Bringing Math to LOD: A Semantic Publishing Platform Prototype for Scientific Collections in Mathematics

Olga Nevzorova, Nikita Zhiltsov, Danila Zaikin, Olga Zhibrik,
Alexander Kirillovich, Vladimir Nevzorov, and Evgeniy Birialtsev

Kazan Federal University,
Kremlyovskaya 18 Str., 420008 Kazan, Russia
{onevzoro,nikita.zhiltsov,ksugltronteal,olgazhibrik,
alik.kirillovich,nevzorovvn}@gmail.com,
IgenBir@yandex.ru

Abstract. We present our work on developing a software platform for mining mathematical scholarly papers to obtain a Linked Data representation. Currently, the Linking Open Data (LOD) cloud lacks up-to-date and detailed information on professional level mathematics. To our mind, the main reason for that is the absence of appropriate tools that could analyze the underlying semantics in mathematical papers and effectively build their consolidated representation. We have developed a holistic approach to analysis of mathematical documents, including ontology based extraction, conversion of the article body as well as its metadata into RDF, integration with some existing LOD data sets, and semantic search. We argue that the platform may be helpful for enriching user experience on modern online scientific collections.

Keywords: Linked Data, Ontology Engineering, Ontology Extraction.

1 Introduction

The Linking Open Data (LOD) initiative[1] has recently revealed the added value of representing heterogeneous data from different content providers as a single "cloud" of interconnected objects. The data are loaded and transformed to RDF from various sources including relational databases, web pages, and semi-structured textual documents. The unified structured representation benefits follow-up Linked Data consumers. For example, contemporary semantic search applications like the semantic search engine Sindice[2] or mashup Sig.ma[3] harness the published data to be able to either handle search queries more accurately or aggregate information about entities users are interested in.

[1] http://linkeddata.org

[2] http://sindice.com/

[3] http://sig.ma/

H. Alani et al. (Eds.): ISWC 2013, Part I, LNCS 8218, pp. 379–394, 2013.

At the same time, the LOD cloud lacks up-to-date and detailed data sets on professional level mathematics. Currently, there exist some unofficial data sets that make available information from well-known publishers and online collections in the academic domain including ACM[4], DBLP[5], and CiteSeer[6], as Linked Data. They have contributed a large amount of scientific article metadata to the LOD cloud. However, exposing only article metadata for mathematical papers is palliative, since the primary objects of interest in these documents are formulas and certain parts such as theorems or proofs. In our particular case, we have faced with the requirements of the publishing department at Kazan Federal University, which plans to make publicly available metadata as well as the contents of 1 330 articles of the "Izvestiya Vuzov. Matematika" (IVM, Proceedings of Higher Education Institutions: Mathematics) journal published in 1997-2009. The publisher expects that it will benefit professional researchers and learning students at the university, by providing them opportunities to get access to a knowledge source integrated into the global knowledge base. Thus, our primary goal is to develop a machinery that facilitates the process and, eventually, constructs a new LOD data set having a collection of mathematical scholarly articles.

In the paper, we present our approach of designing and implementing a programming solution to extract a semantic LOD representation of mathematical scholarly papers in a given digital collection. The core of the approach is modeling the given collection of documents as a unified semantic graph. Both the nodes (mathematical knowledge objects) and the edges (relations between them) in it are defined by a set of math-aware vocabularies that specify the logical structure of mathematical documents (theorems, proofs, definitions, formulas etc.) as well as mathematical concepts. In summary, our key contributions are:

- a thorough domain model that includes an ontology of the logical structure of mathematical scholarly papers along with an ontology of mathematical knowledge concepts in Russian/English;
- a language-independent method for extraction of the logical structure elements;
- a method for extraction of mathematical named entities from texts in Russian;
- a method that connects mathematical named entities to symbolic expressions.

The rest of the paper is organized as follows. In Section 2, we meticulously describe our approach for publishing mathematical scholarly papers as Linked Data. Section 3 contains implementation details of the developed prototype. We report on our evaluation experiments in Section 4. Section 5 provides the data set statistics and several use cases. Section 6 gives a brief overview of related work. We conclude and discuss the future work in Section 7.

