PLOS DIGITAL HEALTH

RESEARCH ARTICLE

DrNote: An open medical annotation service

Johann Frei[1]*, **Iñaki Soto-Rey**[2‡], **Frank Kramer**[1]

1 IT-Infrastructure for Translational Medical Research, Faculty of Applied Computer Science, University of Augsburg, Augsburg, Germany, **2** Medical Data Integration Center, Institute for Digital Medicine, University Hospital Augsburg, Augsburg, Germany

‡ Clinical Partner
* johann.frei@informatik.uni-augsburg.de

Abstract

In the context of clinical trials and medical research medical text mining can provide broader insights for various research scenarios by tapping additional text data sources and extracting relevant information that is often exclusively present in unstructured fashion. Although various works for data like electronic health reports are available for English texts, only limited work on tools for non-English text resources has been published that offers immediate practicality in terms of flexibility and initial setup. We introduce DrNote, an open source text annotation service for medical text processing. Our work provides an entire annotation pipeline with its focus on a fast yet effective and easy to use software implementation. Further, the software allows its users to define a custom annotation scope by filtering only for relevant entities that should be included in its knowledge base. The approach is based on OpenTapioca and combines the publicly available datasets from WikiData and Wikipedia, and thus, performs entity linking tasks. In contrast to other related work our service can easily be built upon any language-specific Wikipedia dataset in order to be trained on a specific target language. We provide a public demo instance of our DrNote annotation service at https://drnote.misit-augsburg.de/.

Author summary

Since much highly relevant information in healthcare and clinical research is exclusively stored as unstructured text, retrieving and processing such data poses a major challenge. Novel data-driven text processing methods require large amounts of annotated data in order to exceed non data-driven methods' performance. In the medical domain, such data is not publicly available and restricted access is limited due to federal privacy regulations. We circumvent this issue by developing an annotation pipeline that works on sparse data and retrieves the training data from publicly available data sources. The fully automated pipeline can be easily adapted by third parties for custom use cases or directly applied within minutes for medical use cases. It significantly lowers the barrier for fast analysis of unstructured clinical text data in certain scenarios.

OPEN ACCESS

Citation: Frei J, Soto-Rey I, Kramer F (2022) DrNote: An open medical annotation service. PLOS Digit Health 1(8): e0000086. https://doi.org/10.1371/journal.pdig.0000086

Editor: Imon Banerjee, Emory University, UNITED STATES

Received: July 21, 2021

Accepted: July 12, 2022

Published: August 15, 2022

Peer Review History: PLOS recognizes the benefits of transparency in the peer review process; therefore, we enable the publication of all of the content of peer review and author responses alongside final, published articles. The editorial history of this article is available here: https://doi.org/10.1371/journal.pdig.0000086

Data Availability Statement: The Wikipedia and WikiData datasets are publicly available at: https://dumps.wikimedia.org/enwiki/ https://dumps.wikimedia.org/dewiki/ https://dumps.wikimedia.org/wikidatawiki/entities/ Our project repository is publicly available at: GitHub: https://github.com/frankkramer-lab/DrNote.

Funding: This work is a part of the DIFUTURE project funded by the German Ministry of Education and Research (Bundesministerium für Bildung und Forschung, BMBF) grant FKZ01ZZ1804E. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

Introduction

Effective processing of natural clinical language data has increasingly become a key element for clinical and medical data analysis. Recent trends in the field of natural language processing (NLP) have established novel data-driven neural approaches to largely improve a broad variety of language and text analysis tasks like neural machine translation, text summarization, question answering, text classification and information extraction in general. Most notably, emerging from semantic word embeddings like Word2Vec [1] and GloVe [2], contextualized word embedding techniques like ELMo [3] or BERT [4] based on the Transformer network architecture [5] are applied in order to solve most of context-specific downstream tasks. Attention-based language models therefore gained popularity among the NLP research community since they are able to outperform simpler rule-based models, statistical methods like conditional random fields and other, neural methods like LSTM-based models on core NLP tasks such as named entity recognition (NER).

On the matter of domain-specific neural approaches for NLP numerous derivatives [6–11] are applied for various NLP downstream tasks. The trend of these neural approaches appear to steer towards end-to-end models [12] which are often optimized for specific purposes [13]. While most works focus on English data, creating cross-lingual approaches [4, 14, 15] for medical applications is difficult due to the lack of sufficient data.

Traditional non-deep learning NLP systems often adopt pipeline-based approaches [16, 17] for text processing in which each pipeline stage performs a modular text processing task, enabling the reuse of single components on different applications and contexts in a simplified fashion. The core components often rely on feature-based machine learning or linguistic rule-based methods, although certain frameworks [16, 18] integrate also neural approaches for certain NLP tasks in more recent versions. For the framework of [18], a domain-specific model [19] has been published for biomedical applications for English text data. For German texts, mEx [20] implements a similar pipeline for clinical texts based on SpaCy [16], albeit its trained models have not been published.

Historically, NLP software for medical applications has been an ongoing research subject. The software system *medSynDiKATe* [21] is an early approach to extract relevant information from pathology finding reports in German language. *Apache cTAKES* [22] is another modular software for medical text processing, following the UIMA architecture, that uses OpenNLP [23] for text analysis. While [22] is mainly designed for English texts, [24] shows only moderate results for German data when using input text translation into English. *HITEx* [25] based on the *GATE* [26] framework, and *MetaMaps* [27] present comparable notable implementations for medical text processing for English text data. Provided as a public web API, *PubTator* [28] is a similar text mining tool for English biomedical text annotations with support for a fixed set of entry types.

From the perspective of commercial software for medical text analysis in German language, *Averbis Health Discovery* [29] provides an industry solution to NLP tasks for clinical applications. For a deeper insight in remaining challenges of non-English medical text processing we point to the review paper [30]. More information on the situation of clinical text analysis methods such as for medical concept extraction and normalization or for clinical challenges in general are presented in review papers [31–33]. In similar contexts, *Trove* [33] is proposed as a framework for weak supervised clinical NER tasks. While the latter work yields a broad overview on key aspects of different methodological concepts and covers weak supervised settings with ontology-based knowledge bases in English, it acknowledges the need for further work on non-English contexts.

For text annotation and entity linking in general, earlier works focus on Wikipedia and WikiData as knowledge base. Entity linking on unstructured texts to Wikipedia was shown in [34–37], even before the WikiData [38] knowledge graph was introduced. Different entity linking approaches were evaluated and compared in [39]. In addition to WikiData, other knowledge bases [40–43] have been released as well. For tagging engines like *TagMe* [37], refined entity linking systems [44, 45] were released. More recently, neural-based entity linking methods have been proposed [46–48].

Motivation

By considering common natural language processing tasks as a learning problem, this inherently implies the need for training data. Since novel Transformer-based architectures have been proven effective on large amount of domain-specific training data [6–11], training such domain-specific models for certain languages from scratch without any pretraining [7, 9, 10] remains a major challenge due to the lack of appropriate datasets in general. Hence, transfer learning approaches are commonly used for use case-specific downstream tasks and integrated in practical application [13], in order to mitigate the required amount of training data and boost the performance of the model.

Open datasets of biomedical texts and clinical letters for English languages have been published [49, 50]. In the particular case of German data resources for clinical letters, the situation is more dire [30, 51, 52] as no large dataset is publicly available.

In addition, one property of natural language processing methods concerns the possible dependency on one specific language: Although works on cross- and multilingual language models like XLM, XLM-R [14, 15] or mBERT [4] present notable results, they indicate higher downstream task performance scores for monolingual models on non low-resource languages.

Since text processing pipelines need to be manually fine tuned for their corresponding downstream task on aggregated training data in order to reach significant level of performance, these pipelines require a high level of technical skill sets in order to apply existing methods based on contextualized word embeddings in dedicated domain contexts.

Contributions

Consequently, in this work we primarily focus on methods that do not rely on techniques like contextualized word embeddings and can be built using a public dataset. From our perspective, this enables a simplified process for build, deployment and application.

This work presents an open annotation tool for unstructured medical texts which implements an entity linking solution.

