Construction of Semantic Data Models

Martha O. Perez-Arriaga(✉), Trilce Estrada, and Soraya Abad-Mota

University of New Mexico, Albuquerque, NM 87131, USA
{marperez,estrada,soraya}@cs.unm.edu

Abstract. The production of scientific publications has increased 8–9% each year during the previous six decades [1]. In order to conduct state-of-the-art research, scientists and scholars have to dig relevant information out of a large volume of documents. Additional challenges to analyze scientific documents include the variability of publishing standards, formats, and domains. Novel methods are needed to analyze and find concrete information in publications rapidly. In this work, we present a conceptual design to systematically build semantic data models using relevant elements including context, metadata, and tables that appear in publications from any domain. To enrich the models, as well as to provide semantic interoperability among documents, we use general-purpose ontologies and a vocabulary to organize their information. The resulting models allow us to synthesize, explore, and exploit information promptly.

Keywords: Table understanding · Information modeling
Semantic data model · Semantic interoperability
Information extraction

1 Analyzing Scientific Publications

Modern research relies significantly on exploring and building on top of existing scientific production. But in recent decades this production has increased exponentially. With this growth, new challenges have to be addressed. Exploration of very large digital corpora can be done manually, but it is time consuming, tedious, and potentially results in incomplete analyses. On its automatic form, this exploration involves two main phases: document retrieval and analysis. Document retrieval can be done efficiently through keyword search, phrase matching, topic categorization, and other more sophisticated information retrieval techniques. The analysis phase has to consider the semantics of the document, as well as its qualitative and quantitative content. However, this information can be hard to get, and in many cases it is buried in tables whose relationships have to be inferred.

Because of the exposed problems, we develop a conceptual design to extract semantic information from digital documents. In general, the design of a data model requires the knowledge of entities that interact with each other. Our design allows us to systematically compose a data model without knowing its elements

J. Filipe et al. (Eds.): DATA 2017, CCIS 814, pp. 46–66, 2018.
https://doi.org/10.1007/978-3-319-94809-6_3

a priori. Thus, we recognize its components and interactions at the same time of analyzing a document.

According to Peckham and Maryanski [2], a *semantic data model* includes two main components: (1) relationships between entities and (2) well-defined semantics for the relationships. The richness of information in a digital publication provides the elements to create a semantic data model. The concepts in a publication are entities that appear along the narrative of a study, in a publication's context, and embedded in tables or other structures in a document. To define, disambiguate, and enrich entities, additional information can be extracted from external sources of knowledge, such as ontologies and the Internet, to use them as semantic annotations. A *semantic annotation* is an association between a concept in a document and a definition contained in an established database or ontology.

To enrich concepts within publications, several works integrate semantic annotations in these documents [3,4]. For instance, BioC [4] provides a design to define annotations from publications of the biomedical field, offering tools for developers to create definitions in XML. The tools are simple to use, however, it is necessary to have programming skills to take full advantage of them. The International Association of Scientific, Technical, and Medical publishers points out the importance of publishing with semantic practices [5]. To include semantic annotations before publishing documents, the Semantic Web Science Association [6] defines rules to improve promoting and sharing articles related to the Semantic Web. Peroni [7] presents semantic publishing and referencing ontologies to create metadata and include semantic annotations in scientific articles. However, these publishing practices are far from being extensively adopted and the bulk of scientific publications in most fields lack even minimum annotation standards. Searching for concepts in documents lacking semantic annotations makes the extraction process harder. Thus, an ideal process to analyze information should be interoperable and include a direct way to identify elements of interest, such as concepts with semantic annotations and context. Semantic interoperability refers to integrated information systems that are able to hide syntax and structural heterogeneity from data sources [8], while providing shared and unambiguous information.

Our goal is to automatically build semantic models on the fly to characterize and annotate documents. Our model creation is fully automatic, as it does not need prior information regarding concepts, entities, or metadata standards. It is also interoperable, as our conceptual design provides an extensible and standardized mechanism for unambiguous information exchange through semantic annotations and provenance. Finally, it is exhaustive, as it takes advantage of qualitative information, found through the document's narrative, as well as quantitative information, found in implicit relationships in tables.

To build data models, we (a) identify and extract context, metadata, and concepts from a publication and its tables; (b) detect and extract semantic relationships; and (c) characterize a semantic data model from each publication. A semantic data model derived from our conceptual design provides a semantic characterization of quantitative and qualitative relationships in the document. Our contributions in this paper are twofold: a formal definition to systematically generate semantic data models to facilitate synthesis, extraction, and comparison of specific information; and a characterization of data models from scientific documents of any domain for semantic interoperability. We extend our previous work [9] introducing the conceptual design of semantic data models for synthesizing publications. Although this design is for publications, the analysis of large volume of information in any domain can benefit using a conceptual design. Using a working example, we demonstrate the possibilities of finding semantic relationships at each level of a model, which can potentially guide building knowledge bases. Furthermore, we improve the methods to disambiguate entities and extract semantic relationships, deepen on the details of our characterization that encloses data integration and semantic interoperability, and measure similarity of semantic relationships to compare information among data models.

In the remainder of this work, Sect. 2 presents a review of related work. Section 3 presents a framework to develop the models, including the organization of a semantic data model using a working example. Section 5 presents an assessment of the methods to generate our models. Section 6 presents the conclusion and future work to be undertaken.

2 Related Work

We review methods used to obtain the main elements to compose the data models. In particular, we present related work to identify and extract semantic relationships, and to summarize information from digital documents.

Important work to extract semantic relationships include the Never Ending Learning Language (NELL) [10], PATTY [11], and Open Information Extraction (OIE) methods [12–14]. The NELL [10] approach creates a knowledge base of categories and relationships from the Web. This approach extracts patterns and relationships, classifies noun phrases, and infers new relationships from their knowledge base. On the other hand, the approach PATTY [11] is based on frequent itemset mining. PATTY also detects relationships from the Web, but this approach uses syntactic, ontological, and lexical features. The Open Information Extraction approach finds new relationships without using patterns [12,13]. Fader et al. state *"OIE can find incoherent and uninformative extractions."* and improve it developing Reverb [15]. Reverb finds relationships using part of speech tags, noun phrase sentences, and syntactic and lexical constrains. Reverb uses a corpus built offline with Web sentences to match arguments in a sentence heuristically, and finds a confidence percent for each relationship using a logistic regression classifier. Reverb and R2A2 are part of the second generation of OIE [14]. R2A2 includes an argument learner, and linguistic and statistical analyses to extract relationships. The learner consists of classifiers using specific patterns, punctuation, and part of speech tags to identify arguments.