[4] http://acm.rkbexplorer.com/

[5] http://dblp.rkbexplorer.com/

[6] http://citeseer.rkbexplorer.com/

2 Approach

In this section, we first describe our domain-specific ontologies that provide a vocabulary for extraction methods. Next, we present our solution for NLP and semantic annotation tasks. Finally, we explain our techniques for article metadata extraction and interlinking with existing LOD data sets.

2.1 Domain Model

Mocassin Ontology. The ontology[7] of our Mocassin project[8] aims to capture the semantics of typical structural elements in mathematical scholarly papers. The ontology is a compromise between the semantics of highly formalized models we have seen in the previous works (discussed in Section 6) and facts that can be extracted by automatic methods. Each structural element in Mocassin Ontology represents the finest level of granularity and has its inherent features, such as starting and ending positions, text contents, and functional role. In particular, the ontology defines some ubiquitous document parts, such as theorems, lemmas, proofs, definitions, corollaries etc. Besides, the ontology declares two types of object binary relations – navigational and restricted. The property instances of the first relation type, which is represented by *refersTo* and *dependsOn* relations, tend to occur in mathematical documents when the author points at significant parts of a publication in the form of referential sentences. The part-whole property (*hasPart*) and *followedBy* property belong to the first type too. An example of a relation of the second type is *proves* relation, which occurs between a proof – the only valid element type here – and a statement the proof justifies. In our application, we follow the closed world assumption, and interpret range and domain of a property as constraints.

To add support of structural elements that are common for scientific publications on a wide range of fields, the ontology imports SALT Document Ontology (SDO) [1], an ontology of the rhetorical structure of scholarly publications. Specifically, it defines Section, Figure, and Table classes.

To enable making connections between structural elements and other objects contained by them and described elsewhere, e.g. mathematical named entities extracted from their text contents, we add a specific property – *mentions* – as follows: $mentions(x, y) \rightarrow (DocumentSegment(x) \vee Table(x) \vee Figure(x) \vee Section(x)) \wedge Thing(y)$. Document Segment class is the root of the Mocassin Ontology hierarchy.

The ontology also defines classes to represent several types of mathematical expressions – Mathematical Expression, Variable, and Formula. The datatype property *hasLatexSource* is defined for storing a LaTeX representation of the expression as a string. Yet, for the purpose of connecting formulas to mathematical named entities, there is *hasNotation* property in the ontology: $hasNotation(x, y) \rightarrow Thing(x) \wedge MathematicalExpression(y)$. For example, it enables us to state a fact that an empty set is denoted with $\emptyset$ in a text.

[7] http://cll.niimm.ksu.ru/ontologies/mocassin

[8] http://code.google.com/p/mocassin/

In addition, the ontology contains logical rules and cardinality axioms. One of the cardinality axioms states that every proof must justify at most one statement. An example logical rule is $dependsOn(x, y) \wedge hasPart(z, y) \rightarrow dependsOn(x, z)$, which e.g. we use to infer dependency between a proof and theorem, if the theorem contains an equation the proof depends on.

The ontology has been developed in OWL2/RDFS languages, which provide rich expressiveness, including cardinality and transitivity, and are also decidable theoretically and practically, for example, by using state-of-the-art reasoners like Pellet and FaCT++, or, to some extent, by in-house reasoners in modern RDF triple stores. A possible use case to exploit this feature is visualization of a dependency graph of theorems in related papers.

$OntoMath^{PRO}$. $OntoMath^{PRO}$ is an applied ontology for automatically processing professional mathematical articles in Russian and English[9]. The ontology defines the concepts commonly used in mathematics as well as the developing and not well established vocabulary (e.g. a term *Bitsadze-Samarsky problem* in differential equations). $OntoMath^{PRO}$ covers a wide range of fields of mathematics such as number theory, set theory, algebra, analysis, geometry, mathematical logic, discrete mathematics, theory of computation, differential equations, numerical analysis, probability theory, and statistics. Each class has a textual explanation, Russian and English labels including synonyms.

