Personalized Knowledge Graphs for the Pharmaceutical Domain

Anna Lisa Gentile(✉), Daniel Gruhl, Petar Ristoski, and Steve Welch

IBM Research Almaden, San Jose, CA, USA
{annalisa.gentile,petar.ristoski}@ibm.com,
{dgruhl,welchs}@us.ibm.com

Abstract. A considerable amount of scientific and technical content is still locked behind data formats which are not machine readable, especially PDF files - and this is particularly true in the healthcare domain. While the Semantic Web has nourished the shift to more accessible formats, in business scenarios it is critical to be able to tap into this type of content, both to extract as well as embed machine readable semantic information.

We present our solution in the pharmaceutical domain and describe a fully functional pipeline to maintain up-to-date knowledge resources extracted from medication Package Inserts. We showcase how subject matter expert(s) can have their own view on the available documents, served by a personalized Knowledge Graph - or rather a view on the graph which is specific to them. We share lessons learned from our initial pilot study with a team of medical professionals. Our solution is fully integrated within the standard PDF data format and does not require the use of any external software - nor to be aware of the underlying graph.

1 Introduction

The Semantic Web community is constantly pushing the barrier on processing and producing knowledge that is understandable by both humans and machines. Nonetheless when it comes to technical and scientific content, much of the information is locked behind formats which are not directly machine readable. In the healthcare domain much information is exchanged via PDF files, especially when the communication is across different organizations or when it is directed to the public. In this work we focus on the specific task of building and maintaining consistent and updated knowledge about pharmaceutical drugs. Typically this task requires subject matter experts and it is complex enough that it takes multiple editorial units, each focusing on different aspects of the domain, to make sure that important information from such documents is extracted, categorized and retained in structured knowledge internal to the organization.

Despite the plethora of available Information Extraction (IE) tools that ease the transition from unstructured data to organized knowledge, many IE

C. Ghidini et al. (Eds.): ISWC 2019, LNCS 11779, pp. 400–417, 2019.
https://doi.org/10.1007/978-3-030-30796-7_25

approaches are based on many underlying assumptions: (i) that raw text is available, i.e. the task of obtaining such text from diverse sources (Web pages, text documents, PDF documents, etc.) is neglected; (ii) that a certain loss in accuracy is expected and tolerated; (iii) that agreement exists, i.e. there is a universal truth about what constitutes correct knowledge; (iv) that examples are ready available - or easily obtainable - to train the models. The reality is that the bootstrapping cost of having such IE tools in place is often too high, especially in business engagements with a short life span, where introducing format transformations, new tooling, new training data introduces disruptions for the end users. In our specific use case of collecting drug information, it is important to retain knowledge in the original format (i.e. the PDF document) but at the same time to be able to identify semantic content within the documents and identify relevant changes with respect to previous version of the same document. This aspect is often neglected by Knowledge Graph construction approaches, where the end product is the populated graph itself rather than the graph in combination with the enriched source documents.

We propose a strategy to perform such IE tasks directly on the input documents, so as to be completely transparent for the end user. As our focus is on PDF documents, we add task-specific semantic annotators directly into the PDF files (which are thus viewable with a standard PDF reader). We offer a combination of ontology based annotators as well as the possibility to add new semantic annotators trained on demand, within the PDF itself, with a human-in-the-loop methodology. Transparently to the end user we collect all information from all versions of the documents in a consolidated Knowledge Graph, which is used to keep track of information changes about each drug.

The novelty of this work is that we perform semantic enrichment directly into PDF files. This is not simply a technical contribution but a methodological one, because to build any human-in-the-loop system, the interaction needs to be continuous and non-disruptive for the subject matter expert. In the proposed use case we let the user continue their knowledge curation task exactly as they were used to - by manually reading and analyzing documents - but we enrich the same exact documents with highlights and comments - which are the result of semantic enrichment. The Knowledge Graph in the backstage is the result of the annotations obtained from standard ontology-based annotators, as well as those obtained by any new annotators that the user wants to train on the fly - we will concretely show how we obtain "salient sentences", as defined and trained by the editorial unit curating Adverse Drug Reaction, and Drug-Drug interaction relations. Lastly we can maintain a *personalized* Knowledge Graph for each editorial unit, where only the results of selected annotators are used to maintain their consolidated knowledge.

The advantage of our proposed solution is its ability to unlock semantic information from proprietary documents - especially PDF - seamlessly, and allowing new annotators to be added modularly on demand, after a training interaction with the end user. The direct integration with the users' current workflow and the transparent use of semantic technologies eases the acceptance of the solution

by the users. The personalized knowledge views - which are built for each individual editing unit - ensure that the users are not overwhelmed with annotations. Instead they only visualize the annotations they are interested in.

The rest of the paper is structured as follows: after exploring available state of the art (Sect. 2) we describe in detail our use case (Sect. 6), our system (Sect. 4) and present results on a sample of real documents (Sect. 5). We conclude with lesson learned and future work (Sect. 7).

2 State of the Art

Much of today's scientific and technical content is locked behind proprietary document formats, making it difficult to consume for analytic systems. There has been much positive shifting especially in the context of scientific publishing, where many publishers have been showcasing the benefit of augmenting scholarly content with semantic information. Examples are the SciGraph project[1] by Springer-Nature, the Dynamic Knowledge Platforms (DKP) by Elsevier[2] among others. Academic projects such as Biotea [8] pursue the same goal of creating machine readable and sharable knowledge extracted from scientific content in proprietary format (specifically XML files). Academic initiatives have been encouraging the idea of "semantic publishing" [17] where the authors themselves augment their scientific papers with semantic annotations, instead of relying on post-processing information extraction performed by third parties. Other initiatives aim at maintaining sharable knowledge about the metadata of scientific publication [13].