We used Reverb in our previous work [9], which detected semantic relationships in publications with a confidence measure. The high ranked relationships found by Reverb missed relationships containing entities derived from tables. Therefore, we used the wealth of information in tabular structures and a context to retrieve relationships with relevant arguments from a publication. Similarly to OIE methods, our approach uses an unsupervised learning and heuristics. However, we focus our attention on finding relationships with entities of interest. Methods for extraction of semantic relationships, which are based on Web and text formats, lack methods to preserve the association between a source document and its relationships, and lack an organization of relationships per document. To overcome these issues, our framework integrates relevant information from publications as relationships and enhances them with external sources of information, organizes the relationships in a common characterization document, preserving the original publication's provenance for further consultation.

The second part of this section reviews work to represent semantic information and to summarize information from digital publications. Hull and King [16] describe semantic data models and their ability to (1) separate concepts in physical and logical forms, (2) minimize overload of relationships, and (3) represent abstract concepts and relationships. The models have been used to include semantics in a database schema, which is difficult to represent and implement. Peckham and Maryanski [2] analyze semantic data models and determine the two main components: relationships between entities and well-defined semantics for the relationships. The models represent data objects and relationships known *a priori*. For example, the *World Traveler Database* modeling [16], which represents relationships among known entities. Conversely, our conceptual design uses entities from a context and tabular layouts embedded in any digital document, as well as relationships from internal elements in a document (i.e., tables and text), and external sources of information, such as the general ontology DBpedia and the Internet. To disambiguate entities, we use DBpedia [17] to describe each entity. DBpedia is a curated structured ontology with entities' properties. Furthermore, DBpedia contains information that follows the Semantic Web principles. If DBpedia lacks an entity's information, this work uses the unsupervised approach *Latent Semantic Indexing* [18] and a publication's context to disambiguate an entity, providing an explanation of the entity at hand from the Internet. Furthermore, we use the Semanticscience Integrated Ontology [19] to provide structure and semantics to the relationships derived from scientific publications. Finally, to represent a publication's metadata, we use the general vocabulary schema.org [20] in a machine-readable format.

Summarization provides a short representation of the content of publications and can be used to analyze them faster. Automatic approaches advance this research area. Nenkova et al. [21] present an extensive description of methods for summarization in different areas, such as medical, journals, and news. Teufel and Moens [22] point out differences in summarizing documents from science or news, the latter being more repetitive and less concise in their arguments. We also notice that scientific documents often have space restrictions and their narrative is succinct. Allahyari and colleagues [23] present a survey of text summarization methods and point out the importance of ontologies in summarization. Several approaches to summarize publications use rethoric analysis using the sentences in each section. Teufel and Moens analyze relevance in sentences and rhetorical status, based on analyzing organization, contribution, and citations in a scientific article. Baralis et al. [24] use frequent itemsets to summarize documents. Our framework generates a short summary, synthesizing a publication based on ontologies, relevant concepts, and semantic relationships.

3 Generation of Semantic Data Models

To find concrete information embedded in a vast collection of scientific documents, we develop a framework to create semantic data models. Figure 1 depicts this process. First, the framework ingests a document in PDF, XML, or text formats. Then it extracts metadata and context from the document's title, keywords, and authors. Then it parses tabular information to perform discovery and interpretation of tabular information. This information is then used to guide the process of entity and relationships discovery. Finally, all of these pieces, metadata, context, entities and semantic relationships, are used to form the semantic data models that characterize the document. These models can be further used for summarization, annotation, and searching purposes. In the following sections, we present the approach to recognize the different elements in a digital publication and compose our models.

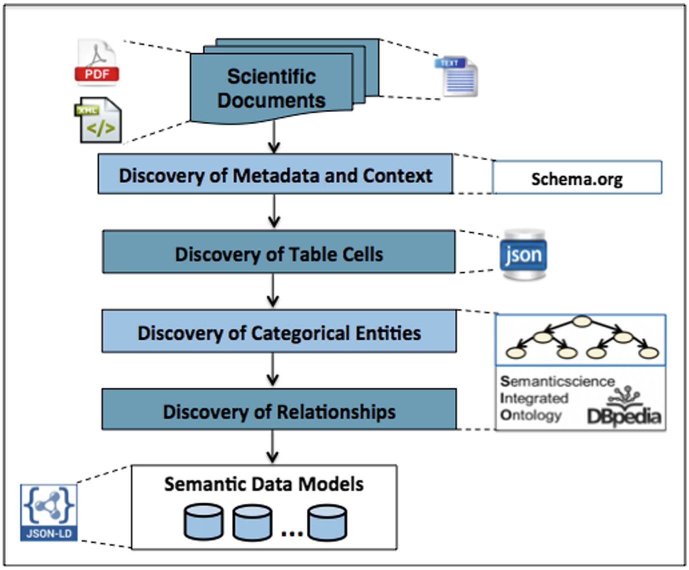
PDF
XML
Scientific Documents
TEXT
Discovery of Metadata and Context
Schema.org
Discovery of Table Cells
json
Discovery of Categorical Entities
Discovery of Relationships
Semanticscience Integrated Ontology
DBpedia
JSON-LD
Semantic Data Models
...

Fig. 1. Framework to generate semantic data models.

3.1 Discovery of Metadata and Context

Metadata. Represents the key information to identify a publication, such as title and author. It is important for digital identification, resource discovery, electronic resources organization, interoperability, archive and preservation [25]. Metadata can be represented in different formats, for instance, using the standard generalized mark-up language, extended markup language (XML), or even using a database to represent and manage metadata attributes [26]. To recognize metadata in digital publications, we directly identify tags associated with metadata terms. Our method searches for tags describing the metadata elements in documents, including `<author>`, `<article-title>` and `<keywords>`. If tags are not found, our framework accepts a digital document in plain text or PDF format to search for metadata. A digital document in PDF is converted to XML using PDFMiner [27]. Then, we use pattern matching on the first page of the publication to extract this information. The keywords defined in a publication are used to represent its context. If keywords are not defined in a document, we search for them. For representing the extracted metadata, we use the schema.org vocabulary [20] because it can represent information from different domains with minimum dependency on ontologies. In particular, we use properties such as creator, headline, and keywords of the scholarly article object to identify the different metadata elements. By using a controlled vocabulary, we ensure an homogeneous data representation, that is vital to facilitate interoperability, sharing, and data collaboration.