Our key contributions can be considered as an ensemble of the following items:

- *Automated build process:* Our annotation tool requires precomputed annotation data in order to perform the entity linking tasks. To preprocess and obtain the annotation data, we provide a fully automated, end-to-end build pipeline that enables the user to adapt our pipeline for custom use cases.
- *Use of public data:* The annotation tool relies on the publicly available, open WikiData and Wikipedia datasets. The datasets are used for initial training of the annotation candidate classifier at build time, and as a knowledge base for entity linking later during the text annotation.
- *Language support:* Our implementation is capable of adopting other languages in its build pipeline. The user can choose a specific language from the set of supported languages in Wikipedia.

- *Usability:* The annotation service offers a simplified RESTful API for entity linking. The user can input plain text into a basic web interface. In addition, PDF documents can be uploaded and processed. In the case of using pretrained annotation data, the annotation service can be easily deployed on premise instantly.

Materials and methods

Open datasets

The need for training data is one of the major issues in the area of data-driven methods. In this work, we combine two open public data sets in order to retrieve appropriate training data.

WikiData. WikiData [38] is a free open knowledge base with multilingual, structured data. Its entities (*items*) are represented in a graph structure, in which each entity consists of an item identifier and main item label. These items can store short *description* texts, *labels* and potential *alias* labels for certain languages. In addition, an item may comprise a list of *statements* to further encode knowledge. Hereby, a statement is defined by a property and a list of corresponding values. These values can either encode explicit structured values or references to other entities in the knowledge base. Furthermore, each item can store references to other wiki entries through its *sitelink* attribute. Given by the nature of its graphical representation, the WikiData repository can be queried through a public SPARQL-API. The entire WikiData knowledge base is also accessible through a file download for local use.

Wikipedia. For the sake of simplicity, the Wikipedia platform is considered in this work as a set of independent open public language-specific wiki sites. Each wiki site is composed by a set of wiki pages. A wiki page consists of a page title and the page content. The page content is written in the *Wikitext* syntax which constitutes a simplified hypertext markup language. Plain texts from a wiki page contain words that can reference other wiki pages, and thereby form a graph-like structure consisting of one node per wiki page. Every wiki page references a corresponding item from the WikiData knowledge base. These references often expose the limitation of linking wiki pages to WikiData items due to the diverging concept scope granularity: For instance, the German wiki page for *Diabetes mellitus* links to the WikiData item *Q12206* which in reverse links back to the German *Diabetes mellitus* wiki page through the entity sitelinks. However, the WikiData item *Q3025883* represents the concept of *Type-2-Diabetes* and its sitelink back to the German wiki page resolves to the wiki page *Diabetes mellitus* with focus on the page section *Diabetes Typ 2*. This implies a potential loss of information due to the granularity mismatch since the mapping between wiki pages and WikiData items does not exhibit the bijective property.

In general, the language-specific wiki dataset can be downloaded in order to obtain the content of all wiki pages.

Text annotation

One of the decisive components that are vital to such an annotation pipeline is the tagging and linking component. The high-level task for this component is to identify all semantically corresponding entities of a given knowledge base for a given text and link the affected text positions by their related entity references. This task is regarded as an entity linking (EL) task.

$$\omega_s = \delta[s], \epsilon_s = link(\omega_s), \epsilon_s \in \Omega, \tag{1}$$

where:

ω = mention

δ = text document

s = identified text span
ϵ = (concept) entity
Ω = knowledge base (KB)
$link(m)$ = entity linking function

The objective to approximate the entity linking function $link(m)$ can be decomposed into two independent subtasks. The first step covers the mention detection and candidate generation. At this step, all possible entities ϵ from the knowledge base Ω that match to the detected mention ω are considered. The second step regards the selection of the best entity candidate for the mention ω (Entity Disambiguation). The proper design of this step heavily depends on the $\omega \leftrightarrow \epsilon$ match scoring function which may incorporate context-dependent scores in addition to context-independent similarity metrics.

In this work, we heavily rely on OpenTapioca [53] for solving the entity linking objective. OpenTapioca leverages the tagging functionality of the Apache Solr software in order to implement the candidate generation step. OpenTapioca creates and prepares a Solr collection in advance to index all relevant terms of the WikiData knowledge base for accelerated mention lookup and tagging.

For the estimation of the matching score of a mention to a corresponding entity candidate, the following local feature vector is defined and sampled for each pair of entity candidate ϵ and mention ω:

$$F(\omega, \epsilon) = (-\log\ \mathrm{p}(\omega), \log\ \mathrm{PR}(\epsilon), \mathrm{stmd}(\epsilon), \mathrm{sl}(\epsilon), 1) \qquad (2)$$

where:
$p(\omega)$ = probability of mention ω in the language model
$PR(\epsilon)$ = PageRank score of entity ϵ in KB Ω
$stmd(\epsilon)$ = number of statements of entity ϵ
$sl(\epsilon)$ = number of sitelinks of entity ϵ

Hereby, a feature matrix FM_0 is constructed from the stacked local feature vectors. The sequence of detected mentions forms a weighted graph G, where each (ω-ϵ) pair yields a node, connected to all nodes with neighboring $\omega_{neighbor}$ mentions. To represent semantic clusters, the connection weights between two nodes a, b with their entities ϵ_b and ϵ_a are modulated by the probability to reach each other or a common entity ϵ_{3rd} in the knowledge base graph Ω. The reachability is limited to first order connectivity. The feature matrix F_0 is propagated along the stochastic adjacency matrix $\tilde{M}$ for n iterations, where M represents the unnormalized (non-stochastic) adjacency matrix of the graph G, resulting in the contextualized feature tensor (FM_0, FM_1, $\cdots$, FM_n). A support vector machine (SVM) is applied to estimate the score for each entity candidate of each mention. For a deeper explanation, we point to the original paper [53].

Whereas the uni-gram language model is computed on the WikiData entity labels as an approximation, and the Page Rank scores for the entities can be computed on the WikiData knowledge base graph structure, it is important to note that the SVM classifier cannot be trained without annotated training data. Thus, the user is required to extract documents and annotate the documents' mentions with their related entity links manually.

OpenTapioca requires a tool-specific *profile* for term indexing that defines all entities which should be part of the knowledge base Ω. The text annotation only covers terms that were previously included in the OpenTapioca *profile*.

Text postprocessing and filtering

In order to allow encoding of prior knowledge about the target annotation structure, it is necessary to analyse the input text data by an NLP toolchain. We rely on the library SpaCy [16] in conjunction with its published pretrained pipeline components. The components mainly consist of a simple universal part-of-speech tagger, a more detailed part-of-speech tagger, a morpholgizer and a dependency parser.

Results

Build pipeline

An automated pipeline has been developed that performs the following steps:

Installation of required components. All required dependencies are installed to the local machine. This mainly includes a container runtime and basic shell tools.

Automated NIF extraction. We acquire the required training data through a pipeline of parsing, extraction and transformation steps of the formerly mentioned datasets. First, the Wikipedia pages for a given language code are parsed. During that step, all referencing terms in the page texts are extracted and their referenced pages are resolved. Given the OpenTapioca *profile*, we then query the SPARQL API to obtain all affected WikiData items. The entire WikiData dataset is parsed and for each WikiData item sitelink that points to its Wikipedia page, we add its item identifier to the processed Wikipedia page in the database to establish a bijective mapping while ignoring more fine-granular WikiData items.

Given the parsed and linked data for WikiData and Wikipedia, we can select all pages that contain referencing terms in the text to other pages if the related WikiData items were included by the OpenTapioca *profile*. By doing so, we treat relevant referencing terms as word annotations and, therefore, can synthesize the required dataset with annotated mentions. To avoid irrelevant text sections, only the sentences with relevant terms are further extracted. For sentence splitting, SpaCy [16] is used. The transformed data is stored in the common NIF format.

OpenTapioca annotation setup. The initialization steps for OpenTapioca are performed as follows: Based on the given OpenTapioca *profile*, all labels and alias terms of the selected WikiData items are loaded and indexed by the Solr instance. In addition, the buildup of the entity graph for the PageRank computation as well as the buildup of the uni-gram language model is run, followed by the training of the SVM classifier on the extracted NIF dataset. The logical data flow process is visualized in the box *Build Stage* of Fig 1.

The build pipeline eventually outputs a single package file that is needed for the instant deployment of the annotation service.