While significant effort has been put into extracting and maintaining semantic information from scientific publications, much of the content is still locked inside PDF files. This is even more true for technical documents that are not necessarily scientific papers, but which still contain extremely valuable information. The use case that we present in this paper specifically focuses on a particular type of technical documents, the medication Package Insert (PI) [12], which provide physicians with information about the proper use and risks of a prescription drug.

There are several efforts in the literature which explore extracting information from PDFs directly. Early examples [19,20] focus on parsing textual content and extracting structured information. They do so without maintaining the user interaction with the original files. This is undesirable, especially in cases where the layout of the text (e.g., tables) or ancillary information (e.g., chemical structures or other illustrations) are critical context to the understanding of the text. More recent examples exploit the specific structure of certain PDF files, therefore also using specific visual clues of the documents to train the extraction models [1,2,18].

On the other hand we propose a solution that is agnostic of any specific structure of the input file and that is fully integrated within a PDF, and thus can

[1] https://www.springernature.com/scigraph.
[2] http://data.elsevier.com/documentation/index.html.

be viewed with the PDF reader the subject matter expert is already using. Our solution allows the user to visually identify the information which is semantically relevant for their business case. Such information is used by the system to train semantic annotators, which are then integrated directly in the PDF viewer tool that the subject matter expert is already using.

In the healthcare and pharmaceutical domains there are numerous proposals to extract semantic information from available data [5,14–16] but the underlying assumption is that they either operate within a proprietary system or that the semantic annotations are performed offline, in a pipeline fashion, where the SMEs cannot tune the models, nor correct or personalize the results. The major methodological advantage in our proposed solution is that we integrate IE tools - based on ontological annotators and on models which are trained on the fly with human-in-the-loop - directly within the PDF data format, fostering acceptance by the subject matter experts.

3 Use Case Description: Extracting Knowledge from Medical Package Inserts

The use case that we address in this work is the following: given a set of drugs of interest, an internal team of knowledge curators (also referred as subject matter experts) has the task to maintain updated knowledge about each drug. The knowledge curators are organized in editorial units, where each unit is tasked with curating specific portion of the knowledge, e.g. one unit may be tasked to identify all adverse drug reactions for a drug, another unit to deal with all dosages information etc. The source documents from where this information need to be extracted are the medication Package Inserts. A Package Insert (PI) is a document included in the package of a medication that provides information about that drug and its use. In U.S.A., annually all pharmaceutical companies must provide updated information about all their drugs to the U.S. Food and Drug Administration (FDA),[3] including the PIs. All this information is then made publicly available on the FDA Web site. *DAILYMED*[4] provides access to 106,938 drug listings including daily updates. Such daily updates can be very useful to monitor the changes in the Package Inserts. For example, new adverse drug reactions could be added, the dosage of the drug is changed, new drug interactions are discovered, etc. Such information is highly valuable to patients and medical practitioners.

The editorial units are tasked to extract relevant information as well as to identify changes in those information every time an updated Package Insert is released for a certain medication. Their workflow involves manually reading the PDF file of a newly released Package Insert, comparing it to the latest previously available version and identifying all relevant new information to be added to the current knowledge base. The tooling they use is mainly based on standard diff

[3] https://www.fda.gov/.
[4] https://dailymed.nlm.nih.gov/dailymed.

tools available within PDF viewer software and then each knowledge curator manually identifies relevant information to be added, changed or deleted from the knowledge base.

Without disrupting their habitual workflow, we perform standard information extraction tasks directly on the PDF documents and embed the results in the PDF format, so that they can decide to visualize additional semantic information to aid their task. Having such semantic annotations - e.g., drug names mentions, adverse drug reactions, dosage terms, and important textual changes - can speed up the process of identifying the most relevant information in the updated documents. The user can still decide to toggle those annotations off, in the same fashion they toggle on/off the display of changes between different versions of the PDFs.

For the particular use case of analyzing PIs, it is important to retain the documents in their exact same form, as in many cases it is crucial to have in situ analysis by a human. Let's take the examples of tables. The table in Fig. 1 lists some potential side effects for a particular drug, which are important to retain in the curated knowledge. Nonetheless, despite the advance of table interpretation techniques [10,22], none of them can guarantee perfect accuracy, especially for tables like the one in Fig. 1 where the schema is particularly complicated to identify and where the caption text is integral for the understanding. Moreover, even for a human in this case it would be difficult to get a complete view of what the risks are with this drug, including the organ systems affected, the symptoms produced by the drug in each organ system, and the relative size of the risk for each symptom unless the table and the accompanying text are read contextually.

Table 1 **Adverse Reactions Occurring at an Incidence of ≥2% in Patients Treated with Telmisartan/Hydrochlorothiazide and at a Greater Rate Than in Patients Treated with Placebo***

	Telmisartan/ Hydrochlorothiazide (n = 414)	Placebo (n = 74)	Telmisartan (n = 209)	Hydrochlorothiazide (n = 121)
Body as a whole				
Fatigue	3%	1%	3%	3%
Influenza-like symptoms	2%	1%	2%	3%
Central/Peripheral nervous system				
Dizziness	5%	1%	4%	6%
Gastrointestinal system				
Diarrhea	3%	0%	5%	2%
Nausea	2%	0%	1%	2%
Respiratory system disorder				
Sinusitis	4%	3%	3%	6%
Upper respiratory tract infection	8%	7%	7%	10%

* includes all doses of telmisartan (20 to 160 mg), hydrochlorothiazide (6.25 to 25 mg), and combinations thereof

Fig. 1. A table of adverse reactions from a medication Package Insert.