Context. Represents the conceptual framework surrounding a publication. Current publishing standards dictate that publications need to define keywords to describe the domain and sub domains of the document. Additionally, the text itself contains concepts that can be used as context as well. To extract the publication's context, our framework first searches for keywords. If keywords are not defined, then the text undergoes a preprocessing step, where we eliminate stop words, and small and large length words. Each word is then used as a unigram. We use the Term Frequency Inverse Document Frequency (TFIDF) method to score the relevance of each word in the context of the document as opposed to a word that is frequently used regardless of its context. This method uses term frequency (TF) to detect a word's relevance to a particular document, and it uses the inverse document frequency (IDF), which is the logarithm of the ratio obtained using total number of documents in a collection and the number of documents where the term appears. To calculate IDF we use Wikipedia as a wide collection of documents, which contains more than five million of documents to date [28]. Once we calculate the TFIDF score for every word in the document, we use the five terms with highest scores as additional keywords of the publication. These keywords become part of metadata that characterizes the semantic model of the publication.

3.2 Discovery of Categorical Entities

In addition to context, we extract categorical entities from keywords, text, and tables. In this section we describe how to recognize, disambiguate, and annotate entities as presented in [9], as well as how to find and associate annotations to entities.

Information Extraction from Tables. For entity extraction, our first step is to explore tables in search for quantitative information. Tables present concise associations in a simple representation. A study found that 74.6% of tables appear in the results section in scientific publications of different domains [29]. The main challenge of discovering and extracting tables from documents is that they are embedded within text and other elements, such as equations and graphics. Also, they are created using different formatting and layouts.

Our framework is able to discover table cells from XML and PDF documents. For the former, we extract tables content from well-formed XML documents using the tags `<table-wrap>` and `<table>`. We use Xpath [30] to detect tags and find a table within a document. Then, the table cells are organized by rows in JavaScript Object Notation (JSON) format. In addition, we determine if a cell is a header or data using the tags `<thead>` and `<tbody>`. A header is a label that represents and groups table information, rows or columns, while data is the actual content of a cell that compose the body of a table. Finally, we detect data types for each table cell using regular expressions. The process for table discovery in PDF documents is more elaborated. We use our previous work in TAO [31] that consists of three modules: (1) Document conversion, which uses PDFMiner [27] to convert a PDF document to an extensive document in XML. The output includes separate XML tags for every text box, character, and space, including its coordinates, font, and size. In this format it is not possible to identify a table and its content without further processing (2) Table detection, parses the output in XML using a combination of layout heuristics to detect table candidates within the document. (3) Table extraction uses table candidates and supervised learning to find a table's content. Specifically, we use k-nearest neighbor logistic regression to find alignment of columns, which determine an actual table. Then, we extract the content of each table cell and analyze cells to classify their data type and function, that is, header and data. TAO's output is saved in JSON format as well.

Entity Recognition and Annotation. To recognize entities, each table cell's content whose data type is a string, undergoes natural language processing using TextBlob [32], which executes a noun phrase analysis to discover entities. To make sure that every entity found is unambiguous we identify it and annotate it through DBpedia [17]. To this end, we search the entity in DBpedia using its naming convention, which utilizes its first capital letter and concatenates words with an underscore. For example, the entity *diabetes management* converts to "Diabetes_management." DBpedia redirects searches that refer to the same concept automatically. Searching for "Glycemic_control" or "Diabetes_treatment", DBpedia returns the entity's resource for "Diabetes_management" because the former concepts appear in the property `wikiPageRedirects` of this entity.

DBpedia offers a Web page, that is, a resource for each entity containing a description, a type, and properties of a particular entity. For instance, the entity *Diabetes* has the resource http://dbpedia.org/page/Diabetes_mellitus, which defines this entity with type *disease* and includes several properties. In particular, the property *abstract* comprises a summarized description of the entity, and we use it as an annotation *"Is A"* for the entity. We also keep the URI and store it as an annotation to identify the entity and for further consultation.

Entity Disambiguation. Even though DBpedia is a curated structured ontology, it does not contain a description for every entity possible. Moreover, if a concept has more than one meaning (i.e., it is ambiguous), DBpedia shows a list of possible concepts under the property *wikiPageDisambiguates*. For example, currently DBpedia shows forty different concepts for the entity *Race*. When the property *wikiPageDisambiguates* exists, we use a global search on the Internet to disambiguate and find a URL to unambiguously explain an entity. To narrow down the search results and to obtain more accurate results, we augment the search with the document's context.

We use the Web Search API from Microsoft Cognitive Services [33] and a tailored Latent Semantic Indexing (LSI) analysis [18] to find the closest explanation to an entity and its context. We use LSI because it performs well categorizing documents with only several hundred dimensions [34]. In particular, we search for an ambiguous entity x and retrieve at most n documents $D = d_1, d_2, \ldots, d_n$ containing x, where $n = 50$. For each $d_i \in D$, we normalize a vector with at most 300 relevant words. Then, we create a normalized query vector q containing the context, title, and abstract of a publication, as well as sentences containing the entity of interest in the original publication. After applying LSI, we select the most similar vector $d_i \in D$ to vector q and we recover its associated URL to use as an identifier for entity x. The URL is a non-formal annotation to represent entity x. It is non-formal because differently from a URI, a URL may change over time, and because its content is not very often curated.

3.3 Discovery of Relationships

Through the process described above, we create a list of unambiguous and annotated categorical entities per document. We then used these entities to find meaningful semantic relationships in the publication. We base our definition of a semantic relationship on concepts by Dahchour et al. [35], who defines a *semantic relationship* as a binary relationship that is domain independent and represents static constrains and rules, such as, classification, generalization, grouping, and aggregation. We discover relationships using table cells and text. To find relationships from tables, we relate each data table cell with its header. The table cells' organization containing rows, column numbers, content, and cell function facilitate this process. We also use the text of the document and table's caption to find additional relationships.

Our first approach, described in [9], used the open information extraction method Reverb [15] to identify relationships. However, Reverb was not able to identify many relevant relationships associated with entities derived from tables. Therefore, we designed a more extensive relationship identification process comprising four steps: (a) segmentation, (b) pattern matching, (c) part of speech tagging, and (d) relationship composition. First, we recover the text of a publication and perform segmentation of sentences, using the Python Natural Language ToolKit [36]. Once we recover all sentences of each publication, the categorical entities are used to search for pattern matching in these sentences. If a sentence contains an entity, we apply part of speech tagging. Then, we detect combinations of tags with verbs, such as `VB`, `VBZ`, `MD VB`, `MD have VBN`, `MD has VBN`, `MD VBZ`, and `VB VBZ`. The verb tags assist to identify sentences containing an action between an entity and other text. If a verb combination is found, we use it as a relationship. For example, the sentence *obesity is related to lack of exercise* contains the relationship *is related*. Last, we use the Semanticscience Integration Ontology [19] (SIO) to formally represent a relationship with its ontology definition. We use pattern matching to relate a definition of a relationship from SIO with our extracted relationships. If a match is not found, we keep a verb as the relationship itself. For instance, the sentence *the experiment is derived from our study* has the verb *is derived*, which is denoted in SIO by the label *is derived from* and id *SIO_000244*. But the sentence *obesity increases high blood pressure* has the verb *increases*, which is not included in SIO, hence, *increase* represents this relationship.

