later.

The basic idea behind the further processing is that the data is prepared in such a way that the processing is as efficient as possible (i.e. parallelised). The type of parallelization is determined automatically based on the distribution of the data (see technical report for details).

trained/tested with		no HTML			less HTML			with HTML		
Method	Measure	25	200	3000	25	200	3000	25	200	3000
CNN	recall	0.3333	0.2945	0.2905	0.4831	0.0338	0.0816	0.3193	0.1116	0.0170
	precision	0.0502	0.1776	0.1377	0.0404	0.1791	0.1793	0.0187	0.1729	0.1677
	F1	0.0873	0.2216	0.1868	0.0745	0.0569	0.1121	0.0353	0.1357	0.0309
RNN	recall	0.0593			0.0626			0.0456		
	precision	0.0863	-	-	0.0327	-	-	0.0241	-	-
	F1	0.0703			0.0429			0.0315		
SVM	recall	0.3920			0.4575			0.4423		
	precision	0.1754	-	-	0.1491	-	-	0.1260	-	-
	F1	0.2424			0.2250			0.1961		

Table 5: Results of organisation name extraction

trained/tested with		no HTML			less HTML			with HTML		
Method	Measure	25	200	3000	25	200	3000	25	200	3000
CNN	recall	0.4643	0.4155	0.4784	0.3444	0.1560	0.4672	0.3197	0.1763	0.1181
	precision	0.3451	0.6808	0.6463	0.2922	0.6895	0.6230	0.2557	0.6167	0.7014
	F1	0.3959	0.5160	0.5498	0.3161	0.2545	0.5340	0.2841	0.2742	0.2022
RNN	recall	0.7826			0.4825			0.0646		
	precision	0.5947	-	-	0.4751	-	-	0.2217	-	-
	F1	0.6758			0.4788			0.1001		
SVM	recall	0.6824			0.6776			0.4252		
	precision	0.6432	-	-	0.6424	-	-	0.4721	-	-
	F1	0.6622			0.6595			0.4474		

Table 6: Results of address extraction

trained/tested with		no HTML			less HTML			with HTML		
Method	Measure	25	200	3000	25	200	3000	25	200	3000
CNN	recall	0.3903	0.6773	0.6143	0.3442	0.5822	0.5526	0.3416	0.5916	0.5208
	precision	0.4321	0.7680	0.6095	0.3783	0.7964	0.7829	0.3302	0.7636	0.8389
	F1	0.4101	0.7198	0.6119	0.3604	0.6727	0.6479	0.3358	0.6667	0.6426
RegEx[17]	recall	0.3578								
	precision	0.6363	-	-	-	-	-	-	-	-
	F1	0.4580								
RNN	recall	0.5084			0.3260			0.1117		
	precision	0.7791	-	-	0.7699	-	-	0.6165	-	-
	F1	0.6153			0.4580			0.1891		
SVM	recall	0.6602			0.6355			0.5254		
	precision	0.8013	-	-	0.7689	-	-	0.7488	-	-
	F1	0.7239			0.6958			0.6175		

Table 7: Results of telephone number extraction

8 RESULTS

8.1 Gold standard

Table 3 shows the results of the gold standard evaluation. The Google Places metadata [45] contains around 187,000 companies in Germany. For 55,807 company website the algorithm found a name which is around 30%. As for the addresses it found 46,124 out of the 187,000 which also means a Discovery Ratio of 24%. While these results are satisfying, the amount of found telephone numbers is significantly lower with a Discovery Ratio of only 10%.

8.2 Extraction

The results of the evaluation for the various attributes are displayed in the following Tables 5 and 8. The methods are listed on the left side of the tables. The gold standard versions (no HTML, less HTML and with HTML) are entered at the top. These are divided into the different record lengths (25, 200, 3,000). In the inner fields are recall,