The terminological sources used during the development are classical textbooks, online resources like Wikipedia and Cambridge Mathematical Thesaurus, scholarly papers from the IVM journal, and personal experience of practicing mathematicians at Kazan Federal University. Thus, we expect that the ontology suffices the expert-level semantics on the fields.

In the ontology, one could distinguish two taxonomies with respect to ISA-relationship – a hierarchy of fields of mathematics and a hierarchy of mathematical knowledge objects. The first one is rather conventional and close to the related part of the Universal Decimal Classification[10]. The top level of the second taxonomy contains concepts of three types: i) basic metamathematical concepts, e.g. Set, Operator, Map, etc; ii) root elements of the concepts related to the particular fields of mathematics, e.g. Element of Probability Theory or Element of Numerical Analysis; iii) common scientific concepts: Problem, Method, Statement, Formula, etc. Due to multiple inheritance, the same class can be a sub-class of several classes. For example, Sparse Grid is a sub-class of both Formula and Element of Theory of Differential Equations.

$OntoMath^{PRO}$ defines three types of object properties:

- a directed relation between a mathematical knowledge object and a field of mathematics (*belongsTo*), e.g. Barycentric Coordinates *belongsTo* Metric Geometry;
- a directed relation of logical dependency between mathematical knowledge objects (*isDefinedBy*), e.g. Christoffel Symbol *isDefinedBy* Connectedness;

[9] `http://cll.niimm.ksu.ru/ontologies/mathematics`

[10] `http://www.udcc.org`

- a symmetric associative relation ("soft dependency") between mathematical knowledge objects (*seeAlso*), e.g. Chebyshev Iterative Method *seeAlso* Numerical Solution of Linear Equation Systems.

$OntoMath^{PRO}$ is developed in OWL-DL/RDFS languages. Numerically, $OntoMath^{PRO}$ contains 3 450 classes, 5 object properties, 3 630 subclass-of property instances, and 1 140 other property instances.

2.2 NLP Annotation

At this stage, we solve a standard task of annotating noun phrases in mathematical texts. In our approach, mathematical expressions are considered as valid parts of noun phrases. That is, they can be prefixes in hyphenated words, e.g. "σ-algebra".

Our solution relies on the "OntoIntegrator" [2], our tool for general-purpose linguistic analysis, which was adapted for peculiarities of mathematical texts, and currently supports only Russian language. It consecutively solves the standard linguistic tasks such as tokenization, sentence splitting, morphological analysis, and noun phrase extraction.

Morphological analysis is based on the Russian grammar dictionary extended with the vocabularies of general and domain-specific abbreviations, and parentheses. The result of the analysis is a grammar markup for words. In addition, homonyms are annotated with a fixed set of grammar annotations.

In Russian, noun phrases (NP) usually consists of the main noun, which we denote as NP.Head, and its left- and right modifiers (NP.Dependent). The relationship between the main noun and its dependent words is syntactical. Constructing noun phrases is described with the rules, which consider the definitive internal structure.

In our case, the main noun can be a noun, a pronominal noun, an abbreviation, a proper noun, a formula, or a citation reference. Among dependent words, there can be adjectives, pronominal adjectives, numerals, participles, adverbs, and prepositions. The noun phrase extraction method seeks noun phrases within a given sentence. Every noun phrase may contain exactly one or several segments, that is, word groups with certain characteristics. Within a segment, all the words are consistent according to their grammar characteristics. If there is more than one segment extracted in the noun phrase, the leftmost segment is considered as the main one and may have arbitrary grammatical characteristics – the case and the number. We assume that the other segments necessarily require the only form – the genitive case of a dependent noun. Gathering segments in a noun phrase is done from the right to the left. While annotating a noun phrase, the NP.Head is distinguished and normalized. The normalized form of the noun phrase is marked with a special "Form" annotation attribute. Math expressions are annotated with special "Math" tags. Further, the annotated noun phrases are used through ontology extraction.