3.4 Organization of a Semantic Data Model

Our proposed semantic data models integrate important information derived from elements found in publications regardless of their domain. Specifically, to compose a semantic data model, we use associations found in tabular patterns and unstructured text. Thus, we define a semantic data model as the structured representation of semantic elements in a publication. Specifically, it contains metadata and context, entities, and semantic relationships.

A semantic model generated by our framework is represented and stored as a JSON-Linked Data (JSON-LD) object. JSON-LD [37] was created to facilitate the use of linked data. A document in JSON-LD can define a type of the information represented using a context and its relationships are not limited by a specific cardinality. This format is compatible with RDF and other variants such as N-quads [38].

Our framework uses JSON-LD to generate a descriptive document with the publication's metadata, entities with annotations, and the actual information from extracted relationships (i.e., relationship identifier, arguments, definition of relationship, and definition identifier). The document in turn contains other links, such as the ones used to annotate entities and to define relationships. In addition, it contains a model context, which denotes the environment of the components of the semantic data model. Note that the context of a semantic data model is different to the context of a publication, which is represented using keywords. Using as a working example the publication: *Neuropeptidomic Components Generated by Proteomic Functions in Secretory Vesicles for Cell-Cell Communication* [39] by Vivian Hook et al., we find the entity *Neuropeptides.* We represent it in JSON-LD, which includes a model context indicating that this entity is identified by a resource from the ontology DBpedia.

```
{
 "context": "{http://dbpedia.org/page/}"
 "Neuropeptides": {
       "id":"Neuropeptides",
       "abstract":"small protein-like molecule (peptides) used
                   by neurons to communicate with each other"
}
```

To compose the semantic data model to be used as a synthesis of a document, we build a semantic hierarchy of its different components. Where the top components represent the most general information and the bottom layers represent specific identifiers and annotations. Figure 2 shows a partial graphic depiction of this hierarchy.

The first layer of the model corresponds to the *metadata and context*, which includes identifiers for keywords, headings, and provenance. The next layer contains *entities*, where specific concepts are identified as relevant entities in the document (e.g., Neuropeptide, pain). Then the following layer contains *semantic relationships*, which link one or more entities and resources (e.g., Has Attribute, Is A). Finally, the bottom layer contains specific *annotations* for both entities and relationships The discovery of relationships in our semantic models allows researchers to find results, experimental settings, and possible associations among concepts in scientific documents. The hierarchical organization of our data model facilitates finding relationships among entities, and later among publications.

4 Conceptual Design of Semantic Data Models

To provide semantic interoperability among digital publications, we present a conceptual design to create semantic data models systematically. From the elements contained in a publication, we identify the finite components of a *semantic data model* as a 3-tuple $SDM = \{M, E, SR\}$. M represents metadata and context, E entities, and SR semantic relationships. Fundamental elements subsumed in this model are table cells that serve to find entities E and semantic relationships SR. To depict the conceptual design of these components, we use the Entity-Relationship (ER) model as described by Elmasri and Navathe [40]. Rectangles indicate entities, diamonds indicate relationships, and values at both ends of a relationship (min, max) are a combined cardinality/participation notation, which indicate the structural constraint of the minimum and maximum participation of an entity in a relationship.

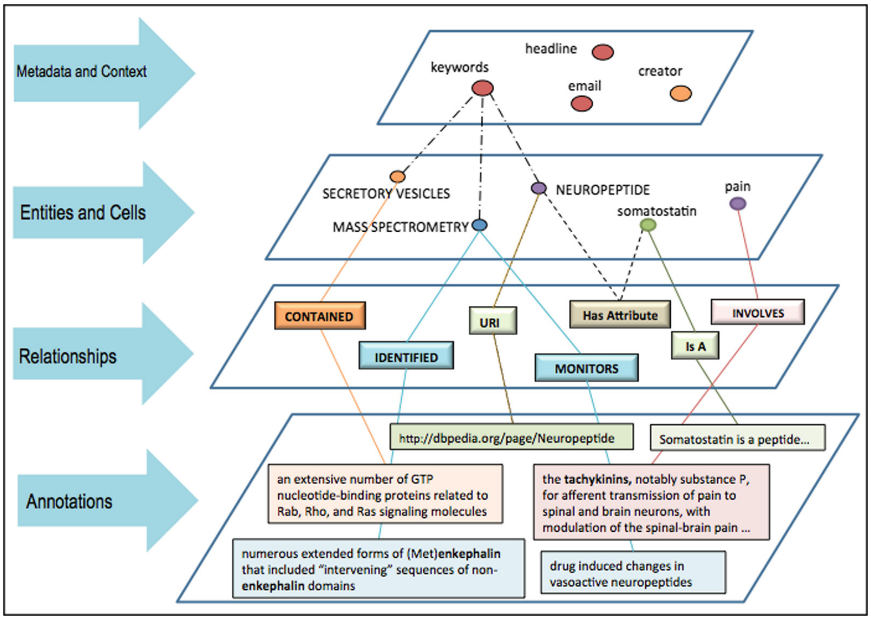
Metadata and Context
headline
keywords
creator
email
Entities and Cells
SECRETORY VESICLES
NEUROPEPTIDE
pain
somatostatin
MASS SPECTROMETRY
Relationships
CONTAINED
URI
Has Attribute
INVOLVES
IDENTIFIED
Is A
MONITORS
Annotations
http://dbpedia.org/page/Neuropeptide
Somatostatin is a peptide...
an extensive number of GTP nucleotide-binding proteins related to Rab, Rho, and Ras signaling molecules
the tachykinins, notably substance P, for afferent transmission of pain to spinal and brain neurons, with modulation of the spinal-brain pain ...
numerous extended forms of (Met)enkephalin that included "intervening" sequences of non-enkephalin domains
drug induced changes in vasoactive neuropeptides

Fig. 2. Semantic data model from working example.

Metadata and Context. The first component of our model, M, includes metadata and context of a publication. Figure 3 shows the conceptual design of this component, which includes a context of a publication as a set of keywords, a title of a publication, author(s), and corresponding email. A data model representing a publication is primarily identified with a unique title, which can also relate to a unique Digital Object Identifier. A publication is associated to one or more authors. If a first author has collaborator(s), then for each pair *first author-collaborator* an *isCoauthor* relationship exists. Finally, each author has one email address.