[17] RegEx was not trained, using it with HTML-tags makes the patterns useless

[18] The model had problems in many cases returning the full mail address after the @ symbol

trained/tested with		no HTML			less HTML			with HTML		
Method	Measure	25	200	3000	25	200	3000	25	200	3000
CNN[18]	recall	0.1331	0.0218	0.0524	0.0992	0.0129	0.0234	0.0798	0.0669	0.0500
	precision	0.1481	0.0643	0.1282	0.1084	0.0528	0.0746	0.1060	0.1921	0.2431
	F1	0.1402	0.0325	0.0744	0.1036	0.0207	0.0356	0.0911	0.0993	0.0829
RegEx[17]	recall	0.3298								
	precision	0.8815	-	-	-	-	-	-	-	-
	F1	0.4800								
RNN	recall	0.9887			0.6500			0.7484		
	precision	0.9839	-	-	0.9630	-	-	0.9637	-	-
	F1	0.9863			0.7761			0.8425		
SVM	recall	0.9080			0.8846			0.6675		
	precision	0.9454	-	-	0.8969	-	-	0.9088	-	-
	F1	0.9263			0.8907			0.7697		

Table 8: Results of e-mail address extraction

precision and F1-measure, which the corresponding methods have achieved on the respective gold standard set.

Each row presents the results for the given method containing the numbers for recall, precision and the F1-score. CNN is referring to the BiLSTM-CNN-CRF, RNN is short for the BiLSTM-RNN with char embeddings and SVM refers to the MITIE framework which builds on traditional structural SVMs.

First, we answer the RQ1 how Neural Nets compare to traditional methods and which method performs the best. All score comparison should be read as absolute differences. The F1-score of the SVM approach is in all cases, using sentence lengths of 25, higher than the BiLSTM-CNN-CRF. Looking at the F1-scores for predicting addresses, you can see that the results of the SVM is about 26% higher than the CNN. The RNN, on the other hand, sometimes performs similar to the SVM and in some cases produces higher F1-scores as seen with the e-mail attribute. For example, it beats the SVM by about 6%. However, in other situations, such as predicting names, it performs worse than both the CNN and SVM. For evaluating phone numbers and email addresses a simple RegEx is used for comparison purposes. In both cases, it performs worse than the RNN and the SVM but better than the CNN.

Continuing with RQ2 if the presence of HTML tags enhances prediction quality the following comparisons are of interest. When predicting names, the F1-score decreased with increasing presence of HTML tags for each sentence length. The F1-scores of the SVM results with a sentence length of 25 declines from 24% without HTML tags over 22% using a few HTML tags to 19% with all HTML tags. The same holds true for the CNN (9%, 5%, 3%) and every other attribute at a sentence length of 25.

Regarding RQ3 how different sentence lengths affect prediction quality: Due to the available resources, the comparison is only based on results from the CNN. No definite assertion can be made since there are conflicting results. For the attribute name, we see the best results with a sentence length of 200 (no HTML, with HTML) and 3,000 (less HTML). For the address, the best results are with a sentence length of 3,000 (no HTML, less HTML) and 25 (with HTML). In some cases, an increase in sentence length improves the results, e.g. for phone numbers with HTML, the prediction improves by 30% with increasing sentence length. In other cases, the prediction for the address deteriorates by 8%.

8.3 Classification

The classification results are shown in Table 4. For single-label classification, the precision and recall are both at 64%. With having to predict more labels, the precision drops to 36% for two labels and 26% for predicting three labels, while the recall increases slightly.

9 DISCUSSION

For the extraction task, the results open up the following perspectives on our research questions. Regarding the first research question we found, that the SVM outperformed the CNN, while the RNN had problems with company name prediction, but outperformed SVM at mail and address prediction. The results show that traditional NER methods such as SVMs can compete with Neural Networks. But Neural Networks with more training resources can achieve even better results under otherwise same conditions. One major drawback of the SVM and RNN is the long training time with the resources at hand. However, the CNN trained multiple magnitudes faster, for example, it only took 24 minutes to train the model on a dataset whereas the MITIE framework required a whole day to complete the training process. Of course, the performance of the CNN lags behind its competitors. In the technical report [44], we provide the results for a training session using a six times larger gold standard to train the CNN model. The SVM was not trainable with so much data under the same circumstances. As expected the results improve with more data and taking the best all over result for each HTML parameter the CNN can compete with the SVM, trained on the smaller training set, requiring still slightly less training time.