Replacing the current NL processor with a module that supports noun phrase extraction as well as handling math symbols, abbreviations, formulas could switch the language to English as an example. A math-aware extension of the Stanford NLP parser[11] is a promising candidate.

2.3 Semantic Annotation

During this phase, we perform annotating documents in terms of the domain ontologies.

Mining the Logical Structure. Our method [3] receives NLP annotations and extracts structural elements according to the Mocassin and SALT SDO ontologies. This procedure falls into two tasks: (i) recognizing the types of structural elements; (ii) recognizing the semantic relations between them. As a result, the method outputs a semantic graph that contains, on the one hand, structural elements as nodes, each of which is assigned to a particular ontology class or marked "unrecognized" otherwise, and, on the other hand, ontology relation instances as edges. Aside from the object properties, each node has annotations corresponding to its title, text contents, and page numbers in the compiled PDF document. The information may be used in further applications for organizing a convenient navigation through document parts or highlighting more specific relevant search results.

Mathematical Named Entity Extraction. This task is a classification of extracted noun phrases as instances of $OntoMath^{PRO}$ classes, i.e., mathematical named entities (MNEs).

Our extraction method is uncertain and is based on a overlap of words in a noun phrase and ontology labels, respectively. We use Jaccard similarity coefficient as a confidence measure. Therefore, the method implies choosing the threshold value for filtering out wrong matchings. Specifically, given an NP and an ontology class, the confidence score C is defined according to the following rules:

- C ranges from 0 (minimal confidence) to 1 (maximal confidence);
- if the class label does not contain the main word of the NP (NP.Head), then $C = 0$;
- if the length (in terms of word count) of the class label is greater than the length of the NP, then $C = 0$;
- otherwise, C is equal to the Jaccard similarity coefficient for sets of words.

For example, the score between a noun phrase "Sobolev-like space" and a class Sobolev Space is equal to 2/3. On the contrary, the score between "number" and Fermat Number is equal to 0 because of the different lengths, or the score between "integral of the function of square-rooting" and Function of Square-rooting is also equal to 0 due to the different main words in the phrases.

[11] `http://nlp.stanford.edu/software/lex-parser.shtml`

Connecting MNEs to Formulas. We solve the following tasks within a single document:

- parsing mathematical expressions, i.e., detection of variables and seeking their occurrences in mathematical formulas;
- matching mathematical variables with noun phrases.

The method relies on “Math”, token, sentence and NP annotations. Regular expressions are used as a main tool during formula analysis. At the beginning, a formula is refined from special markup elements and redundant spaces. Then, the formula is split into separate elements, the delimiters are braces, brackets, operation symbols, punctuation marks, and spaces. The given elements are assigned to specific groups – markup keywords, indices, numbers etc. Each unclassified element is checked additionally on that its starting symbol is not a number, or if the element is in the set of Greek letters. As a result, all the mathematical expressions are divided into three groups – variables, formulas, and auxiliary fragments.

All the variables and formulas are stored in the index, which contains information about occurrences of variables in formulas. We provide an example that illustrates the semantics of such a relationship.

Example 1. Given a text fragment (translated from Russian):

Let $\overline{\alpha}$ be a second fundamental form of n-surface $\overline{M}$, $\overline{\nabla}$ is a Levi-Civita connection of the metric $\overline{g}$. Then, the equality holds:

$$\partial_X dfY - df\overline{\nabla}_X Y = \overline{\alpha}(X, Y).$$

The text fragment contains variables $\overline{\alpha}$, n, $\overline{M}$, $\overline{\nabla}$, $\overline{g}$, and a formula that uses $\overline{\alpha}$ and $\overline{\nabla}$ variables. Implicit bound variables X and Y are defined nowhere in the document, and, therefore, not included into the index. The instances of *hasPart* relation induced from the inclusions are depicted in Figure 1.