Cells and Categorical Entities. Figure 4 shows the conceptual design of cells C and categorical entities E. First, the element C is composed by a set of cells contained in a given table. The definition of a semantic relationship between cells is indicated by a **header cell**, which *hasAttribute* contained in a **data cell**. A header cell can be related to more than one data cell, while a data cell has to have exactly one primary header cell.

The element E represents a set of entities. The entities can be found in tabular structures and within the context of a publication. To detect entities E, we can use cells in C with data type string and keywords representing a context. Figure 4 shows the semantic relationship **cell** *canBe* an **entity**, to indicate that an entity can be found in a cell. The relationship **entity** *composedBy* **keyword** indicates that E at least contains one keyword, and that a keyword can be in one or more entities.

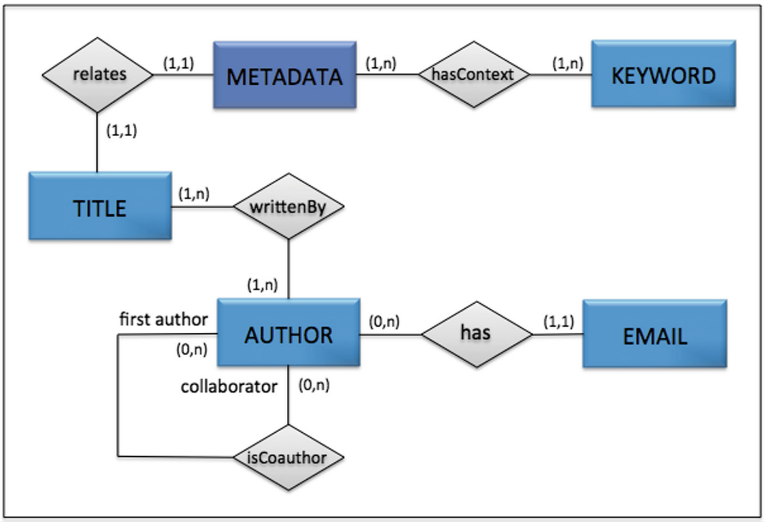

Fig. 3. Metadata components. Reprinted from [41]

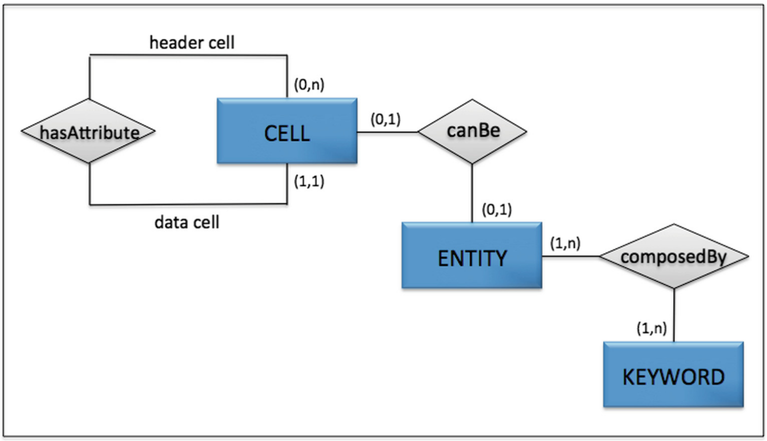

Fig. 4. Cells and entities. Reprinted from [41]

Semantic Relationships. The last element of our model contains semantic relationships SR. The components of SR include entities, text, and semantic annotations. The annotations for categorical entities contain information from DBpedia or the Internet. Figure 5 shows the relationship of an entity and its semantic annotations. In particular, a semantic annotation consists of two relationships: **ontology** *defines* an **entity** and a **URI** *describes* an **entity**. A URI is a Universal Resource Identifier and a URL is a Uniform Resource Locator. The relationships for a semantic annotation include *defines* or *IsA*, and *describes*.

If semantic annotations are not found, an entity can relate to a URL, where a **URL** *explains* an **entity**. The relationship *explains* is not necessarily a semantic annotation because a URL can change over time and may not contain a formal definition. Figure 5 also shows **entity** *associates* to a phrase in **text**.

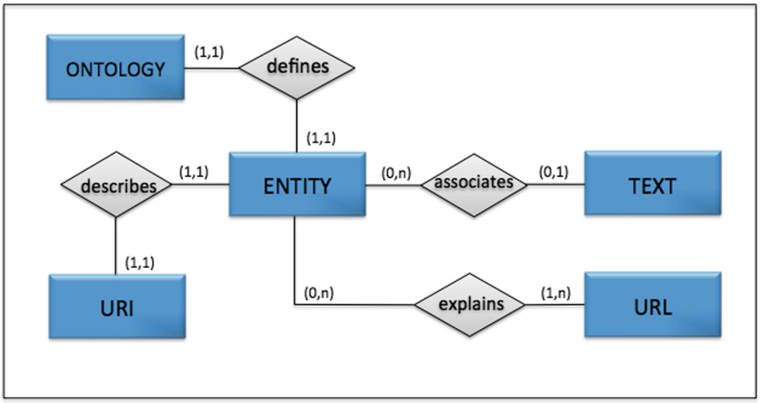

Fig. 5. Semantic relationships. Based on [41]

The semantic relationship *associates* can acquire different labels describing relationships from text. We can find these labels defined in the Semanticscience Integrated Ontology [19], which contains formal definitions of relationships, for instance, *is derived from* and *has basis*. The relationships from SIO are not exhaustive, therefore, verbs can also represent relationships, as explained in Sect. 3.3.

5 Evaluation

To assess the effectiveness of our method, we extend the evaluation from [9] for discovery of semantic relationships. In particular, we use two datasets. The datasets contain publications downloaded from the PubMed Web site ftp://ftp.ncbi.nlm.nih.gov/pub/pmc/oa_bulk/. The first dataset $PubMed_1$ comprises fifty publications with various topics, 449 text pages and 133 tables with different tabular formats embedded either in one- or two-column documents. We prepared a gold standard with manually-defined entities for each table in the publications. With similar layout characteristics, the second dataset $PubMed_2$ is composed by seventy five publications with 670 text pages and 190 tables.

We evaluated our methods quantitatively and qualitatively with four experiments to measure our framework's ability to (1) recognize and annotate entities, (2) disambiguate entities, (3) identify semantic relationships between entities, and (4) generate similar semantic relationships among data models. The dataset $PubMed_1$ is used in the first three experiments.