Service

The annotation service provides a platform for text input processing through an HTTP-based RESTful API as well as through a basic web interface. The implementation integrates the annotation strategy of OpenTapioca and the entity tagging process through an Apache Solr instance. For a successful deployment, the prebuilt file package is required which stores the classifier and language model information as well as the index database of the Solr collection for accelerated entity lookup.

In addition to the plain text annotation, the service features the processing of PDF documents with options for data input and output. Concerning the data input, the text from a PDF document is extracted in order to apply the entity linking task. Therefore, PDF documents with digital text information can be directly processed. In case of scanned documents that

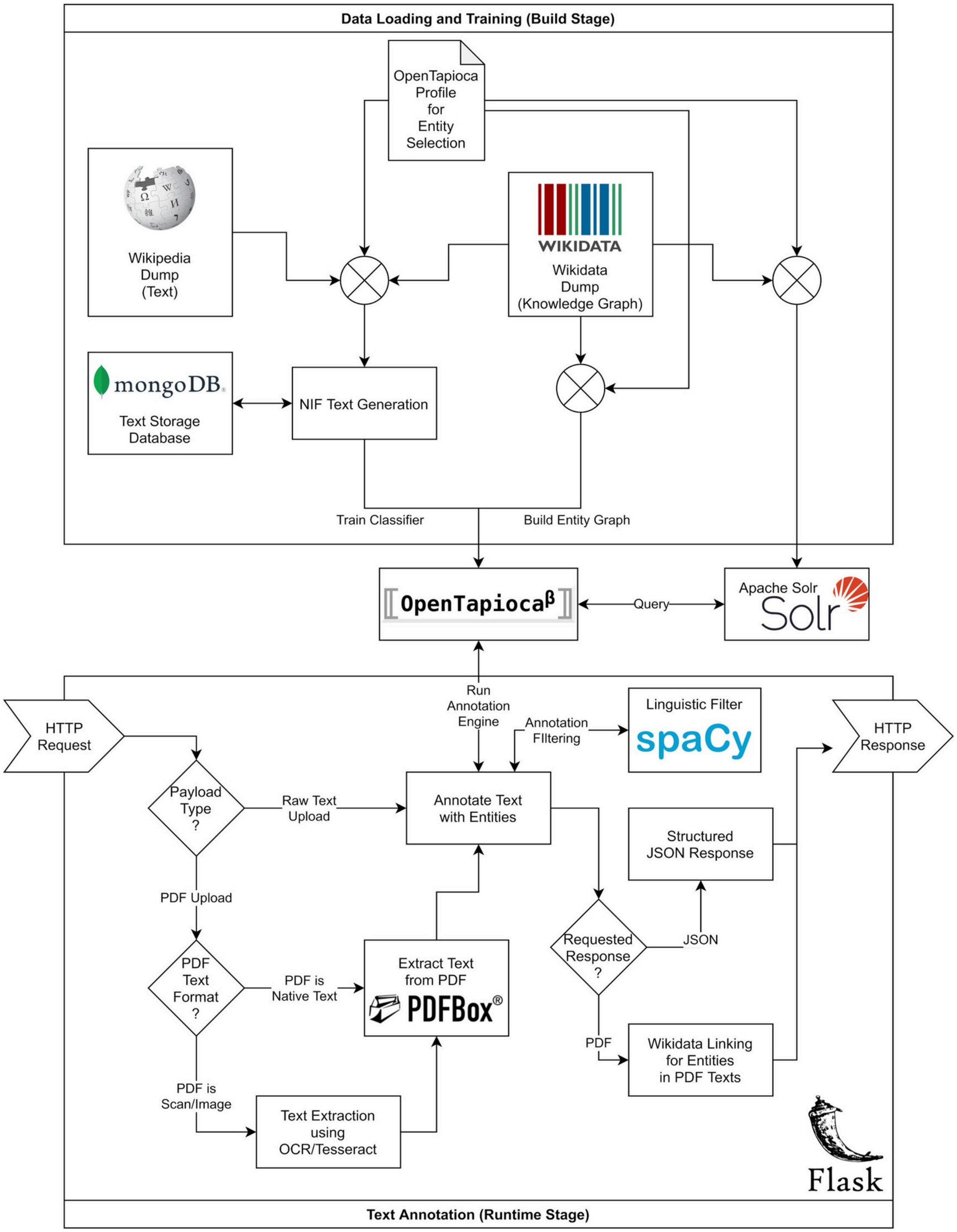


Fig 1. Visualization of logical data flow: The box "Build Stage" describes the components for data generation and preparation. The box "Runtime Stage" illustrates the processing of an annotation request. The components of OpenTapioca and Apache Solr are shared during build and runtime stage.

https://doi.org/10.1371/journal.pdig.0000086.g001 encode their information in an image format, an additional OCR step is applied. The OCR step is based on the Tesseract [54] software.

Concerning the data output, the annotation information can be either provided as a machine-readable JSON response or as a PDF document with embedded hyperlinks to the corresponding WikiData item page for all identified mentions. The service provides the option to postprocess the input text and its annotations in order to filter out annotations which are implausible based on their linguistic or structural properties. For instance, one may require the annotations to have at least one word token tagged as a noun in certain scenarios.

The logical data flow for the annotation service is depicted in the box *Runtime Stage* of Fig 1.

Build for medical use case

We apply the developed build pipeline and annotation service for our central use case for medical text analysis. The computed initialization data for the annotation service was retrieved by our build pipeline for our specified use case.

One of the most relevant information in medical letters includes data which is associated to symptoms, diagnoses, drugs and medications. Therefore, the entity selection process is managed in the way to cover all WikiData items that represent direct or indirect instances of these concepts in the knowledge base.

Our strategy to select all relevant entries leverages the graphical structure of the WikiData knowledge base. An item is a part of the knowledge base index if at least one of the following conditions is satisfied:

- The item has a *Disease Ontology* (P699) statement entry.
- The item has an *UMLS CUI* (P2892) statement entry.
- The item has a *MeSH descriptor ID* (P486) statement entry.
- The item has a *MeSH tree code* (P672) statement entry.
- The item is a subclass of *Medication* (P12140).

Since the entity linking task can only detect references to entities that have been indexed through the build pipeline, an effective feature selection contributes crucially to the capabilities of our annotation service. In our medical use case scenario, the need for multiple selection features can be demonstrated by the fact that the WikiData knowledge base can be considered incomplete. For instance, 27786 unique entities with an associated UMLS CUI statement can be found in WikiData at the time of writing. In contrast, the UMLS metathesaurus (2020AB) consists of 15938386 total entries and 4413090 unique CUIs. Adding the *MeSH descriptor ID* (P486) to the UMLS CUI selection feature increases the number of entries by 26932 and therefore can add highly relevant items to the knowledge base despite the problem of missing WikiData UMLS references.

Using multiple direct features of an item for entity selection, however, only mitigates the described issue. In addition to such direct features, we demonstrate that utilizing the hierarchical internal WikiData structure can be beneficial in order to further reduce the item miss rate due to the lack of data or incomplete data in WikiData items: The item *Medication* (Q12140) is referenced by several other items through the property *Subclass of* (P279), and thus, all items that are a subclass of *Medication* can be directly selected as relevant entries. In this context, not only first order subclass items are included but also all n-degree subclass items from the WikiData hierarchy. For instance the item *Opioid* (Q427523) is selected through the hierarchy path *Medication* $\subseteq$ *Analgesic* $\subseteq$ *Opioid* where $\subseteq$ is a *Subclass of* reference.

Table 1. Build Processing Times.