It is for this reason that *in situ* presentation of the results is so critical in this particular use case, where the editorial units are required to guarantee perfect accuracy and completeness of the produced knowledge. We therefore do

not replace the manual annotation process - which is a requirement to guarantee perfect results - but rather enrich the documents with multiple semantic annotations, that can be approved or disapproved by the SMEs and aid the annotation process.

We offer a combination of ontology based annotators as well as the possibility to add new semantic annotators which are trained on demand, within the PDF itself, with a human-in-the-loop methodology, and modularly added to the document. In Sects. 4 and 5 we give all the details of the implemented annotators as well as quantitative indications of how they contribute to the enrichment of the documents.

4 System Overview

In the following we will focus on the specific instantiation of the system for analyzing PI documents and discuss the specific implemented annotators. Nonetheless the architecture and the methodology is general and can be replicated for different use cases and domains.

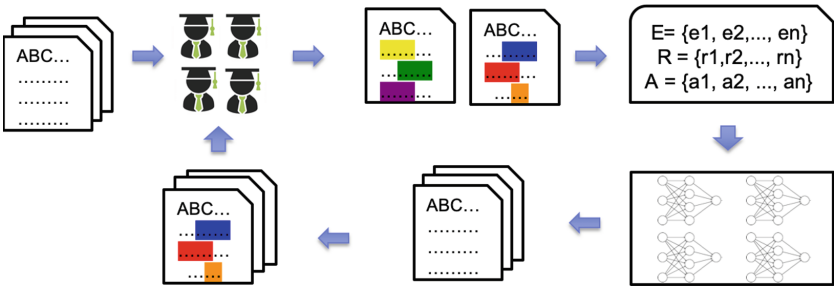


Fig. 2. Document annotation system workflow.

The system takes as input a collection of PI documents $D = \{d_1, d_2, \ldots, d_n\}$ and enriches each document d_i with semantic annotations. The implemented semantic annotators include a set of entity types $E = \{e_1, e_2, ..., e_n\}$, relations between entities $R = \{r_1, r_2, ..., r_n\}$ and textual annotations $A = \{a_1, a_2, \ldots, a_n\}$.

Figure 2 shows the overall workflow of the system. In essence the annotation process runs in two phases, i.e., (i) initialization and (ii) adjudication.

Initialization. The subject matter experts (SMEs) upload the desired collection of PI documents to the system. The system prompts any readily available semantic annotators, which results can be immediately added within the documents. Using existing knowledge based annotators gives us a fast access to readily available knowledge, which can improve the efficiency of the SMEs in

their work. For example, for this specific use case we rely on BioPortal,[5] which provides a text annotator based on several hundreds bio-medical ontologies which are extensively used in applications in the pharmaceutical domain. After those annotations are added, the SMEs start annotating the documents, i.e. marking the entities, relations and textual annotations of interest, and by doing so populating the sets of entities E, relations between entities R and textual annotations A. Once a small number of semantic annotation have been added, the system builds state-of-the-art machine learning models for each type of semantic annotations.

Entity Extraction. For identifying entities of interest, the system offers two co-existing options: (i) standard knowledge based annotators as well as (ii) allowing the users to build custom named entity recognition systems. While the selected knowledge based annotators can already provide extensive coverage, in many cases the SMEs require the identification of custom types of entities, for which there is no existing knowledge, or it cannot be trivially obtained. Therefore, the system allow the SMEs to define entity types, and provide models for rapid identification of such entities. For this purpose we use our internal dictionary expansion approach, the *Domain Learning Assistant* (DLA). The SMEs start by manually annotating a few entities of each type, which are then used by *DLA* to propose new candidate entities. DLA currently employs two set expansion engines: Explore and Exploit (EnE) [4,9] and Glimpse [3,6]. *EnE* builds a neural language model on the input text corpus, word2vec and BiLSTM, and given a few initial seeds, identifies new potential dictionary entries. EnE operates in two phases: (i) the Explore phase identifies instances in the input text corpus which are similar to existing dictionary entries, where the similarity is based on the term vectors from the neural language model, using cosine similarity; (ii) the Exploit phase constructs more complex multi-term phrases based on the instances currently in the dictionary, based on a relatedness function. EnE uses the input documents D to generate a single neural language model which is used for generating the dictionaries for all the types of entities. EnE proposes novel additional entities in an incremental fashion: the more the SME accepts proposed entities, the more are suggested. The full description of the EnE algorithm can be found in [9] and the evaluation shows that the EnE approach outperforms the related work approaches on 5 different dictionaries in 3 different tasks. *Glimpse* on the other hand explores the pattern space. It generates patterns of words that occur on either side of seed terms and scans the corpus for other words that match those patterns[6]. Patterns are scored as a function of their produced matches. The peculiarity of *Glimpse* is that it generates all potential patterns (typically tens of millions) and then retrospectively scores them after searching

[5] https://bioportal.bioontology.org/.

[6] As an example, if we consider "apple" as a seed term, Glimpse looks for all occurrences of "apple" in the underlying corpus and generates patterns using wildcards, such as "I like to eat * for breakfast" and "I invest in * stock". Further details can be found in [3,6].

their occurrence in the text. A very high speed pattern matcher allows it to scan gigabytes of text with millions of patterns in just a couple minutes. The two engines are complementary to each other. EnE is quite fast, but does better on entities that are a 1–3 tokens long. Glimpse is slower, but works well on longer token entities (3+). Their combination allows the SMEs to build large lexicons in a short time [4,9].