5.1 Experiment: Entity Recognition and Annotation

The first experiment measures our method's ability to recognize and annotate entities. We use *recall* and *precision* to obtain the F1-measure for entity recognition. *Recall* is the ratio between the correct number of entities detected and the total number of entities in a table. *Precision* is the ratio between the correct number of entities detected and the total number of entities detected. *Recall* and *precision* are also used to evaluate accuracy for entity annotation.

Our gold standard contains 2, 314 entities, and our method recognized 1, 834. We obtained a recall of 79.2% and a precision of 94.3%, yielding a F1-measure of 86.1% (see Table 1).

For entities annotated, we found 1, 262, from these entities, 72.5% were found in DBpedia. However, only 785 contained a unique description in DBpedia, that is, 45.1%. For the rest, our method used the context of a publication and the LSI process described in [9] to annotate 955 entities (i.e., we annotated 54.9% of entities with our LSI + context method). From the 1, 834 entities recognized, 1, 740 were correctly annotated, yielding a recall of 94.8%, precision of 97%, and F1-measure of 95.9%.

Table 1. Experiment entity recognition and annotation. Reprinted from [9].

Entity recognition			
Entities	Recall	Precision	F1 measure
Recognized	0.79	0.94	0.86
Annotated	0.95	0.97	0.96

5.2 Experiment: Entity Disambiguation

The second experiment evaluates our entity disambiguation methods. We quantify how including the context of each publication affects the entity disambiguation process. From 1, 740 entities, 955 of them needed disambiguation. The first part of this experiment did not use a context to disambiguate entities. Still, our framework was able disambiguate 838 entities correctly without context, with precision 89%, recall 87%, and an F1-measure of 88%. During the second part of this experiment, we used keywords as a context, yielding 900 entities disambiguated with precision of 95%, recall 94%, and an F1-measure of 94.5%. Table 2 reports these results.

To evaluate the quality of the disambiguation process, the URLs obtained from this process were manually classified as reliable and non-reliable (see Table 2 column *Non Rel. URLs*). Reliable URLs were derived from known organizations including universities, digital libraries, and hospitals. While non-reliable URLs required further analysis. Manually reviewing URLs ensures that a Web page or document is related to an unresolved entity and its publication. However, it does not ensure that the explanation of the entity is correct. The non-reliable URLs when no context was used, were 117, that is 12.3% of the total disambiguated entities. The non-reliable URLs found when additional context was used, were 55, that is 5.8%. Therefore, the use of context reduced the non-reliable links by more than half.

Table 2. Experiment entity disambiguation. Reprinted from [9].

Entity disambiguation				
Method	Recall	Precision	F1 measure	Non Rel. URLs
No context	0.87	0.89	0.88	12.3%
Context	0.94	0.95	0.94	5.8%

5.3 Experiment: Identification of Semantic Relationships

The third experiment evaluated both quantitatively and qualitatively the relationships found by our methods. The relationships extracted from tables and text contain relevant concepts in a structured presentation. First, we measure the total number of relationships with high rank, in particular a confidence score ≥ 0.70 using Reverb. Second, we measure the total number of relationships extracted with our new method using an unsupervised method and relevant entities. Furthermore, we manually evaluated qualitatively the relationships classifying them as complete and incomplete. The latter refers to a relationship missing an argument.

From tables in $PubMed_1$ we found $11,268$ relationships. Exclusively from text while using Reverb, we found 865 high ranking (confidence ≥ 0.70) relationships. On the other hand, using categorical entities with our new approach, we found $1,397$ relationships. A human judge analyzed the completeness of relationships manually. Results are reported in Table 3. From the total number of relationships obtained from tables, $10,102$ or 89% of relationships were complete. From the relationships extracted from text using Reverb, 703 or 81% of relationships were complete. The rest, that is 19% was labeled as incomplete. For the approach using entities from tables and text, a judge identified $1,336$ or 90% of them as complete, while the rest 10% was labeled as incomplete. Our improved approach found almost twice the number of relationships than the number found by Reverb. Thus, the we were able to empirically demonstrate that the number of extracted relationships increased using relevant entities.

Table 3. Experiment semantic relationships. Based on [41]

Semantic relationships			
Method	Rel. found	Rel. complete	% complete
Tables	11,268	10,102	89%
Text reverb	865	703	81%
Text entity	1,397	1,336	90%

Using a publication's context (i.e., keywords) and concepts from tables' cells to find entities ensures their relevance. But we still found several incomplete or malformed relationships from text containing information from tables. An area of improvement for our approach is to extract and eliminate irrelevant tables' information in a document to eliminate some false positive relationships from text.

5.4 Experiment: Determining Semantic Similarity

The last experiment used $PubMed_2$ to generate 75 semantic data models. The sets of relationships contained in the models were compared with one another using cosine similarity. Figure 6 shows the matrix of scaled similarity among data models. Every cell i, j in the matrix represents the similarity of model i with model j, thus, the matrix is symmetrical. A lighter shade indicates higher similarity. Through this matrix it is possible to quantitatively identify clusters of documents that share semantic elements such as context, entities, and relationships within a collection.

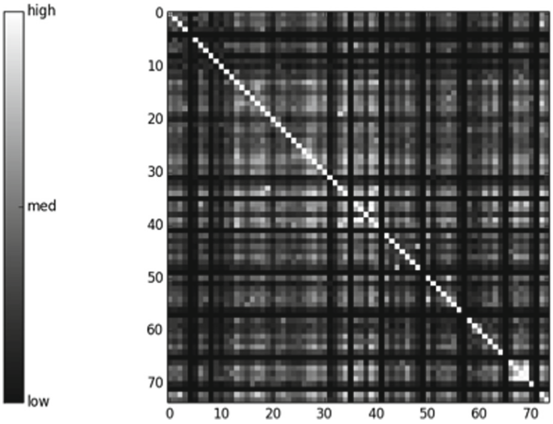

Fig. 6. Scaled similarity matrix.

For the cluster of documents with higher similarity, we performed a manual check to determine the actual similitude of semantic relationships of the data models. The manual check corroborated the high similarity of documents as well as similar entities and context in their models.

We select as an example, a pair of data models from our dataset. The models represent the publications *Plasma Vitamin E and Blood Selenium Concentrations in Norwegian Dairy Cows: Regional Differences and Relations to Feeding and Health* [42] (left) and *Lameness and Claw Lesions of the Norwegian Red Dairy Cattle Housed in Free Stalls in Relation to Environment, Parity and Stage of Lactation* [43] (right).