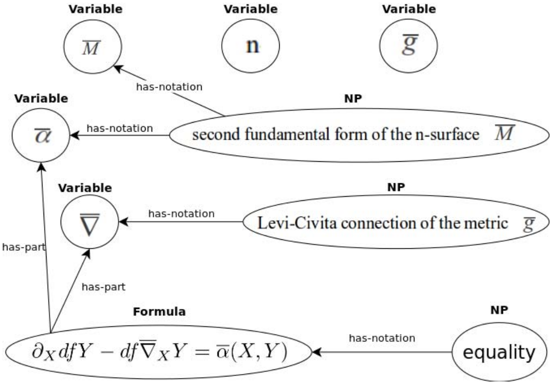

Fig. 1. A semantic graph to Example 1

The next step is connecting noun phrases to extracted variables and formulas. In principle, there are two possible cases of mutual positioning of a variable and an NP: first, an NP may contain a variable, and, second, the elements follow each other.

In the first case, an NP is the only candidate for linking. The simplest variation is if the NP contains a single main word. In Example 1, we have an NP "equality $", where $ is a formula in the NP. This means that the formula will be linked with this NP (see Figure 1). The complex variation is if an NP contains more than one word. In Example 1, variable $\overline{g}$ will not be linked with an NP "Levi-Civita connection of the metric $", because the main word is "Levi-Civita connection" and the variable is a complement here. Similarly, we ignore expression prefixes: in Example 1, "n" is left without linking, but a variable $\overline{M}$ will be linked with an NP "second fundamental form of the $-surface $".

In the second case, the key idea behind analysis is a concept of maximal feasible distance (MFD) in terms of symbol positions between "Math" and NP annotations in the text. For a given pair, we constrain MFD to be always less than the length of a sentence that contains both the annotations. The optimal value for MFD can be found empirically and, as our experiments have shown, the results are robust to its actual value. Finally, the method chooses the closest NP annotation to a given formula. Though, some cases are handled specifically, e.g. such popular text patterns as "[formula] – [NP]" with the dash in the middle.

2.4 Article Metadata Extraction

At this stage, we solve a task of extraction and conversion of article metadata as well as bibliographic references according to a standardized vocabulary. For this purpose, we choose AKT Portal Ontology[12]. Comparing to its alternatives, such as BIBO[13] and SWRC[14], the ontology covers the academic domain in more details and is widely used in existing LOD data sets. The extraction method:

- crawls a collection of documents and extracts from the headers the following information – title, author names, their affiliation, journal title, journal volume, and publication year;
- makes identifiers out of publication titles;
- post-processes bibliographies using the identifiers.
- prepares the article data for for serializing according to the AKT schema.

Article URIs are generated compatible with URLs on MathNet.Ru[15], a large online digital collection. In particular, it means that article URIs from our data set can be easily dereferenced in an Internet browser.

[12] http://www.aktors.org/ontology/

[13] http://bibliontology.com/specification

[14] http://ontoware.org/swrc

[15] http://mathnet.ru

2.5 Interlinking

We solve a task of interlinking the IVM data set with existing data sets in the LOD cloud. Essentially, the task is two-fold: first, aligning $OntoMath^{PRO}$ ontology with DBpedia, and, second, seeking duplicates in the AKT based LOD data sets. Our solution is not integrated with the processing units described above, and, unlike them, requires additional human efforts. We heavily use Silk application[16] for both the subtasks.

Aligning *OntoMath*^{PRO} with DBpedia. It is based on the following features:

- class and resource labels (*rdfs:label* property);
- links to Wikipedia – during the development of the ontology, some definitions were imported from Wikipedia and refer to it. We compare these references with *foaf:primaryTopic* and *rdfs:labels* property values in DBpedia.

For interlinking, we only use DBpedia resources that belong to the Mathematics category and its subcategories (e.g. Algebra, Geometry, Mathematical logic, Dynamical Systems) up to 5 levels with respect to *skos:broader* property. This is mainly caused by the shortcomings of Silk and DBpedia concerning handling and representing transitive properties[17].

After the linking has been accomplished, we generate triples connecting the classes of the $OntoMath^{PRO}$ with the resources from DBpedia by using *skos:closeMatch* property.

Seeking Duplicates in AKT Based Data Sets. We have investigated data sets based on the AKT schema. It turns out that the CORDIS data set[18] is the only appropriate one at the moment. Matching has been performed using information about organizations. In particular, *akt:name* and *akt:has-pretty-name* properties are used.