Stage	Substage	Time	Multicore	Profile
NIF & OpenTapioca	Data download	5h	no	independent
NIF	Page & redirect extraction	2h	no	independent
NIF	Entity extraction	16h	yes	independent
NIF	Entity filtering	24m	yes	independent
NIF	Pagelinks extraction	106h	yes	independent
NIF	NIF file generation	1h	no	dependent
OpenTapioca	Language model creation	16h	no	independent
OpenTapioca	Link extraction	15h	no	independent
OpenTapioca	Link sorting	45s	no	independent
OpenTapioca	Link sparse matrix conversion	24m	no	independent
OpenTapioca	Page rank computation	26m	no	independent
OpenTapioca	Entity indexing	11h	no	dependent
OpenTapioca	Classifier training	2m	partly	dependent

Build times for German medical use case: NIF-based substages interact with a MongoDB database. MongoDB supports read operations on multiple cores. Profile-dependent stages require recomputation on profile changes.

https://doi.org/10.1371/journal.pdig.0000086.t001

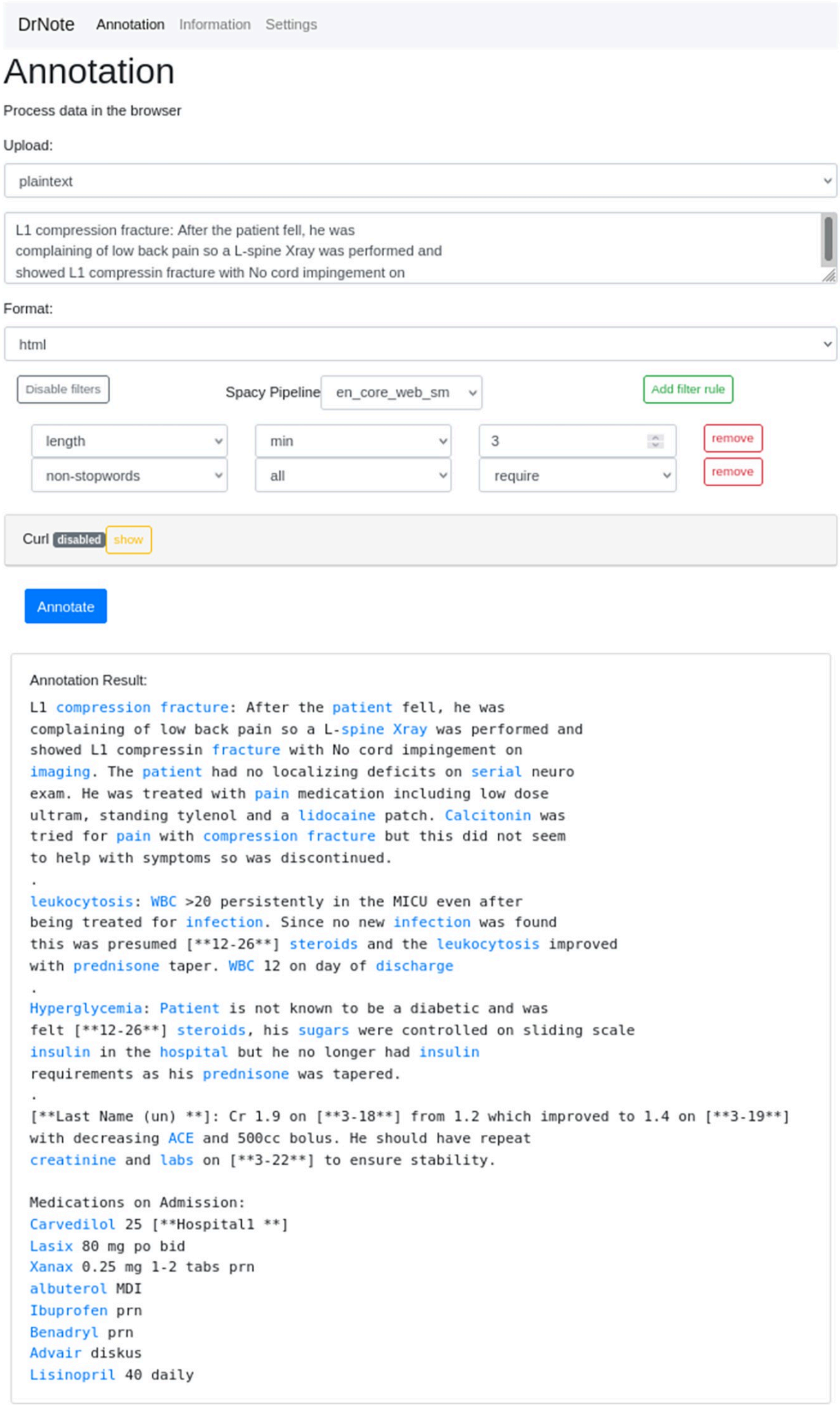
The build pipeline was executed for an German OpenTapioca profile with the formerly mentioned item selection features. The processing and training was performed on an 8-core Intel Xeon Silver 4210 virtual machine with 128GB memory. The computation times for various pipeline substages are depicted in Table 1. While later substages depend on the defined OpenTapioca profile and require recomputation on profile changes, earlier substages are independent of profile changes. Regarding multicore scaling, parts of the pipeline support the multiprocess architecture model. The computation times vary based on the number of WikiData entities and Wikipedia pages. For the presented computation times, the NIF generation stage processed 95.1M WikiData entities and 5.6M German Wikipedia pages.

As described in the previous chapter, our annotation service exposes a simple HTTP REST interface as well as a graphical web interface. The web interface is depicted in Fig 2 for an in-browser text annotation of an anonymized text snippet from the MIMIC-III [49] dataset. For demonstration purposes, we created a PDF document with the same text content and submitted the document to the PDF upload interface to retrieve an annotated PDF document with its embedded annotation links as output. The result is shown in Fig 3.

Our pretrained data for an instant service deployment as well as the source code is available at our project repository page at https://github.com/frankkramer-lab/DrNote.

Performance evaluation

To evaluate the annotation performance we compare our method with Apache cTAKES (version 4.0.0.1) and PubTator (https://www.ncbi.nlm.nih.gov/research/pubtator/api.html) as baseline. Since our method is designed for multilingual use cases and for non-English data in specific, we focus on German text data for performance comparisons. To avoid inadequate evaluation issues such as missing UMLS references or ambiguous mappings between non-isomorphic knowledge bases, we consider the annotation task as a binary text segmentation task at which the annotation spans define the binary segmentation mask. For clinical contexts, we randomly drew 50 samples from the GERNERMED [55] test set and manually corrected incorrect annotation spans, since the dataset is based on an automated translation of the *n2c2 2018*



Annotation

Process data in the browser

Upload: plaintext

L1 compression fracture: After the patient fell, he was
complaining of low back pain so a L-spine Xray was performed and
showed L1 compressin fracture with No cord impingement on

Format: html

Spacy Pipeline: en_core_web_sm

length | min | 3
non-stopwords | all | require

Annotation Result:

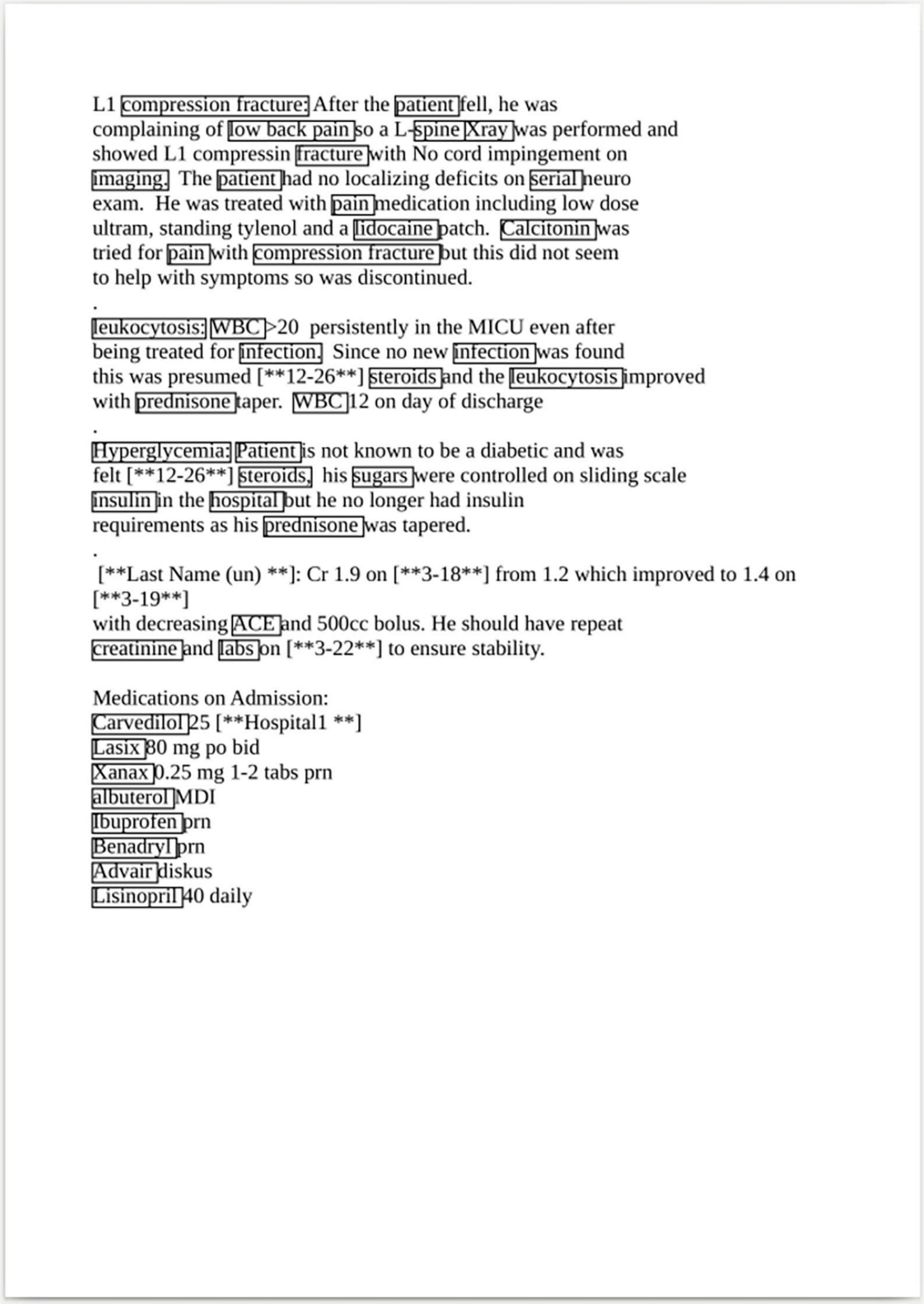
```
L1 compression fracture: After the patient fell, he was
complaining of low back pain so a L-spine Xray was performed and
showed L1 compressin fracture with No cord impingement on
imaging. The patient had no localizing deficits on serial neuro
exam. He was treated with pain medication including low dose
ultram, standing tylenol and a lidocaine patch. Calcitonin was
tried for pain with compression fracture but this did not seem
to help with symptoms so was discontinued.
.
leukocytosis: WBC >20 persistently in the MICU even after
being treated for infection. Since no new infection was found
this was presumed [**12-26**] steroids and the leukocytosis improved
with prednisone taper. WBC 12 on day of discharge
.
Hyperglycemia: Patient is not known to be a diabetic and was
felt [**12-26**] steroids, his sugars were controlled on sliding scale
insulin in the hospital but he no longer had insulin
requirements as his prednisone was tapered.
.
[**Last Name (un) **]: Cr 1.9 on [**3-18**] from 1.2 which improved to 1.4 on [**3-19**]
with decreasing ACE and 500cc bolus. He should have repeat
creatinine and labs on [**3-22**] to ensure stability.

Medications on Admission:
Carvedilol 25 [**Hospital1 **]
Lasix 80 mg po bid
Xanax 0.25 mg 1-2 tabs prn
albuterol MDI
Ibuprofen prn
Benadryl prn
Advair diskus
Lisinopril 40 daily
```

Fig 2. Demo for web interface. Browser-based annotation on example data from the MIMIC-III [49] dataset.

https://doi.org/10.1371/journal.pdig.0000086.g002

ADE and Medication Extraction Challenge [50] dataset with automated annotation alignments. All labels except for *Drug* were omitted for comparison reasons.

In order to quantify the domain-shift bias in non-clinical contexts on the biomedical *Mantra GSC* [56] datasets. In these datasets, the annotations are linked to their corresponding



L1 compression fracture: After the patient fell, he was complaining of low back pain so a L-spine Xray was performed and showed L1 compressin fracture with No cord impingement on imaging. The patient had no localizing deficits on serial neuro exam. He was treated with pain medication including low dose ultram, standing tylenol and a lidocaine patch. Calcitonin was tried for pain with compression fracture but this did not seem to help with symptoms so was discontinued.

.

leukocytosis: WBC >20 persistently in the MICU even after being treated for infection. Since no new infection was found this was presumed [**12-26**] steroids and the leukocytosis improved with prednisone taper. WBC 12 on day of discharge

.

Hyperglycemia: Patient is not known to be a diabetic and was felt [**12-26**] steroids. his sugars were controlled on sliding scale insulin in the hospital but he no longer had insulin requirements as his prednisone was tapered.

.

[**Last Name (un) **]: Cr 1.9 on [**3-18**] from 1.2 which improved to 1.4 on [**3-19**]
with decreasing ACE and 500cc bolus. He should have repeat creatinine and labs on [**3-22**] to ensure stability.

Medications on Admission:
Carvedilol 25 [**Hospital1 **]
Lasix 80 mg po bid
Xanax 0.25 mg 1-2 tabs prn
albuterol MDI
Ibuprofen prn
Benadryl prn
Advair diskus
Lisinopril 40 daily

Fig 3. Demo for PDF annotation. A PDF page demo with embedded annotations on example data from MIMIC-III [49] dataset.

https://doi.org/10.1371/journal.pdig.0000086.g003

UMLS entries. Since WikiData lacks large parts of the UMLS references mentioned (Medline: 90 out of 309 UMLS concepts known, EMEA: 121 out of 425 UMLS concepts known), the DrNote scores are also evaluated on a filtered set of UMLS annotations that are reference in WikiData, yet in all setups, DrNote annotations were limited to entities that are subclasses

Table 2. Annotation Performance Evaluation.

Dataset	Method	F1 score
GERNERMED	cTAKES	0.632
GERNERMED	DrNote	**0.722**
GERENRMED	PubTator	0.523
Medline GSC	cTAKES	0.148
Medline GSC	DrNote	**0.226**
Medline GSC	PubTator	0.123
EMEA GSC	cTAKES	0.162
EMEA GSC	DrNote	**0.261**
EMEA GSC	PubTator	0.0728
Medline GSC	DrNote (filtered)	0.414
EMEA GSC	DrNote (filtered)	0.503

Evaluation results of cTAKES, PubTator and DrNote (ours) on various datasets. Filtered results exclude annotations from the ground truth if their corresponding UMLS CUI is not referenced in WikiData.

https://doi.org/10.1371/journal.pdig.0000086.t002

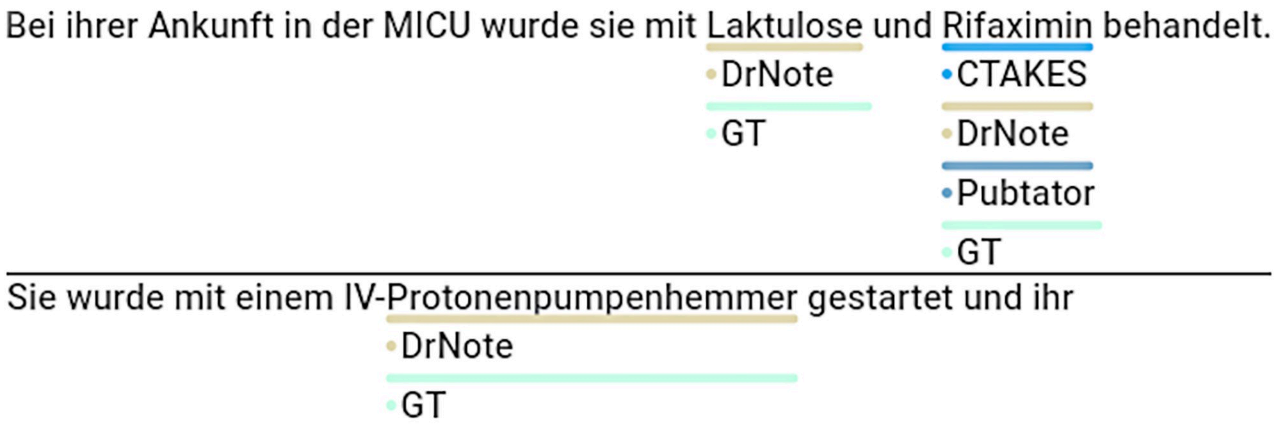
(P279) or instances of (P31) of medications (Q12140) for comparison reasons. Apache cTAKES uses the UMLS metathesaurus directly and therefore does not suffer from incomplete UMLS data. Given its focus on biomedical texts, PubTator supports the entity concepts *gene*, *disease*, *chemical*, *species*, *mutation* and *cellline*. The entity concept chemicals is identified by PubTator through a search-based dictionary lookup in the MeSH thesaurus.