Figure 7 shows several relationships found in each model and between this pair of models. The models allow the detection of common entities, these are: *milk*, *cow*, and *silage*. The latter concept was found in a relationship from text of the left model and in the set of entities of the right model. Using dashed lines, we show how the common entities in the models relate to each other. The similar concepts in both models enrich and increase the semantic similarity between them. Each model has particular relationships, which are well-defined and structured. The left model has semantic annotations using relationships IsA and URI, as well as non-formal annotations using URLs and a verb. For instance, the entities *mastitis* and *paresis* contain semantic annotations to define and describe them. This model also presents a document with *URL*: http://oregonstate.edu/dept/EOARC/sites/default/files/638.pdf to explain the entity *cowrepro*. We also observe in the model on the right, that it contains semantic annotations of entities, such as *wood*, *silage*, and *pasture*. In addition, the concept *cow* uses the verb *were trimmed* to identify a relationship to text. The models show annotations of entities from a publication, DBpedia [44], and the Internet.

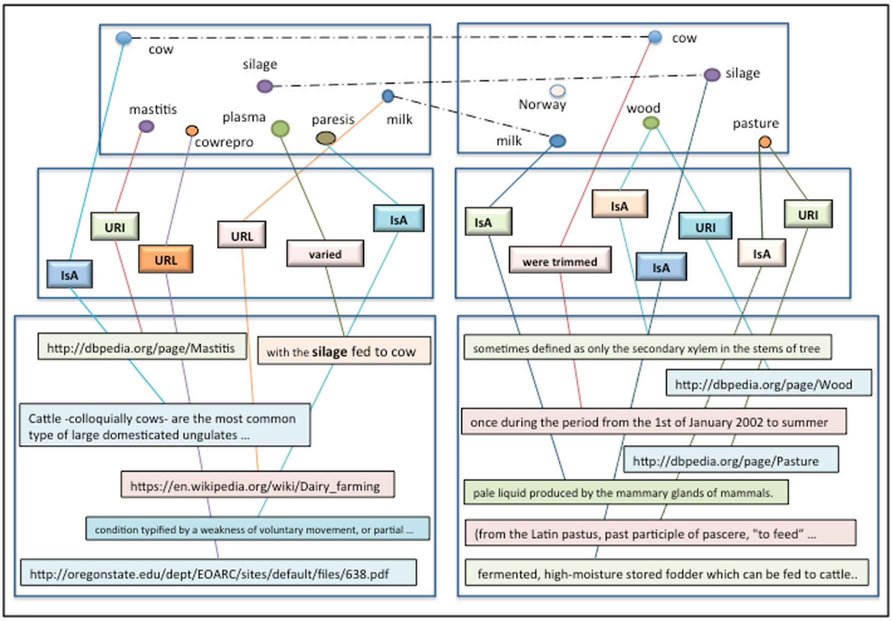
cow
silage
mastitis
plasma
paresis
milk
cowrepro
URI
URL
IsA
URL
varied
IsA
http://dbpedia.org/page/Mastitis
with the silage fed to cow
Cattle -colloquially cows- are the most common type of large domesticated ungulates ...
https://en.wikipedia.org/wiki/Dairy_farming
condition typified by a weakness of voluntary movement, or partial ...
http://oregonstate.edu/dept/EOARC/sites/default/files/638.pdf
cow
silage
Norway
wood
pasture
milk
IsA
IsA
were trimmed
URI
IsA
URI
IsA
sometimes defined as only the secondary xylem in the stems of tree
http://dbpedia.org/page/Wood
once during the period from the 1st of January 2002 to summer
http://dbpedia.org/page/Pasture
pale liquid produced by the mammary glands of mammals.
(from the Latin pastus, past participle of pascere, "to feed" ...
fermented, high-moisture stored fodder which can be fed to cattle..

Fig. 7. Relationships between two data models.

The organization of our data model facilitates finding relationships among entities and later among publications. Because the relationships can extend to more than a pair of data models, the creation of a network of related data models should be easy to accomplish with this organization.

6 Conclusion and Future Work

In this paper we present a conceptual design to generate semantic data models from digital publications systematically. The components of our data models can be used to understand and organize relevant information, synthesizing a publication. The main components of our models contain information derived from the discovery of metadata and context, categorical entities, and semantic relationships.

To organize and annotate the semantic models, we use general-purpose ontologies and a vocabulary to structure well-defined entities and semantic relationships. In particular, we match defined relationships representing association, classification, aggregation, and generalization. Therefore, capturing unambiguous semantic associations between entities. The discovery of well-defined semantic relationships in our models enable researchers to automatically exploring large collections of documents, and retrieving structured and concrete results, experimental settings, and possible hierarchical associations among concepts in a scientific environment.

The representation of our models in a machine-readable format facilitates interoperability. Keeping the provenance of a publication related to a semantic data model can help track back the source of information used to build a particular model. The provenance information is useful to ensure that researchers can access the primary source of information and investigate further an important or relevant publication.

We evaluated our approach to build semantic models through a set of experiments at different points of the process pipeline. Our approach was able to analyze documents and extract quantitative and qualitative information to compose semantic models. Our experiments show the effectiveness of our framework to recognize, enrich, and disambiguate entities, as well as to discover semantic relationships automatically. Furthermore, we compared the similarity of relationships among data models and present a visual depiction of the depth of information that can be used to compare publications automatically.

For future work, we plan to use the data models to create a semantic network. A set of models can compose a network using similar context and entities. Furthermore, we envision that the networked structure and definition of relationships can be used to compare and contrast findings and arguments within and among digital publications.

References

1. Bornmann, L., Mutz, R.: Growth rates of modern science: a bibliometric analysis based on the number of publications and cited references. J. Assoc. Inf. Sci. Technol. **66**(11), 2215–2222 (2015)
2. Peckham, J., Maryanski, F.: Semantic data models. ACM Comput. Surv. (CSUR) **20**(3), 153–189 (1988)
3. Prli, A., Martinez, M.A., Dimitropoulos, D., Beran, B., Yukich, B.T., Rose, P.W., Bourne, P.E., Fink, J.L.: Integration of open access literature into the RCSB Protein Data Bank using BioLit. BMC Bioinformatics **11**, 1–5 (2010)