3 Implementation

In this section, we provide implementation details of our prototype.

The overall infrastructure of the publishing workflow is depicted in Figure 2. LaTeX is the only input document format supported by the prototype at the moment. Then, we use the arXMLiv tools [4] to convert LaTeX source files into a convenient XML representation. The NLP annotation module is based on the facilities of “OntoIntegrator” [2], a proprietary software tool for linguistic analysis of texts in Russian, developed by two of the authors. It supports XML as an

[16] `http://www4.wiwiss.fu-berlin.de/bizer/silk/`

[17] As we noticed during experiments, using the deeper levels may even lead to poor results. For example, there is a transitive chain between Topology and Alice in Wonderland categories!

[18] `http://cordis.rkbexplorer.com`

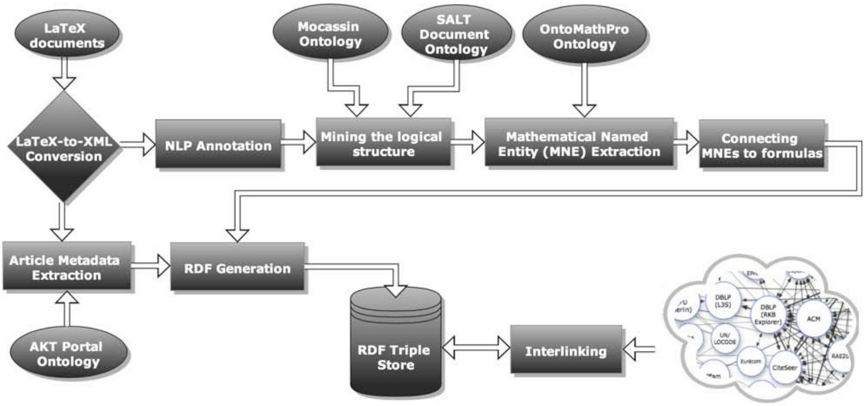

Fig. 2. Prototype Architecture

input/output format. The module for MNE extraction is implemented as a JS script[19]. It accepts XML files for processing and an OWL file of $OntoMath^{PRO}$ ontology. Relying on the NLP annotations, it complements XML files with additional attributes. The module for mining the logical structure is a part of the Mocassin project, an open source mathematical semantic search engine in Java. It processes XML documents using the GATE architecture[20] along with custom processing analyzers. The module for connecting MNEs to formulas is implemented as a GATE plugin[21]. Article metadata extraction is carried out by the special Bash scripts[22]. All the data from the previous steps flow together into the RDF generation unit to be converted to RDF. For the purpose, we use the OpenRDF Sesame library[23] written in Java, which prepares the RDF triple statements and saves them into the triple store, a Virtuoso Community Edition server instance[24]. Virtuoso is a high-performance RDBMS server with extensive RDF/SPARQL support and materialized OWL reasoner. Interlinking is supported by a custom SILK configuration script that uses a list of DBpedia categories related to mathematics[25].

4 Experiments and Evaluation

We have conducted an evaluation of some critical performing tasks to make sure that the extracted data are of high quality. In the section, we present the results and discuss possible failures of the developed methods.

[19] http://bit.ly/cll-mne-extraction
[20] http://gate.ac.uk/
[21] http://bit.ly/cll-gate-morph-formula
[22] http://bit.ly/cll-akt-metadata-extraction
[23] http://www.openrdf.org/
[24] http://sourceforge.net/projects/virtuoso/
[25] http://bit.ly/cll-interlinking

NLP Annotation. We randomly selected 24 documents out of the collection and checked 10 623 NLP annotations assigned by our method. It turns out that the sentence segmentation task is solved at high level of precision (98.9%) and recall (98.83%). Some errors occur, if the author places a period, the mark of the sentence end, inside a mathematical environment. Then, the method for NP extraction gives precision no less than 88%. The error types are as follows: missing fixed prepositional phrases (5%), missing right definition (2%), incomplete NP structure (2%) etc. The method can be improved by more deep syntactical analysis (e.g. of participial phrases) and considering more fixed phrases of mathematical vernacular.