The evaluation results are given as f1 scores based on the character-level text segmentation masks from the ground truth (GT) and the predicted segmentation in Table 2.

While our method exhibits substantially better text segmentation f1 score performance in comparison to cTAKES and PubTator, and demonstrates considerable results on the clinical dataset, all methods show subpar results on biomedical datasets. We mainly attribute this circumstance to the fact that both biomedical datasets include annotation phrases in part-of-speech (PoS) forms other than nouns, in which case both methods tend to fail. While cTAKES and PubTator are incapable of German word stemming due to the focus on English, our method relies on the nominalized WikiData labels and fails for similar reasons. However, our method seems to perform better on all datasets which we attribute to the broader set of common alias labels in WikiData compared to the related UMLS or MeSH entry labels as illustrated in Fig 4. In contrast to cTAKES, our method can also use linguistic information to avoid obvious PoS-related annotation errors as shown in Fig 5. cTAKES is still able to detect certain German UMLS entities due to the fact that the German language represents the largest non-English language in the UMLS metathesaurus, however PubTator uses the English MeSH database and does not include the German MeSH terms. Conversely, PubTator is able to detect specialized codes from MeSH whereas cTAKES does not detect certain codes (Fig 6) although the displayed code *RAD001* is present in the UMLS database. With respect to our method, we also identified scenarios in which correct annotations were skipped due to our filter mechanism meant to exclude non-medication items as shown in Fig 7. For this particular instance, the item *Steroid* had been annotated correctly, yet due to the structure in the WikiData graph, the item (Q177911) is not classified as being a subclass or instance of medication and skipped for that reason.

Discussion

Considering the dire state of natural language processing tools with support for multi-language data input in the medical context, the presented annotation service can offer useful services for

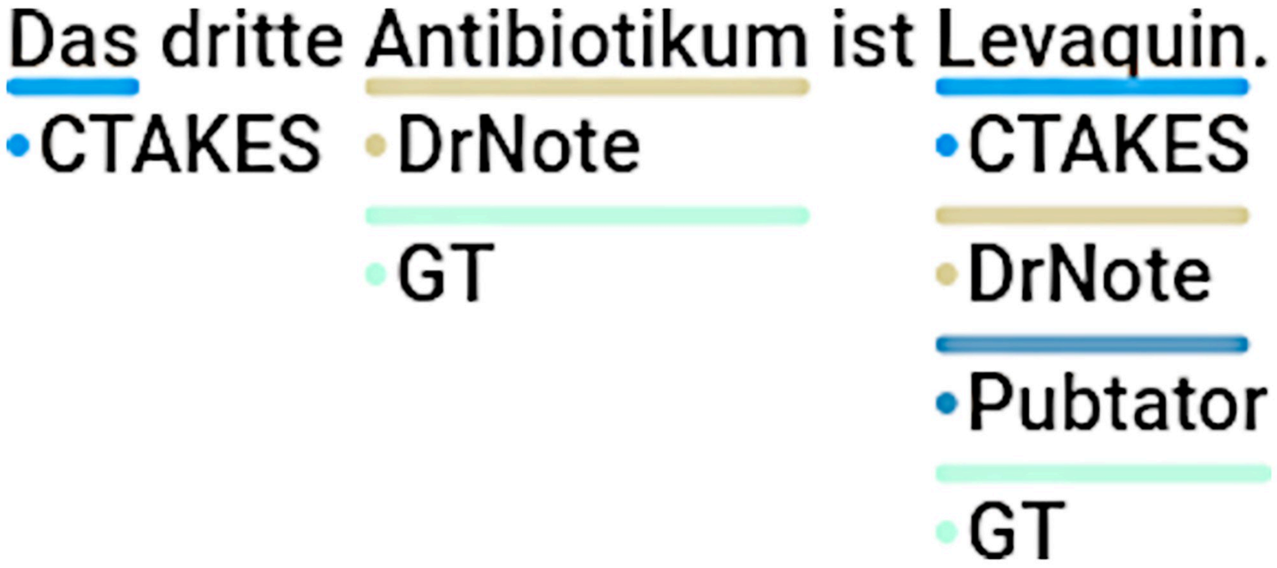


Bei ihrer Ankunft in der MICU wurde sie mit Laktulose und Rifaximin behandelt.

Sie wurde mit einem IV-Protonenpumpenhemmer gestartet und ihr

Fig 4. Comparative Sample 1 and 2. Specialized German term missing in UMLS or MeSH, (*Laktulose*,*Protonenpumpenhemmer*, cTAKES & PubTator).

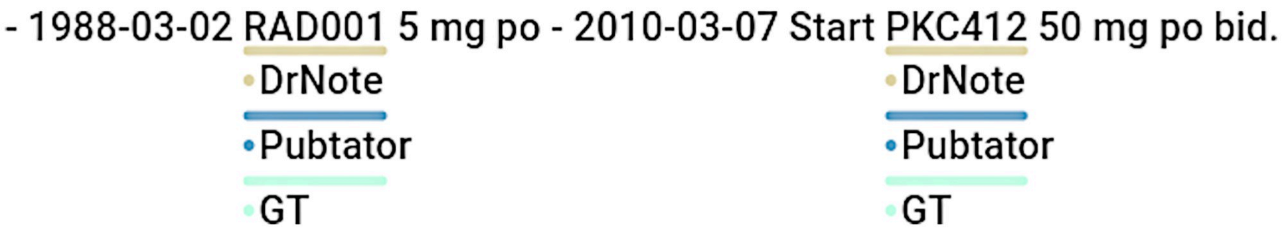
https://doi.org/10.1371/journal.pdig.0000086.g004



Das dritte Antibiotikum ist Levaquin.

Fig 5. Comparative Sample 3. Artifacts from lack of German linguistics (*Das*, cTAKES).

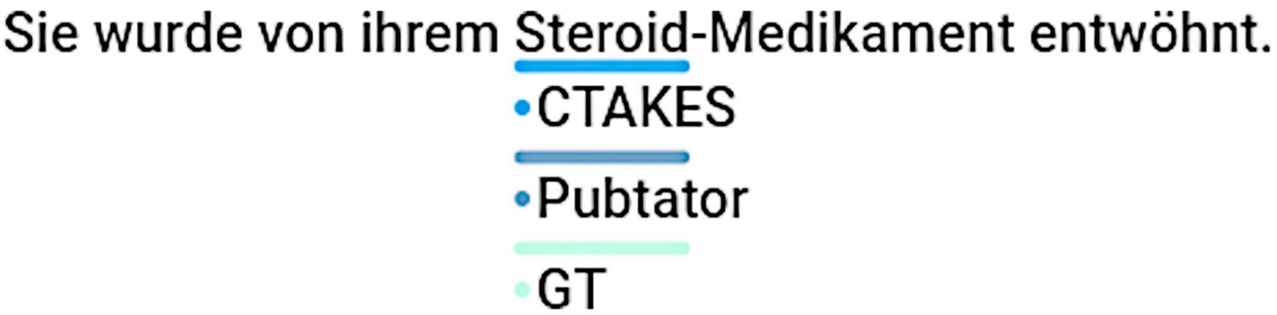
https://doi.org/10.1371/journal.pdig.0000086.g005



- 1988-03-02 RAD001 5 mg po - 2010-03-07 Start PKC412 50 mg po bid.

Fig 6. Comparative Sample 4. Weakness of cTAKES to detect certain codes (*RAD001*, cTAKES).

https://doi.org/10.1371/journal.pdig.0000086.g006



Sie wurde von ihrem Steroid-Medikament entwöhnt.