Relation Extraction. We employ a state-of-the-art neural network architecture for relation classification [21]. The input of the neural network is a text with two marked entities, i.e., the entities for which the system tries to extract a relation, and it doesn't require complicated syntactic or semantic preprocessing of the text. The first layer of the network is a word embeddings layer, where each token in the input text is replaced with an n-dimensional embedding vector. In the second layer of the network a feature vector is generated, which is a concatenation of lexical and sentence level features. As lexical features we use the marked entities and the surrounding tokens, i.e. one token on the left and one on the right of the target entity. The sentence level features include word contexts with window size of 3 and positional features, i.e., the distance between the entities in the text, which are then passed through a convolutional layer and one non linear layer to get the final sentence level feature vector. The lexical feature vector and the sentence level vector are concatenated in one single vector and fed into a fully connected softmax layer, where the output is the confidence score for each of the relations in the domain. The system builds a separate model for each type of entities, i.e., each model processes one type of entities, and N possible relations between the entities.

Textual Annotations. We offer the option to identify full sentences which are of particular interest for the SMEs, e.g. all sentences which as a whole express potential adverse drug events. We use a Convolutional Neural Network (CNN) text classifier and we train it to classify sentences and paragraphs. The architecture of our CNN is inspired by Collobert et al. [7] and Kim et al. [11], which have shown high performance in many NLP tasks. We selected the following parameters for the CNN model: an input embedding layer, 4 convolutional layers followed by max-pooling layers, a fully connected softmax layer, rectified linear units, filter windows of 2, 3, 4, 5 with 100 feature maps each, dropout rate of 0.2 and mini-batch size of 50. For the embedding layer we use word2vec embeddings trained on 20,000 package inserts, with size 300. We train 100 epochs with early stopping.

In conclusion, for all the annotations, the SMEs provide some seed examples of the information they want to extract and specify external Knowledge Resources to be used, if any. The learning models for each type of annotations are learnt and updated as the user interacts with the system.

Adjudication. The initial models are applied to the whole document collection. The SME performs the adjudication of the produced semantic annotations and

can (i) correct the mistakes made by the automatic annotation system and (ii) identify and add missing annotations. The adjudication is done directly in the document, and all the collected information is transferred to the system. After each batch of corrections (where the batch size can be adjusted), the models are retrained and reapplied on the rest of the documents. New semantic annotations can be added at any time, i.e., once a new item is added the models are retrained and are able to identify the new item, being entity or textual annotation.

With such a system the SME has full control of what types of semantic annotations will be identified, and they can enforce that the accuracy of the system is always above a certain threshold (even 100% if they are willing or required to manually review the whole collection). The system simply assists the user to improve their efficiency in identifying the semantic annotations of interest, and reducing the human error.

5 System in Action: Document Annotation and Knowledge Graph Generation

In this section we show how we use the system for annotating a set of PDF medication Package Inserts and generating a knowledge graph from them (Sect. 5.1). We then go into quantitative details of our experiments (Sect. 5.2). The experiments are performed with an internal team of healthcare professionals on 300 random drugs, for which we retrieved the last 5 versions of their package insert, totaling to a set of 1,500 documents.[7] The backend system is deployed on a machine with 100 cores and 1TB of RAM. The time to update the models is within couple of seconds, which is not noticeable for the users.

5.1 Processing the Documents

The processing of the documents is done in 3 steps: (i) document parsing; (ii) document annotation and (iii) knowledge graph population.

PDF Parsing. To keep our system independent of the input data format, we transform each document to an internal JSON representation model, thus enabling the processing of any type of document, as long as a parser is implemented. As for processing PDF files we use the Apache PDFBox library,[8] which provides functionalities for creating new PDF files, manipulating existing documents and the ability to extract content from documents. Furthermore, it allows injecting JavaScript code directly in the PDF document, which we use to implement the full human-in-the-loop interaction - accepting, correcting, rejecting or adding new annotations. In the JSON file we keep all content, structural and meta-data information to produce the exact same PDF file when needed. For

[7] We make sure that all selected drugs have at least 5 versions, obtained from DAILYMED.

[8] https://pdfbox.apache.org/.

example, we preserve each token with the information for the bounding box of the token, the style and the token's id. Furthermore, we have implemented a set of rules for identifying sentences, sections, titles and tables, which are also stored in our internal data model. Many of the PDF files in our internal use case are available as images (i.e. they are older scanned documents), but it is still very important to include them in the knowledge base. For those files we use optical character recognition system Tesseract OCR,[9] to first convert the images into machine-encoded text, which is then converted to our internal data model.

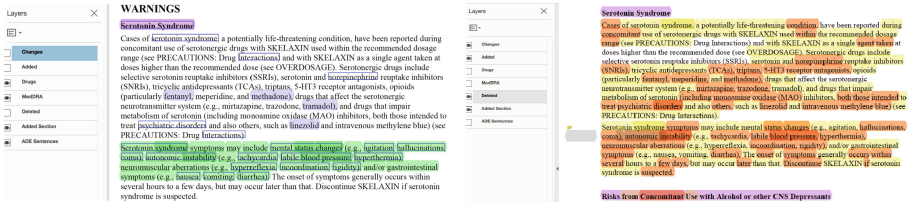
All the JSON files are stored in a non-SQL MongoDB database,[10] for fast retrieval and text search.