Regarding the second research question, the resulting data shows that this is not the case. The scores of the tested extraction methods indicated a negative correlation between the number of HTML tags encoded into the training data and the quality of prediction. We assume that it is not possible to transfer the semi-structural information from the HTML structure into the training data since there is no latent pattern inside the HTML structure that could be

exploited by the NLP methods. The HTML tokens just added noise which made it more difficult for the extractors to recognize the valuable natural language patterns.

Regarding the third research question, there is no clear answer. We expected that with longer sentences the results get better since it provides a wider context to the NER method, especially when HTML tags are present because they consume a lot of space to unfold their structure. But neither do the results consistently show that longer sentences are advantageous nor that they are disadvantageous.

For the document classification task, the results are sufficient to provide an initial categorization of new company websites.

10 CONCLUSION

In Conclusion, this paper proposes a concept of how to design a system which meets the ACMs requirements for transparency and accountability. Most of the requirements were directly met by providing full provenance.

In addition, this paper compared multiple approaches for extracting and classifying entities from German websites and found that Deep Learning approaches still have problems beating classical methods like SVMs without proper fine-tuning. SVMs, however, were able to outperform other traditional methods like CNN or RNN by up to 20%. Furthermore, we can conclude that the semi-structure that lies upon the HTML notation does not provide beneficial information that supports the extraction task, but rather irritating noise. Training with different sentence lengths did not provide any useful insight since the results showed no signs of clear patterns.

For the matching of entities within the linked data graph structure RDF, this paper presents a prototypical implementation that shows that matching and merging on our data model is possible in an acceptable timeframe and with adequate resources consumption.

11 FUTURE WORK

To evaluate the entire system, it must first be set up completely. We have already implemented a prototype that addresses stage 1 to 3 of the concept as Figure1 shows. However, the concept is designed for collaboration, i.e. a user basis must be created after the system has been set up. Therefore, in addition to the technical aspects, many user-related problems must also be clarified. After a certain period of time, the entire project can be evaluated. The goal is to achieve a system similar to Linux in which the community improves the system together. This should establish long-term confidence in the system.

The following sections describe how the aspects described in this paper can be improved in the future. We focus on improvements to the gold standard, the extraction, the classification and the merging.

11.1 Gold standard

The performance of the extraction process is limited by the quality of the gold standard. Thus, in the future with more resources available a lot of improvements can be applied.

The first step would be to tweak and optimize the parameters of the gold standard creation process, namely the single and multi-string similarity thresholds.

While automatic annotation is fast and creates a big gold standard, doing it manually almost always provides a higher quality. Since it requires too many resources an alternative would be to inspect, verify and make corrections on the computer-produced annotations using semi-automatic labelling software such as BRAT [57]. Having an audited gold standard also means that you can now calculate the Ratified Ratio metric which allows for better evaluation of parameter adjustings.

Extending the range of annotated attributes is a good an idea as well. Possible candidates could be managers or employees, opening hours, events and relations. Sources for these attributes are available on platforms such as the Facebook Graph API [36] that include opening hours and events of organisations.

Furthermore, it could help to include downloadable contents from the webpages, such as PDFs and other text files, to the annotation process to achieve an even higher information foundation.

11.2 Extraction

To improve the comparison and evaluation of the NER methods, all methods should be evaluated and trained on the full gold standard with every sentence length and every HTML tag version. Including more methods and improving the method parameters would also be beneficial to the significance of the comparison.

This paper compared off-the-shelf NER methods trained and tested on crawled, mostly unfiltered websites. For the future, it is required to adjust these existing models to better work with semi-structured website text and even build entirely new methods specialized to work on these kinds of texts. The preprocessing on the websites can also be improved, i.e. by filtering repetitive top, bottom, and sidebars, removing ads and other non-related content. This way the models are fed with the main content that does not include noisy texts. Incorporating the layout and position of content boxes into the algorithm using the DOM tree or screenshots of the webpage might also help it to gain further spatial and contextual knowledge to improve prediction capabilities as shown in some newer works [23].

11.3 Classification

The classification performance can be improved in the future by using a larger gold standard. Since our users are able to suggest and improve organisation labels these updated labels will help to improve the classification result more and more over time, thus, helping the user again.

For methodical improvements, one can think about integrating the extracted information as well as webpage images into the classification process for further experiments.