Fig 7. Comparative Sample 5. Ignored annotations due to WikiData graph structure (*Steroid* as no subclass/instance of medication), DrNote).

https://doi.org/10.1371/journal.pdig.0000086.g007

research applications and related text analysis tasks. Inherently to the chosen dictionary-based entity detection and linking approach, the capabilities of the service as well as its limitations exclude the tool for certain tasks: The entity detection method is limited to only recognize entities that are part of the WikiData knowledge base label and alias term sets, excluding semantically related terms or slightly altered, corrupted terms that are closely related to their correct term when evaluated on the Levenshtein distance metric. A potential remedy for this issue can be the use of a spellchecker component. The entity disambiguation step cannot reject false positive mentions in situations where the entity candidate presents imprecise label or alias values. For instance, the item *Universe* (Q1) contains the word *all* as an alias value. Subsequently, *all* will be linked to Q1 in the case that Q1 was previously included by the entity selection step. To effectively counter such artifacts, a deeper semantic understanding is required. However, the buildup of semantic understanding is mostly handled by utilizing large training data from certain target domains and poses a major disadvantage of data-driven methods since large datasets can be challenging to obtain and may jeopardize robust multi-language support. By offering advanced annotation filter rules based on linguistic features through SpaCy, this may alleviate the problem in situations where a pretrained SpaCy pipeline is available for the corresponding text language.

Currently, our approach is strictly tied to the label and alias terms from WikiData that are typically nominalized. Therefore, relevant terms in different part-of-speech configurations such as adjectives cannot be detected due to the lack of language-dependent stemming or lemmatization. Further work on such improvements is considered future work.

The synthesis of the annotated dataset of relevant mentions from the Wikipedia and WikiData datasets and its transformation into the NIF file format only considers links of mentions at a page to another referenced page whenever the link was inserted manually by a Wikipedia author. In frequent cases, only the first mention of a referenced concept is linked by the authors on the page, despite the fact that the mention text may appear multiple times on the same Wikipedia page. This may induce lower recall scores in contrast to a complete and manual annotation. Our mitigation approach reduces the probability of including false negative terms by only extracting single sentences from the Wikipedia page texts.

Conclusion

In this work we introduced our annotation service DrNote as an open platform for entity linking in the context of medical text processing with multi-language support. The annotation service can operate directly on precomputed initialization data which are provided for instant deployment. An fully automated build pipeline was presented to enable users to customize the annotation service for specific needs while the generated dataset solely relies on open public data. We presented the feature support for PDF document processing and annotation as well as the integration of SpaCy for advanced linguistic-based annotation filtering.

Common limitations of the chosen entity linking approach were further discussed as well as its conceptual drawback compared to competing data-driven approaches. While purely data-driven approaches may enable huge advancements over traditional approaches, their individual applicability for certain languages in the medical context remains to be challenging due to the lack of sufficiently large training data. Privacy concerns and legal restrictions for data use and access may hinder further improvements on the availability of such datasets in the future.

Author Contributions

Conceptualization: Johann Frei.

Data curation: Johann Frei.

Funding acquisition: Frank Kramer.

Investigation: Johann Frei.

Methodology: Johann Frei.

Project administration: Frank Kramer.

Resources: Frank Kramer.

Software: Johann Frei.

Supervision: Iñaki Soto-Rey.

Validation: Johann Frei.

Writing – original draft: Johann Frei.

Writing – review & editing: Iñaki Soto-Rey, Frank Kramer.

References

1. Mikolov T, Chen K, Corrado G, Dean J. Efficient Estimation of Word Representations in Vector Space. In: 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings; 2013.
2. Pennington J, Socher R, Manning CD. Glove: Global Vectors for Word Representation. In: Moschitti A, Pang B, Daelemans W, editors. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL. ACL; 2014. p. 1532–1543.
3. Peters ME, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, et al. Deep Contextualized Word Representations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). New Orleans, Louisiana: Association for Computational Linguistics; 2018. p. 2227–2237.
4. Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:181004805 [cs]. 2019;.
5. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. Advances in neural information processing systems. 2017; 30.
6. Peng Y, Yan S, Lu Z. Transfer Learning in Biomedical Natural Language Processing: An Evaluation of BERT and ELMo on Ten Benchmarking Datasets. In: Proceedings of the 18th BioNLP Workshop and Shared Task. Florence, Italy: Association for Computational Linguistics; 2019. p. 58–65.
7. Rasmy L, Xiang Y, Xie Z, Tao C, Zhi D. Med-BERT: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction. npj Digital Medicine. 2021; 4(1):1–13. https://doi.org/10.1038/s41746-021-00455-y PMID: 34017034
8. Beltagy I, Lo K, Cohan A. SciBERT: A Pretrained Language Model for Scientific Text. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International

Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Hong Kong, China: Association for Computational Linguistics; 2019. p. 3615–3620.

9. Alsentzer E, Murphy J, Boag W, Weng WH, Jindi D, Naumann T, et al. Publicly Available Clinical BERT Embeddings. In: Proceedings of the 2nd Clinical Natural Language Processing Workshop. Minneapolis, Minnesota, USA: Association for Computational Linguistics; 2019. p. 72–78.

10. Lee J, Yoon W, Kim S, Kim D, Kim S, So CH, et al. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics. 2020; 36(4):1234–1240. https://doi.org/10.1093/bioinformatics/btz682 PMID: 31501885

11. Li F, Jin Y, Liu W, Rawat BPS, Cai P, Yu H. Fine-Tuning Bidirectional Encoder Representations From Transformers (BERT)–Based Models on Large-Scale Electronic Health Record Notes: An Empirical Study. JMIR Medical Informatics. 2019; 7(3):e14830. https://doi.org/10.2196/14830 PMID: 31516126

12. Kamath U, Liu J, Whitaker J. Deep learning for NLP and speech recognition. vol. 84. Springer; 2019.

13. Liang H, Tsui BY, Ni H, Valentim CCS, Baxter SL, Liu G, et al. Evaluation and accurate diagnoses of pediatric diseases using artificial intelligence. Nature Medicine. 2019; 25(3):433–438. https://doi.org/10.1038/s41591-018-0335-9 PMID: 30742121

14. Conneau A, Lample G. Cross-lingual language model pretraining. Advances in neural information processing systems. 2019; 32.

15. Conneau A, Khandelwal K, Goyal N, Chaudhary V, Wenzek G, Guzmán F, et al. Unsupervised Cross-lingual Representation Learning at Scale. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Online: Association for Computational Linguistics; 2020. p. 8440–8451.

16. Explosion AI Gmbh. SpaCy: Industrial-Strength Natural Language Processing; 2022. https://spacy.io/.

17. Manning C, Surdeanu M, Bauer J, Finkel J, Bethard S, McClosky D. The Stanford CoreNLP Natural Language Processing Toolkit. In: Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations. Baltimore, Maryland: Association for Computational Linguistics; 2014. p. 55–60.

18. Qi P, Zhang Y, Zhang Y, Bolton J, Manning CD. Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. In: ACL; 2020.

19. Zhang Y, Zhang Y, Qi P, Manning CD, Langlotz CP. Biomedical and clinical English model packages for the Stanza Python NLP library. Journal of the American Medical Informatics Association. 2021; 28(9):1892–1899. https://doi.org/10.1093/jamia/ocab090 PMID: 34157094

20. Roller R, Alt C, Seiffe L, Wang H. mEx—An Information Extraction Platform for German Medical Text. In: Proceedings of the 11th International Conference on Semantic Web Applications and Tools for Healthcare and Life Sciences (SWAT4HCLS'2018). Semantic Web Applications and Tools for Healthcare and Life Sciences (SWAT4HCLS-2018), December 3-5, Antwerp, Belgium; 2018.

21. Hahn U, Romacker M, Schulz S. medSynDiKATe—a natural language system for the extraction of medical information from findings reports. International Journal of Medical Informatics. 2002; 67(1):63–74. https://doi.org/10.1016/S1386-5056(02)00053-9 PMID: 12460632

22. Savova GK, Masanz JJ, Ogren PV, Zheng J, Sohn S, Kipper-Schuler KC, et al. Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications. Journal of the American Medical Informatics Association: JAMIA. 2010; 17(5):507–513. https://doi.org/10.1136/jamia.2009.001560 PMID: 20819853

23. Apache Foundation. Apache OpenNLP; 2022. https://opennlp.apache.org/.

24. Becker M, Böckmann B. Extraction of UMLS concepts using Apache cTAKES for German language. In: Schreier G, Ammenwerth E, Hörbst A, editors. Health Informatics Meets eHealth. vol. 223 of Studies in Health Technology and Informatics. IOS Press; 2016. p. 71–76.