Document Annotation. As explained in Sect. 4, our system is able to generate models on the fly for various semantic annotations. In our experiment, we use the following semantic annotators:

- *Entities:* We use the BioPortal API[11] for annotating the documents with existing entities from different types. BioPortal offers 764 biomedical ontologies, which contain valuable information for annotating Medication Package Inserts. In this particular case we matched the documents against all the ontologies and ranked them by their utility (number of returned matches) and let the user decide which ones to retain. Besides the external Knowledge Base, the team of SMEs built 6 internal entity extraction models using the DLA approach, i.e., "Symptoms", "Dosage", "Frequency", "Body Part", "Route" and "Clinician". The models were generated iteratively, as the SMEs were making progress through the document collection.
- *Relations:* In this experiment, we build one model for identifying drug-drug interactions (DDI) between drugs. The DDI relation extraction model achieved F-score of 76.92% on our set of documents.
- *Textual annotations:* In this experiment, we build one model for identifying sentences that express Adverse Drug Reactions (ADE). The ADE sentence classification model achieved F-score of 83.2% on our set of documents. We compare this approach to two baseline text classification approaches, i.e., Random Forest and Support Vector Machines, using bag of words with TF-IDF representation, achieving 76.63% and 69.81% F-Score, respectively.
- *Structural annotations:* As the Medication Package Inserts files can significantly change over different versions, it is of paramount value for the SMEs to quickly identify those changes. As we are storing all the content and structural information of the documents, we can easily identify content and structural changes in each version, including content relocation.

All these annotations are added to the JSON representation of the document and stored internally. Furthermore, we preserve the provenance of the editorial

[9] https://github.com/tesseract-ocr/tesseract.
[10] https://www.mongodb.com/.
[11] http://data.bioontology.org/documentation.



a) *Example of semantically annotated Medication Package Insert.*

b) *Example of content and structural annotations of a Medication Package Insert.*

Fig. 3. Examples of annotated Medication Package Insert.

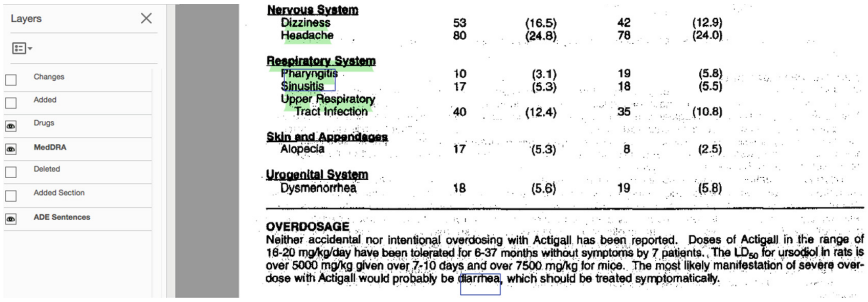


Fig. 4. Example of annotated Medication Package Insert saved in an Image PDF format

unit that applied the changes to the documents. After each annotation we re-create the PDF and present to the SMEs, where each editorial units will be presented documents with the "personalized" annotations. The produced semantic annotations enrich the initial document, without altering its layout (and potentially obscuring the context needed to understand the text). To realize that, we add semantic layers on top of the original document, where each layer contains the information for a specific semantic annotation. Figure 3 shows an example of an annotated PI, as depicted in Adobe Acrobat Reader.[12] The results of the semantic annotators can be toggled on and off at will. For annotators that implement entity resolution (those produced by knowledge based annotators), the recognized entities are linkable and refer to the external sources. Figure 4 shows an example of annotated image PDF file.

Knowledge Graph Population. To generate the knowledge graph, for each drug we maintain a unique ID. For each drug we then maintain the set of different versions, including meta-data, e.g., date of publishing. For each version we store all the semantic annotations. For the entity types, we store the links to the corresponding ontologies or DLA models. We use the relation extraction models to set links between the drugs in the graph, e.g., DDI links. An excerpt of the

[12] https://get.adobe.com/reader/.

resulting knowledge graph for the collection of 300 drugs, is given in Fig. 5. The example shows the different versions of the drugs "Exparel" and "Betadine". For each drug we have a set of different semantic annotations, which are linked to the corresponding ontologies and DLA models. Furthermore, as the DDI relation extraction model identified that there is a relation between the two drugs, we add a link between them in the graph (labeled as "DDI").

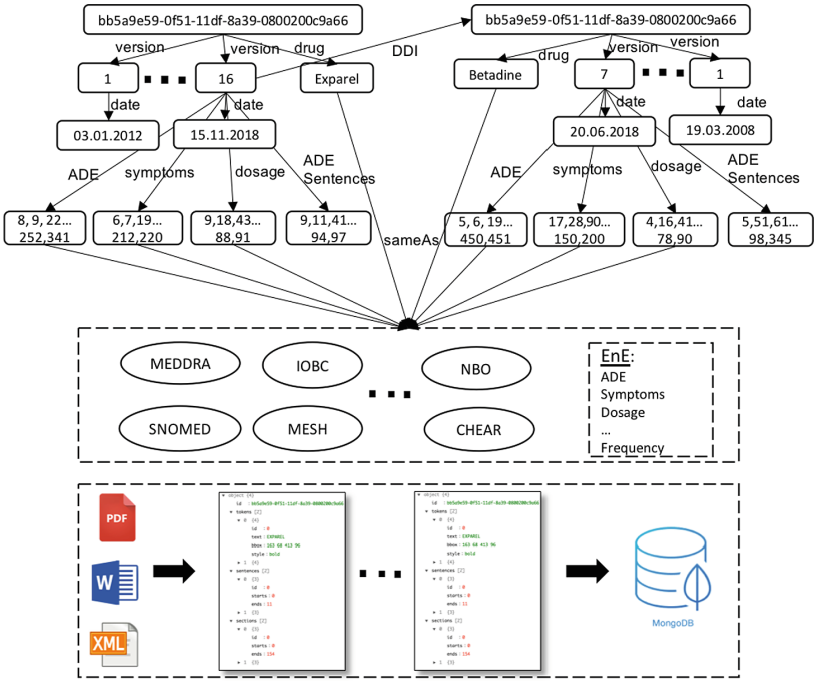


Fig. 5. Architecture of the linked Knowledge Graph generation pipeline.