MNE Extraction. While indexing the entire collection, the NLP subsystem outputs 330 462 NPs. The module of MNE extraction links 138 032 (41.7%) NPs to the ontology classes with non-zero confidence score values. After filtering documents on the field of mathematical analysis and removing duplicates, we had 16 300 unique MNE candidates, which were grouped into buckets according to observed confidence score values and were given to an expert in mathematical analysis for manual checking. Table 1 shows the distribution of recall/precision estimates depending on varying the confidence score threshold.

Table 1. Evaluation of MNE Extraction

Confidence score threshold	# of candidates	# of correct candidates	Recall	Precision
0.27	16 300	12 255	1.000	0.752
...	...	...	...	...
0.33	*15 964*	*12 117*	*0.989*	*0.759*
...	...	...	...	...
0.57	**2 470**	**2 426**	**0.198**	**0.982**
...	...	...	...	...
1.00	1 254	1 254	0.102	1.000

Finally, for constructing the RDF data set, we choose the following strategy, which accents the precision:

- for candidates with confidence score greater than 0.57, we generate "hard" relation instances (rdf:type), where every NP is treated as an individual of the linked ontology classes;
- for candidates, which confidence is between 0.33 and 0.57, we generated "soft" relation instances (skos:closeMatch).

Connecting MNEs to Formulas. We have studied the quality of connecting MNEs to formulas depending on the actual value of MFD. We manually select 8 documents from different fields of mathematics. The overall evaluation statistics is shown in Table 2.

Table 2. Statistics of Connecting MNEs to Formulas. TP means true positive, TN – true negative, FP – false positive, FN – false negative.

MFD	TP, %	TN, %	FP, %	FN, %	Accuracy, %
15	36.3	30.5	23.9	9.3	66.8
20	42.3	25.5	25.7	6.5	**67.8**
25	41.0	20.7	23.0	15.3	61.7

In total, there are 1 247 mathematical expressions and 1 357 NPs. The optimal value of MFD is equal to 20. It gives 67.8% in accuracy. We emphasize that this value absorbs the errors of NLP annotation and some misspellings in the texts, e.g. replacing dashes with hyphens. Additionally, varying MFD in a range between 15 and 40 has 64.0% mean accuracy with 2.7% standard deviation, which supports our claim that choosing MFD for our method is not so critical in practice. Among the necessary improvements of the formula linking method, there are a special handling of equation groups and accurately filtering of mathematical expressions.

Aligning *OntoMath^{PRO}* with DBpedia. The alignment has resulted in 947 connections with 907 *OntoMath^{PRO}* classes (some classes were linked with several DBpedia resources). Thus, the ontology coverage is about 27%. The manual assessment gave a precision estimate of 95%. The errors come from the following issues:

- inconsistencies in interwiki linking in Wikipedia: ontomathpro:Sum of the Series ≠ dbpedia:Convergence_tests
- an issue with homonymous concepts and categories in DBpedia: ontomathpro:Ideal ≠ dbpedia:Ideal_ethics, the latter occurs in the transitive chain of categories: Philosophy of Life → Life → Universe → Astronomical dynamical systems → Dynamical Systems.

Seeking Duplicates in AKT Based Data Sets. The module returns only 91 correct and 13 wrong duplicates of organizations from the CORDIS data set. It means that there is no much overlap between these data sets. The module failed to find duplicates of all the types in the DBLP data set due to the absence of such data (in case of organizations) and retrieving limits of its SPARQL endpoint.

5 IVM Data Set: Statistics and Use Cases

The resulting RDF data set[26] contains 854 284 triples including the descriptions of 43 963 variables, 17 397 formulas, 4 190 theorems, 3 035 proofs, 2 356 lemmas, 1

[26] The data set can be accessed via a SPARQL endpoint – `http://cll.niimm.ksu.ru:8890/sparql-auth`, the endpoint is secured, please email the authors to get access to it.