11.4 Merging

While we have shown how a matching and merging process could look like it is necessary in the hereafter, in order to carry out a complete and quantitative evaluation, to establish a gold standard. Based on this, iterative improvements in the process can be initiated. The focus should be on removing redundant attributes and adjusting limits to achieve further performance improvements A fixed matching mode can also be selected as soon as a clear overview of the data exists.

REFERENCES

[1] Inc. Amazon Mechanical Turk. 2018. Human intelligence through an API. https://www.mturk.com/. Accessed: 2018-05-14.

[2] Alberto Bartoli, Andrea De Lorenzo, Eric Medvet, and Fabiano Tarlao. 2018. Active learning of regular expressions for entity extraction. *IEEE transactions on cybernetics* 48, 3 (2018), 1067–1080.

[3] Omar Benjelloun, Hector Garcia-Molina, David Menestrina, Qi Su, Steven Euijong Whang, and Jennifer Widom. 2009. Swoosh: a generic approach to entity resolution. *The VLDB JournalThe International Journal on Very Large Data Bases* 18, 1 (2009), 255–276.

[4] Indrajit Bhattacharya and Lise Getoor. 2006. Entity resolution in graphs. *Mining graph data* (2006), 311.

[5] Mikhail Bilenko, Raymond Mooney, William Cohen, Pradeep Ravikumar, and Stephen Fienberg. 2003. Adaptive name matching in information integration. *IEEE Intelligent Systems* 18, 5 (2003), 16–23.

[6] Christian Bizer, Tom Heath, and Tim Berners-Lee. 2009. Linked data-the story so far. *International journal on semantic web and information systems* 5, 3 (2009), 1–22.

[7] Christian Bizer and Andreas Schultz. 2009. The berlin sparql benchmark.

[8] Mihaela A Bornea, Julian Dolby, Anastasios Kementsietsidis, Kavitha Srinivas, Patrick Dantressangle, Octavian Udrea, and Bishwaranjan Bhattacharjee. 2013. Building an efficient RDF store over a relational database. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*. ACM, 121–132.

[9] Max Braun, Daniel Schmeiß, Ansgar Scherp, and Steffen Staab. 2010. Stevie-collaborative Creation and Exchange of Events and Pois on a Mobile Phone. In *Proceedings of the 2Nd ACM International Workshop on Events in Multimedia (EiMM '10)*. ACM, New York, NY, USA, 35–40. https://doi.org/10.1145/1877937.1877948

[10] Chia-Hui Chang and Shao-Chen Lui. 2001. IEPAD: information extraction based on pattern discovery. In *Proceedings of the 10th international conference on World Wide Web*. ACM, 681–688.

[11] William Cohen, Pradeep Ravikumar, and Stephen Fienberg. 2003. A comparison of string metrics for matching names and records. In *Kdd workshop on data cleaning and object consolidation*, Vol. 3. 73–78.

[12] Paul B. de Laat. 2017. Big data and algorithmic decision-making: can transparency restore accountability? *SIGCAS Computers and Society* 47, 3 (2017), 39–53. https://doi.org/10.1145/3144592.3144597

[13] Bundesministerium der Justiz und für Verbraucherschutz. 2016. Leitfaden zur Impressumspflicht. http://www.bmjv.de/DE/Verbraucherportal/DigitalesTelekommunikation/Impressumspflicht/Impressumspflicht_node.html. Accessed: 2018-07-08.

[14] Paramveer S Dhillon, Dean P Foster, and Lyle H Ungar. 2015. Eigenwords: Spectral word embeddings. *The Journal of Machine Learning Research* 16, 1 (2015), 3035–3078.

[15] Thomas G Dietterich. 2000. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*. Springer, 1–15.

[16] Ahmed K Elmagarmid, Panagiotis G Ipeirotis, and Vassilios S Verykios. 2007. Duplicate record detection: A survey. *IEEE Transactions on knowledge and data engineering* 19, 1 (2007), 1–16.

[17] Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S Weld, and Alexander Yates. 2005. Unsupervised named-entity extraction from the web: An experimental study. *Artificial intelligence* 165, 1 (2005), 91–134.