25. Zeng QT, Goryachev S, Weiss S, Sordo M, Murphy SN, Lazarus R. Extracting principal diagnosis, co-morbidity and smoking status for asthma research: evaluation of a natural language processing system. BMC Medical Informatics and Decision Making. 2006; 6(1):30. https://doi.org/10.1186/1472-6947-6-30 PMID: 16872495

26. Cunningham H, Tablan V, Roberts A, Bontcheva K. Getting More Out of Biomedical Documents with GATE's Full Lifecycle Open Source Text Analytics. PLOS Computational Biology. 2013; 9(2): e1002854. https://doi.org/10.1371/journal.pcbi.1002854 PMID: 23408875

27. Aronson AR, Lang FM. An overview of MetaMap: historical perspective and recent advances. Journal of the American Medical Informatics Association. 2010; 17(3):229–236. https://doi.org/10.1136/jamia.2009.002733 PMID: 20442139

28. Wei CH, Kao HY, Lu Z. PubTator: a web-based text mining tool for assisting biocuration. Nucleic Acids Research. 2013; 41(W1):W518–W522. https://doi.org/10.1093/nar/gkt441 PMID: 23703206

29. Averbis Gmbh. Averbis Health Discovery; 2022. https://averbis.com/de/health-discovery/.

30. Névéol A, Dalianis H, Velupillai S, Savova G, Zweigenbaum P. Clinical Natural Language Processing in languages other than English: opportunities and challenges. Journal of Biomedical Semantics. 2018; 9 (1):12. https://doi.org/10.1186/s13326-018-0179-8 PMID: 29602312

31. Percha B. Modern Clinical Text Mining: A Guide and Review. Annual Review of Biomedical Data Science. 2021; 4:165–187. https://doi.org/10.1146/annurev-biodatasci-030421-030931 PMID: 34465177

32. Fu S, Chen D, He H, Liu S, Moon S, Peterson KJ, et al. Clinical concept extraction: A methodology review. Journal of Biomedical Informatics. 2020; 109:103526. https://doi.org/10.1016/j.jbi.2020.103526 PMID: 32768446

33. Fries JA, Steinberg E, Khattar S, Fleming SL, Posada J, Callahan A, et al. Ontology-driven weak supervision for clinical entity classification in electronic health records. Nature Communications. 2021; 12 (1):2017. https://doi.org/10.1038/s41467-021-22328-4 PMID: 33795682

34. Kulkarni S, Singh A, Ramakrishnan G, Chakrabarti S. Collective annotation of Wikipedia entities in web text. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. KDD'09. New York, NY, USA: Association for Computing Machinery; 2009. p. 457–466.

35. Mihalcea R, Csomai A. Wikify! linking documents to encyclopedic knowledge. In: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management. CIKM'07. New York, NY, USA: Association for Computing Machinery; 2007. p. 233–242.

36. Milne D, Witten IH. Learning to link with wikipedia. In: Proceedings of the 17th ACM conference on Information and knowledge management. CIKM'08. New York, NY, USA: Association for Computing Machinery; 2008. p. 509–518.

37. Ferragina P, Scaiella U. TAGME: on-the-fly annotation of short text fragments (by wikipedia entities). In: Proceedings of the 19th ACM international conference on Information and knowledge management. CIKM'10. New York, NY, USA: Association for Computing Machinery; 2010. p. 1625–1628.

38. Vrandečić D, Krötzsch M. Wikidata: a free collaborative knowledgebase. Communications of the ACM. 2014; 57(10):78–85. https://doi.org/10.1145/2629489

39. Hachey B, Radford W, Nothman J, Honnibal M, Curran JR. Evaluating Entity Linking with Wikipedia. Artificial Intelligence. 2013; 194:130–150. https://doi.org/10.1016/j.artint.2012.04.005

40. Bollacker K, Evans C, Paritosh P, Sturge T, Taylor J. Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD international conference on Management of data. SIGMOD'08. New York, NY, USA: Association for Computing Machinery; 2008. p. 1247–1250.

41. Auer S, Bizer C, Kobilarov G, Lehmann J, Cyganiak R, Ives Z. DBpedia: A Nucleus for a Web of Open Data. The Semantic Web. 2007; p. 722–735. https://doi.org/10.1007/978-3-540-76298-0_52

42. Suchanek FM, Kasneci G, Weikum G. Yago: a core of semantic knowledge. In: Proceedings of the 16th international conference on World Wide Web. WWW'07. New York, NY, USA: Association for Computing Machinery; 2007. p. 697–706.

43. Hoffart J, Suchanek FM, Berberich K, Weikum G. YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. Artificial Intelligence. 2013; 194:28–61. https://doi.org/10.1016/j.artint.2012.06.001

44. Piccinno F, Ferragina P. From TagME to WAT: a new entity annotator. In: Proceedings of the first international workshop on Entity recognition & disambiguation. ERD'14. New York, NY, USA: Association for Computing Machinery; 2014. p. 55–62.

45. Ponza M, Ferragina P, Piccinno F. Swat: A system for detecting salient Wikipedia entities in texts. Computational Intelligence. 2019; 35(4):858–890. https://doi.org/10.1111/coin.12216

46. Kolitsas N, Ganea OE, Hofmann T. End-to-End Neural Entity Linking. CoRR. 2018;abs/1808.07699.

47. Kannan Ravi MP, Singh K, Mulang' IO, Shekarpour S, Hoffart J, Lehmann J. CHOLAN: A Modular Approach for Neural Entity Linking on Wikipedia and Wikidata. In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume. Online: Association for Computational Linguistics; 2021. p. 504–514.

48. van Hulst JM, Hasibi F, Dercksen K, Balog K, de Vries AP. REL: An Entity Linker Standing on the Shoulders of Giants. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. New York, NY, USA: Association for Computing Machinery; 2020. p. 2197–2200.

49. Johnson AEW, Pollard TJ, Shen L, Lehman LwH, Feng M, Ghassemi M, et al. MIMIC-III, a freely accessible critical care database. Scientific Data. 2016; 3(1):160035. https://doi.org/10.1038/sdata.2016.35 PMID: 27219127

50. Henry S, Buchan K, Filannino M, Stubbs A, Uzuner O. 2018 n2c2 shared task on adverse drug events and medication extraction in electronic health records. Journal of the American Medical Informatics Association: JAMIA. 2020; 27(1):3–12. https://doi.org/10.1093/jamia/ocz166 PMID: 31584655

51. Starlinger J, Kittner M, Blankenstein O, Leser U. How to improve information extraction from German medical records. it—Information Technology. 2017; 59(4):171–179.

52. Hellrich J, Matthies F, Faessler E, Hahn U. Sharing models and tools for processing German clinical texts. Studies in Health Technology and Informatics. 2015; 210:734–738. PMID: 25991250

53. Delpeuch A. OpenTapioca: Lightweight Entity Linking for Wikidata. In: Kaffee LA, Tifrea-Marciuska O, Simperl E, Vrandecic D, editors. Proceedings of the 1st Wikidata Workshop (Wikidata 2020) co-located with 19th International Semantic Web Conference(OPub 2020), Virtual Conference, November 2-6, 2020. vol. 2773 of CEUR Workshop Proceedings. CEUR-WS.org; 2020.

54. Kay A. Tesseract: an open-source optical character recognition engine. Linux Journal. 2007; 2007 (159):2.

55. Frei J, Kramer F. GERNERMED: An open German medical NER model. Software Impacts. 2022; 11:100212. https://doi.org/10.1016/j.simpa.2021.100212

56. Kors JA, Clematide S, Akhondi SA, van Mulligen EM, Rebholz-Schuhmann D. A multilingual gold-standard corpus for biomedical concept recognition: the Mantra GSC. Journal of the American Medical Informatics Association: JAMIA. 2015; 22(5):948–956. https://doi.org/10.1093/jamia/ocv037 PMID: 25948699