4. Comeau, D.C., Islamaj Doan, R., Ciccarese, P., Cohen, K.B., Krallinger, M., Leitner, F., Lu, Z., Peng, Y., Rinaldi, F., Torii, M., Valencia, A.: BioC: a minimalist approach to interoperability for biomedical text processing. In: Database, bat064 (2013)
5. Ware, M., Mabe, M.: The STM report: an overview of scientific and scholarly journal publishing (2015)
6. The Semantic Web Science Association. http://swsa.semanticweb.org/
7. Peroni, S.: Semantic Web Technologies and Legal Scholarly Publishing. LGTS, vol. 15. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-04777-5
8. Ouksel, A.M., Sheth, A.: Semantic interoperability in global information systems. ACM Sigmod Rec. **28**(1), 5–12 (1999)
9. Perez-Arriaga, M.O., Estrada, T., Abad-Mota, S.: Table interpretation and extraction of semantic relationships to synthesize digital documents. In: Proceedings of the 6th International Conference on Data Science, Technology and Applications, pp. 223–232 (2017)
10. Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka Jr., E.R., Mitchell, T.M.: Toward an architecture for never-ending language learning. AAAI **5**, 1306–1313 (2010)
11. Nakashole, N., Weikum, G., Suchanek, F.: PATTY: a taxonomy of relational patterns with semantic types. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 1135–1145. Association for Computational Linguistics (2012)
12. Yates, A., Cafarella, M., Banko, M., Etzioni, O., Broadhead, M., Soderland, S.: TextRunner: open information extraction on the web. In Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations, pp. 25–26. Association for Computational Linguistics (2007)
13. Etzioni, O., Banko, M., Soderland, S., Weld, D.S.: Open information extraction from the web. Commun. ACM **51**(12), 68–74 (2008)
14. Etzioni, O., Fader, A., Christensen, J., Soderland, S., Mausam, M.: Open information extraction: the second generation. IJCAI **11**, 3–10 (2011)
15. Fader, A., Soderland, S., Etzioni, O.: Identifying relations for open information extraction. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 1535–1545. Association for Computational Linguistics (2011)
16. Hull, R., King, R.: Semantic database modeling: survey, applications, and research issues. ACM Comput. Surv. (CSUR) **19**(3), 201–260 (1987)
17. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: DBpedia - a crystallization point for the Web of Data. Web Semant. Sci. Serv. Agents World Wide Web **7**(3), 154–165 (2009)
18. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. J. Am. Soc. Inf. Sci. **41**(6), 391–407 (1990)
19. Dumontier, M., Baker, C.J., Baran, J., Callahan, A., Chepelev, L., Cruz-Toledo, J., Del Rio, N.R., Duck, G., Furlong, L.I., Keath, N., Klassen, D.: The Semanticscience Integrated Ontology (SIO) for biomedical research and knowledge discovery. J. Biomed. Semant. **5**(1), 1–11 (2014)
20. Data Model - schema.org. http://schema.org/docs/datamodel.html
21. Nenkova, A., McKeown, K.: Automatic summarization. Found. Trends® Inf. Retrieval **5**(2–3), 103–233 (2011)
22. Teufel, S., Moens, M.: Summarizing scientific articles: experiments with relevance and rhetorical status. Comput. Linguist. **28**(4), 409–445 (2002)

23. Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E.D., Gutierrez, J.B., Kochut, K.: Text Summarization Techniques: A Brief Survey. arXiv preprint arXiv:1707.02268, pp. 1–9 (2017)
24. Baralis, E., Cagliero, L., Jabeen, S., Fiori, A.: Multi-document summarization exploiting frequent itemsets. In: Proceedings of the 27th Annual ACM Symposium on Applied Computing, pp. 782–786, ACM (2012)
25. National Information Standards Organization Press: Understanding metadata. National Information Standards, vol. 20 (2004)
26. Perez-Arriaga, M.O., Wilson, S., Williams, K.P., Schoeniger, J., Waymire, R.L., Powell, A.J.: Omics Metadata Management Software (OMMS). Bioinformation **11**(4), 165172 (2015). https://doi.org/10.6026/97320630011165
27. Shinyama, Y.: PDFMiner: python PDF parser and analyzer (2015). Accessed 11 June 2015
28. Statistics - En.wikipedia.org. https://en.wikipedia.org/wiki/Wikipedia:Statistics
29. Kim, S., Han, K., Kim, S.Y. and Liu, Y.: Scientific table type classification in digital library. In: Proceedings of the 2012 ACM Symposium on Document Engineering, pp. 133–136. ACM (2012)
30. Berglund, A., Boag, S., Chamberlin, D., Fernndez, M.F., Kay, M., Robie, J., Simon, J.: XML path language (xpath). World Wide Web Consortium (W3C) (2003)
31. Perez-Arriaga, M.O., Estrada, T., Abad-Mota, S.: TAO: system for table detection and extraction from PDF documents. In: The 29th Florida Artificial Intelligence Research Society Conference, FLAIRS 2016, pp. 591–596. AAAI (2016)
32. Loria, S., Keen, P., Honnibal, M., Yankovsky, R., Karesh, D., Dempsey, E.: TextBlob: simplified text processing. Secondary TextBlob: Simplified Text Processing (2014)
33. Microsoft Cognitive Services. https://azure.microsoft.com/en-us/services/cognitive-services/bing-web-search-api
34. Zukas, A., Price, R.J.: Document categorization using latent semantic indexing. In: Proceedings 2003 Symposium on Document Image Understanding Technology, UMD, pp. 1–10 (2003)
35. Dahchour, M., Pirotte, A., Zimányi, E.: Generic relationships in information modeling. In: Spaccapietra, S. (ed.) Journal on Data Semantics IV. LNCS, vol. 3730, pp. 1–34. Springer, Heidelberg (2005). https://doi.org/10.1007/11603412_1
36. Bird, S.: NLTK: the natural language toolkit. In: Proceedings of the COLING/ACL on Interactive Presentation Sessions, pp. 69–72. Association for Computational Linguistics (2006)
37. World Wide Web Consortium. JSON-LD 1.0: a JSON-based serialization for linked data (2014)
38. JSON-LD Playground. http://json-ld.org/playground
39. Hook, V., Bark, S., Gupta, N., Lortie, M., Lu, W.D., Bandeira, N., Funkelstein, L., Wegrzyn, J., OConnor, D.T.: Neuropeptidomic components generated by proteomic functions in secretory vesicles for cellcell communication. AAPS J. **12**(4), 635–645 (2010)
40. Elmasri, R., Navathe, S.B.: Fundamentals of Database Systems. Pearson, Boston (2015)
41. Perez-Arriaga, M.O.: Automated Development of Semantic Data Models Using Scientific Publications. University of New Mexico, USA (2018)
42. Sivertsen, T., Vernes, G., Steras, O., Nymoen, U., Lunder, T.: Plasma vitamin e and blood selenium concentrations in norwegian dairy cows: regional differences and relations to feeding and health. Acta Veterinaria Scandinavica **46**(4), 177 (2005)

43. Sogstad, A.M., Fjeldaas, T., Steras, O.: Lameness and claw lesions of the norwegian red dairy cattle housed in free stalls in relation to environment, parity and stage of lactation. Acta Veterinaria Scandinavica **46**(4), 203 (2005)
44. DBpedia. http://dbpedia.org