[18] Fred Feldman. 1970. Leibniz and" Leibniz'Law". *The Philosophical Review* (1970), 510–522.

[19] ZBW Leibniz Information Centre for Economics. 2017. STW Thesaurus for Economics. http://zbw.eu/stw/version/latest/about. Accessed: 2018-05-14.

[20] OpenStreetMap Foundation. 2018. OpenStreetMap API. https://wiki.openstreetmap.org/wiki/API. Accessed: 2018-05-14.

[21] Lihao Ge and Teng-Sheng Moh. 2017. Improving text classification with word embedding. In *Big Data (Big Data), 2017 IEEE International Conference on*. IEEE, 1796–1805.

[22] Bundesanzeiger Verlag GmbH. 2018. Company Register. https://www.unternehmensregister.de/ureg/?submitaction=language&language=en. Accessed: 2018-05-14.

[23] Tomas Gogar, Ondej Hubáek, and Jan Sedivy. 2016. Deep Neural Networks for Web Page Information Extraction, Vol. 475. 154–163.

[24] Google. 2018. Google API. https://developers.google.com/places/?hl=de. Accessed: 2018-07-10.

[25] Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural networks : the official journal of the International Neural Network Society* 18, 5-6 (2005), 602–610. https://doi.org/10.1016/j.neunet.2005.06.042

[26] Harry Halpin, Patrick J Hayes, James P McCusker, Deborah L McGuinness, and Henry S Thompson. 2010. When owl: sameas isnt the same: An analysis of identity in linked data. In *International Semantic Web Conference*. Springer, 305–320.

[27] Tom Heath, Michael Hausenblas, Chris Bizer, Richard Cyganiak, and Olaf Hartig. 2008. How to publish linked data on the web. In *Tutorial in the 7th International Semantic Web Conference, Karlsruhe, Germany*.

[28] Mauricio A Hernández and Salvatore J Stolfo. 1998. Real-world data is dirty: Data cleansing and the merge/purge problem. *Data mining and knowledge discovery* 2, 1 (1998), 9–37.

[29] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[30] Mahmood Hosseini, Alimohammad Shahri, Keith Phalp, and Raian Ali. 2018. Four reference models for transparency requirements in information systems. *Requir. Eng.* 23, 2 (2018), 251–275. https://doi.org/10.1007/s00766-017-0265-y

[31] Jun-Ting Hsieh, Chengshu Li, and Wendi Liu. [n. d.]. Effective Word Representation for Named Entity Recognition. ([n. d.]).

[32] Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991* (2015).

[33] Jens Ihlenfeld. 2011. "Gebt die Unternehmensdaten frei!". https://www.golem.de/1107/85090.html. Accessed: 2018-07-06.

[34] Emmanouil Ikonomakis, Sotiris Kotsiantis, and V Tampakas. 2005. Text Classification Using Machine Learning Techniques. 4 (08 2005), 966–974.

[35] Facebook Inc. 2018. Facebook AI Research. https://research.fb.com/category/facebook-ai-research/. Accessed: 2018-05-17.

[36] Facebook Inc. 2018. Facebook Graph API. https://developers.facebook.com/docs/graph-api/.

[37] Yelp Inc. 2018. Yelp Fusion API. https://www.yelp.com/. Accessed: 2018-05-14.

[38] Robert Isele, Anja Jentzsch, and Christian Bizer. 2011. Efficient Multidimensional Blocking for Link Discovery without losing Recall.. In *WebDB*.

[39] Thorsten Joachims, Thomas Finley, and Chun-Nam John Yu. 2009. Cutting-plane training of structural SVMs. *Machine Learning* 77, 1 (2009), 27–59.

[40] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of Tricks for Efficient Text Classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Association for Computational Linguistics, 427–431.

[41] Hanna Köpcke and Erhard Rahm. 2010. Frameworks for entity matching: A comparison. *Data & Knowledge Engineering* 69, 2 (2010), 197–210.

[42] Nicholas Kushmerick. 2000. Wrapper induction: Efficiency and expressiveness. *Artificial intelligence* 118, 1-2 (2000), 15–68.