5.2 Quantitative Results and Data Profiling

In this section, we give insights of the resulting knowledge graph for the selected 300 drugs. First, we show the importance of preserving and linking different versions of the Medication Package Inserts for each drug. Second, we show the importance of involving human-in-the-loop for identifying the correct semantic annotations.

In the first experiment, we capture how the documents evolve over time with each new version. As a proxy of the document changes we use the number of semantic annotations as a measure for comparison between the different versions. We divide all the documents in 5 bins, where bin number 5 contains all

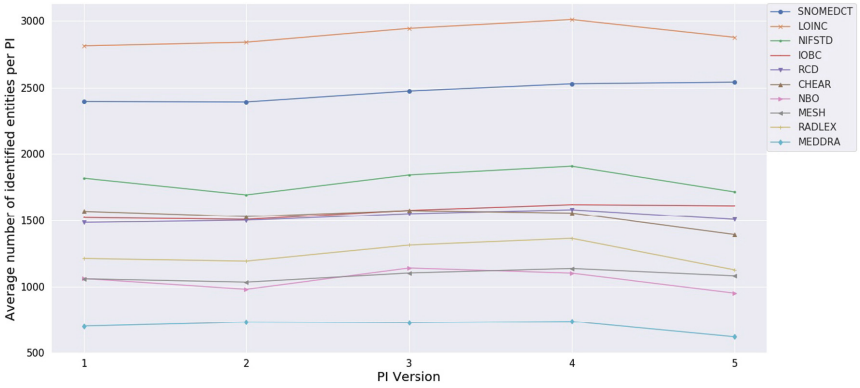


Fig. 6. Average number of identified entities per PI, over 5 different versions.

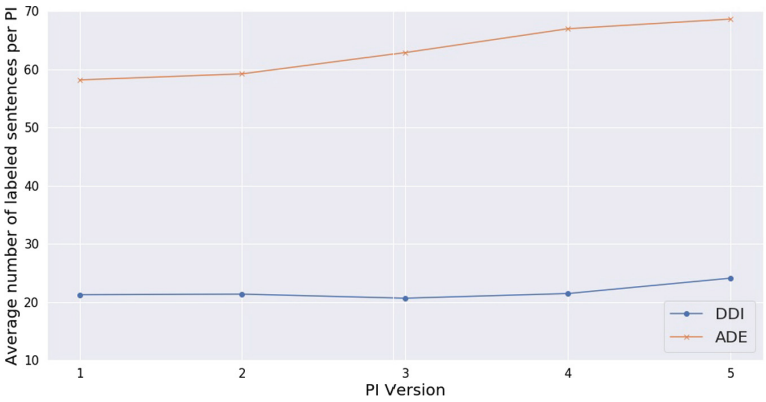


Fig. 7. Average number of identified drug-drug interaction relations (DDI) and Adverse Drug Reactions textual annotations (ADE), over 5 different versions.

the latest versions of the 300 PIs and each other bin contain their previous versions respectively. We count the number of identified semantic annotations for each bin. Figure 6 shows the identified entities per PI for the top 10 ontologies from BioPortal.[13] Figure 7 shows the average number of (i) sentenced labeled as expressing relevant Adverse Drug Reaction (ADE) and (ii) identified drug-drug interactions (DDI). In both charts we can observe that there is significant fluctuation between the different versions of PIs. While in Fig. 6 there is no apparent trend, in Fig. 7 we can observe a strongly increasing trend of discovered ADE textual annotations and DDI relations. This confirms that there are significant changes between different versions of the PIs, and it is important to identify those changes.

[13] Top 10 as for this use case, i.e. those 10 ontologies producing the bigger number of annotations in total.

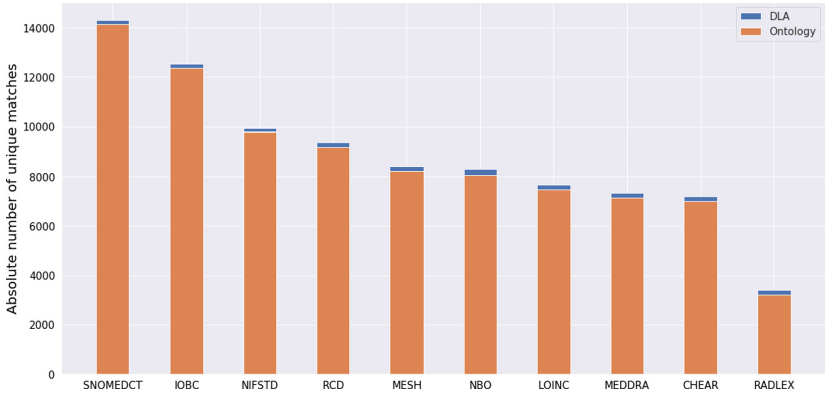


Fig. 8. Number of identified entities using BioPortal and DLA models in the whole document collection. (Color figure online)