[43] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural Architectures for Named Entity Recognition. http://arxiv.org/pdf/1603.01360

[44] Marius Leka, Fabian Steputat, Björn Werner, Oke Nissen, Daniel Niecke, Tjark Krause, and Levin von Hollen. 2018. Open Entity: Technical Report. (2018).

[45] Google LLC. 2017. Places API Web Service - Ortsdaten. https://developers.google.com/places/web-service/details?hl=de. Accessed: 2018-05-17.

[46] Xuezhe Ma and Eduard Hovy. [n. d.]. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. http://arxiv.org/pdf/1603.01354v5

[47] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 3111–3119. http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf

[48] Alistair Miles and Sean Bechhofer. 2009. SKOS simple knowledge organization system reference. *W3C recommendation* 18 (2009), W3C.

[49] David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes* 30, 1 (2007), 3–26.

[50] Axel-Cyrille Ngonga Ngomo and Sören Auer. 2011. Limes-a time-efficient approach for large-scale link discovery on the web of data.. In *IJCAI*. 2312–2317.

[51] Zaiqing Nie, Ji-Rong Wen, and Wei-Ying Ma. 2012. Statistical Entity Extraction From the Web. *Proc. IEEE* 100, 9 (2012), 2675–2687. https://doi.org/10.1109/JPROC.2012.2191369

[52] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. 1532–1543. http://www.aclweb.org/anthology/D14-1162

[53] Thierry Poibeau and Leila Kosseim. 2001. Proper name extraction from non-journalistic texts. *Language and computers* 37 (2001), 144–157.

[54] Lisa F Rau. 1991. Extracting company names from text. In *Artificial Intelligence Applications, 1991. Proceedings., Seventh IEEE Conference on*, Vol. 1. IEEE, 29–32.

[55] Nils Reimers, Judith Eckle-Kohler, Carsten Schnober, Jungi Kim, and Iryna Gurevych. 2014. Germeval-2014: Nested named entity recognition with neural networks. (2014).

[56] Nils Reimers and Iryna Gurevych. 2017. Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Copenhagen, Denmark, 338–348. http://aclweb.org/anthology/D17-1035

[57] Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. brat: a Web-based Tool for NLP-Assisted Text Annotation. In *Proceedings of the Demonstrations Sessionat EACL 2012*. Association for Computational Linguistics, Avignon, France.

[58] Trias Thireou and Martin Reczko. 2007. Bidirectional Long Short-Term Memory Networks for predicting the subcellular localization of eukaryotic proteins. *IEEE/ACM transactions on computational biology and bioinformatics* 4, 3 (2007), 441–446. https://doi.org/10.1109/tcbb.2007.1015

[59] Erik F Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*. Association for Computational Linguistics, 142–147.

[60] Makoto Tsukada, Takashi Washio, and Hiroshi Motoda. 2001. Automatic Web-Page Classification by Using Machine Learning Methods. In *Proceedings of the First Asia-Pacific Conference on Web Intelligence: Research and Development (WI '01)*. Springer-Verlag, London, UK, UK, 303–313. http://dl.acm.org/citation.cfm?id=645960.673927

[61] Ahmet Uyar and Farouk Musa Aliyu. 2015. Evaluating search features of Google Knowledge Graph and Bing Satori: Entity types, list searches and query interfaces. *Online Information Review* 39, 2 (2015), 197–213. https://doi.org/10.1108/OIR-10-2014-0257

[62] Katrine Juel Vang. 2013. Ethics of Google's Knowledge Graph: some considerations. *J. Inf., Comm, Ethics in Society* 11, 4 (2013), 245–260. https://doi.org/10.1108/JICES-08-2013-0028

[63] Julius Volz, Christian Bizer, Martin Gaedke, and Georgi Kobilarov. 2009. Silk-A Link Discovery Framework for the Web of Data. *LDOW* 538 (2009).

[64] W3C. [n. d.]. OWL Web Ontology Language. https://www.w3.org/TR/owl-ref/#sameAs-def. Accessed: 2018-06-27.

[65] Alexander Yates, Michael Cafarella, Michele Banko, Oren Etzioni, Matthew Broadhead, and Stephen Soderland. 2007. Textrunner: open information extraction on the web. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*. Association for Computational Linguistics, 25–26.