In the second experiment, we show the importance of having a human-in-the-loop within the process of semantic annotation. The aim of this experiment is to highlight that regardless of the coverage that standard ontologies can provide - and in this use case BioPortal achieves very high coverage - the SMEs will always need custom entity extraction models which are application dependent. As DLA models are built directly on the corpus at hand and capture the subjective needs of the SMEs, their usefulness is perceived as high by the knowledge curators. Even a small dictionary of ∼100 entities has a high number of matches and this is because the entities accepted by the human-in-the-loop for the dictionary are the ones which are highly relevant to them for the specific use case. The standard ontologies produce a higher number of matches, but that is also a result of their sheer size. Figure 8 depicts the number of unique entities identified in the top 10 ontologies (in orange color). The blue stacked bars show the fraction of entities that was not present in the ontologies and was matched using the DLA models. We can see that the models built using the DLA approach are able to identify a number of entities that are missed by the ontologies, which despite being small in absolute numbers, it is significant if comparing the size of the DLA models to the size of the ontologies. To quantify the contribution of the DLA models in relative terms, we count the average number of matches per document per each ontologies and calculate the fraction of the ontology they represent. Figure 9 depicts the average of this fraction on all the documents, grouped by each of the considered ontologies and the DLA models. For most of the ontologies the matches are basically contributed by 3% to 6% of the whole ontology, while the DLA models match more than 16% of its entities per document. In turns, this is perceived by the users as higher control on the annotation process, as the personalized model represent a concise and efficient view of the domain. Most of the entities that are not matched by the ontologies are result of the specificity of the annotation task by the SMEs. For example, a lot of phrases used to express the recommended frequency of usage for a medication are not

covered by the ontologies, e.g., "NUM times per day", "as-needed", "every NUM hours", similarly for the dosage, e.g., "NUM mg", "NUM tsp", "NUM ml". Such instances are easily identified with the DLA model when deployed in a human-in-the-loop environment. Furthermore, the DLA models can also capture misspellings, e.g., "dialy" was used in a number of PIs, instead of "daily".

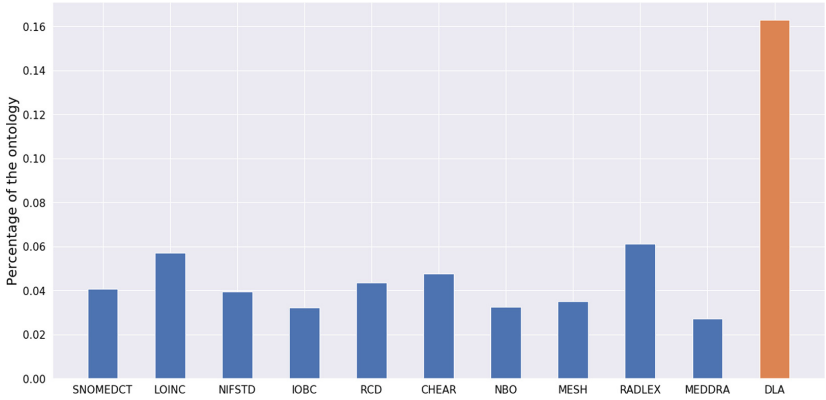


Fig. 9. Matched fraction of entities per ontology and DLA models per document.

6 Pilot Study and Lessons Learned

The pipeline and tooling described in this work are currently used internally by one of the IBM business units working on curating a pharmaceutical Knowledge Graph. The team involved in our pilot consists of 2 Medical Information Specialists and around 20 medical professionals - mainly pharmacists and a few nurses - that have the mission of synthesizing and summarizing various medical content, especially clinically-focused information on drugs, diseases, and toxicology management. Their primary and solely full time task is to perform this knowledge curation. The curated knowledge serves as Knowledge Base for point-of-care clinical decision support tools but also to create consumer-focused drug and disease information for patients who can be informed participants in their care.

Since its deployment on our internal cloud, the extraction pipeline has been used by the team to process between 4 to 15 documents per day. The team was previously only relying on built-in functionalities from Adobe Acrobat Reader, such as the diff tool, but since the deployment of our pipeline and after an initial comparison with their previous solution, the team entirely relies on our extraction pipeline. From the informal feedback that we gathered from the team, they identified two striking benefits of the extraction pipeline. First, the ability to

compare different versions of medical documents to precisely identify and highlight the changes which are semantically relevant, while ignoring the multitude of changes that have no impact on the final KG. Second, the possibility to decide which annotations to visualize and when, so to reduce the information overload and to focus only on the current task at hand. The important lesson learned for us, by being the technology providers for the editorial team, was the importance of abstracting as much as possible from the semantic technology itself and focusing on providing the minimal but most useful annotations to the users. A key value for our user was the fact that we consolidated the knowledge and showed only the portion which was useful to the current user, for the particular task at hand. The clinical head of the team analyzed the time commitment of their team when using our extraction pipeline and stated that they saved a significant amount of time, up to 3.5 h per document when dealing with image PDF documents (which could take up to 8 h to review). Image PDF documents constitute 8% of their material and our extraction pipeline was the only available solution to semantically annotate and compare them as they had no alternative solution before.

7 Conclusions and Future Work

Accurate understanding of technical documents is crucial, especially when updating knowledge graphs where every error can be compounded by downstream analytics. This understanding often hinges on the context in which the information appears. Especially in the case of data formats such as PDFs, this leads to the costly requirement of human review of a large number of documents.

In this paper we showcase our human-in-the-loop pipeline to transparently deliver semantic annotations within PDF documents. Within-document annotations also allow to materialize personalized graphs on-the-fly on any selection of documents, document versions, annotation types, user permissions, etc. We enable the subject matter experts to rapidly create and train their own annotation engines, as well as using any readily available ontology based annotator, all within the PDF documents, without suffering disruption from changing tools or formats. While we showcase the use of the system with the use case of medical Package Inserts, the approach is applicable to many other scenarios, such as scientific publishing, the recruiting business - where many applicants' resumes are PDF files - legal contracts, electronic parts data sheets, material hazard data sheets, aircraft flight manuals, just to name a few. In all these scenarios it is crucial to be able to quickly train the extraction models and to deliver the results in a way that is familiar to the subject matter expert, as well as having ways to track changes between document updates.

References

1. Abekawa, T., Aizawa, A.: SideNoter: scholarly paper browsing system based on PDF restructuring and text annotation. In: COLING 2016, pp. 136–140 (2016)

2. Ahmad, R., Afzal, M.T., Qadir, M.A.: Information extraction from PDF sources based on rule-based system using integrated formats. In: Sack, H., Dietze, S., Tordai, A., Lange, C. (eds.) SemWebEval 2016. CCIS, vol. 641, pp. 293–308. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46565-4_23
3. Alba, A., Coden, A., Gentile, A.L., Gruhl, D., Ristoski, P., Welch, S.: Multi-lingual concept extraction with linked data and human-in-the-loop. In: Proceedings of the Knowledge Capture Conference, p. 24. ACM (2017)
4. Alba, A., Gruhl, D., Ristoski, P., Welch, S.: Interactive dictionary expansion using neural language models. In: Proceedings of the 2nd International Workshop on Augmenting Intelligence with Humans-in-the-Loop co-located with (ISWC 2018). CEUR Workshop Proceedings, vol. 2169, pp. 7–15. CEUR-WS.org (2018)
5. Barisevičius, G., et al.: Supporting digital healthcare services using semantic web technologies. In: Vrandečić, D., et al. (eds.) ISWC 2018. LNCS, vol. 11137, pp. 291–306. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00668-6_18
6. Coden, A., Gruhl, D., Lewis, N., Tanenblatt, M., Terdiman, J.: SPOT the drug! an unsupervised pattern matching method to extract drug names from very large clinical corpora. In: HISB 2012, pp. 33–39 (2012). https://doi.org/10.1109/HISB.2012.16
7. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. J. Mach. Learn. Res. **12**(Aug), 2493–2537 (2011)
8. Garcia, A., Lopez, F., Garcia, L., Giraldo, O., Bucheli, V., Dumontier, M.: Biotea: semantics for Pubmed central. PeerJ. 1–26 (2018). https://doi.org/10.7717/peerj.4201
9. Gentile, A.L., Gruhl, D., Ristoski, P., Welch, S.: Explore and exploit. Dictionary expansion with human-in-the-loop. In: Hitzler, P., et al. (eds.) ESWC 2019. LNCS, vol. 11503, pp. 131–145. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-21348-0_9
10. Gentile, A.L., Kirstein, S., Paulheim, H., Bizer, C.: Extending rapidminer with data search and integration capabilities. In: Sack, H., Rizzo, G., Steinmetz, N., Mladenić, D., Auer, S., Lange, C. (eds.) ESWC 2016. LNCS, vol. 9989, pp. 167–171. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-47602-5_33
11. Kim, Y.: Convolutional neural networks for sentence classification. In: EMNLP 2014, pp. 1746–1751. ACL, October 2014. https://doi.org/10.3115/v1/D14-1181
12. Nathan, J.P.: The package insert. US Pharmacist **40**(5), 8–10 (2015)
13. Nuzzolese, A.G., Gentile, A.L., Presutti, V., Gangemi, A.: Conference linked data: the scholarlydata project. In: Groth, P., et al. (eds.) ISWC 2016. LNCS, vol. 9982, pp. 150–158. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46547-0_16
14. Piro, R., et al.: Semantic technologies for data analysis in health care. In: Groth, P., et al. (eds.) ISWC 2016. LNCS, vol. 9982, pp. 400–417. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46547-0_34
15. Roberts, K., Demner-Fushman, D., Tonning, J.M.: Overview of the TAC 2017 adverse reaction extraction from drug labels track. In: TAC (2017)
16. Segura Bedmar, I., Martinez, P., Sánchez Cisneros, D.: The 1st DDIExtraction-2011 challenge task: extraction of drug-drug interactions from biomedical texts. In: Proceedings of the 1st Challenge Task on Drug-Drug Interaction Extraction 2011 (2011)
17. Shotton, D.: Semantic publishing: the coming revolution in scientific journal publishing. Learn. Publishing **22**(2), 85–94 (2009). https://doi.org/10.1087/2009202

18. Staar, P.W.J., Dolfi, M., Auer, C., Bekas, C.: Corpus conversion service: a machine learning platform to ingest documents at scale. In: SIGKDD 2018, pp. 774–782. ACM (2018). https://doi.org/10.1145/3219819.3219834
19. Yuan, F., Liu, B.O.: A new method of information extraction from PDF files. In: ICMLC 2005, pp. 18–21. IEEE (2005)
20. Yuan, F., Liu, B., Yu, G.: A study on information extraction from PDF files. In: Yeung, D.S., Liu, Z.-Q., Wang, X.-Z., Yan, H. (eds.) ICMLC 2005. LNCS (LNAI), vol. 3930, pp. 258–267. Springer, Heidelberg (2006). https://doi.org/10.1007/11739685_27
21. Zeng, D., Liu, K., Lai, S., Zhou, G., Zhao, J., et al.: Relation classification via convolutional deep neural network. In: COLING, pp. 2335–2344 (2014)
22. Zhang, Z.: Effective and efficient semantic table interpretation using tableminer$^+$. Semant. Web **8**(6), 921–957 (2017). https://doi.org/10.3233/SW-160242