015 definitions and other mathematical entities indexed. Below, we demonstrate several use cases using SPARQL queries to illustrate possible applications.

Use Case 1. Let us assume, we would like to find articles with theorems about finite groups.

```
PREFIX moc: <http://cll.niimm.ksu.ru/ontologies/mocassin#>
PREFIX math: <http://cll.niimm.ksu.ru/ontologies/mathematics#>
SELECT ?article WHERE {
?article moc:hasSegment ?theorem .
?theorem moc:mentions ?entity; a moc:Theorem .
?entity a math:E2183
}
```

In this query, we use Theorem, a Mocassin ontology class, and its properties *hasSegment* and *mentions* along with a class Finite Group (E2183) from the $OntoMath^{PRO}$ ontology.

Use Case 2. The next query is to determine the fields a particular article belongs to.

```
define input:inference
 "http://cll.niimm.ksu.ru/ontologies/mathematics/rules"
PREFIX moc: <http://cll.niimm.ksu.ru/ontologies/mocassin#>
PREFIX math: <http://cll.niimm.ksu.ru/ontologies/mathematics#>
SELECT ?field ?label WHERE {
<http://mathnet.ru/ivm327> moc:hasSegment _:a .
_:a moc:mentions _:b . _:b a _:c .
_:c owl:equivalentClass _:d . _:d owl:onProperty math:P3 ;
owl:allValuesFrom ?field . ?field rdfs:label ?label
} GROUP BY ?field
```

A URI *http://mathnet.ru/ivm327* maps to an article URL on MathNet.Ru. A *math:P3* stands for the inverse property for *belongsTo*. The query outputs classes that represent some mathematical domains, such as Discrete Mathematics, Theory of Computation, Mathematical analysis, and Probability Theory, that are relevant to the given article.

Use Case 3. Finally, for Empty Set, a certain DBpedia concept, we would like to determine its notations occurred in the articles.

```
PREFIX moc: <http://cll.niimm.ksu.ru/ontologies/mocassin#>
SELECT ?latexSource from  <http://cll.niimm.ksu.ru/ivm> WHERE {
?class skos:closeMatch dbpedia:Empty_set .
?notation moc:hasLatexSource ?latexSource .
?entity moc:hasNotation ?notation;
a ?class .
}
```

This query may help to choose the proper notation for a beginning researcher in mathematics. On our data set, the search results are as follows: ω, $\emptyset$, $\omega \in \mathcal{D}$.

6 Related Work

Mathematical knowledge representation, as a field, has its own rich history. There have been developed various models and tools to formalize different aspects of the mathematical domain. For example, domain-specific languages, such as MathLang [5] and OMDoc [6], give opportunities to build semantically enriched models of a mathematical document and natively support representing logical structure elements like theorems or definitions. However, creating such highly formalized mathematical documents is still a laborious process. The paper [7] presented an approach to author math lecture notes with specific sTEX macro package. This work primarily focuses on mathematical formulas and elements of the logical structure and appears to be the first work aiming to fit mathematical texts and LOD together.

Historically, the Bourbaki group's series of books was the first ever attempt to create an ontology of mathematical knowledge rooted in G. Cantor's set theory. Their seminal work establishes a conceptual framework for defining mathematical entities organized in different fields. There have been a few applied domain models developed in the digital era. For example, [8] presents a formal ontology of mathematics for engineers that covers abstract algebra and metrology. Cambridge Mathematical Thesaurus[27] contains a taxonomy of about 4 500 entities connected with logical dependency and associative relationships. This resource covers terms from the undergraduate level mathematics. Next, relying on Wikipedia, Encyclopedia of Science, and the engaged research community, the ScienceWISE project ontology [9] gives over 2 500 mathematical definitions connected with ISA-, part-whole, associative, and importance relationships. The project focuses on achieving a consensus of opinion among mathematicians about given definitions. In the context of modeling mathematical concepts with the help of Semantic Web tools, we would like to note a recent adaptation of Mathematics Subject Classification[28] using SKOS as a linked data set [10]. From this perspective, our $OntoMath^{PRO}$ ontology overlaps with this data set in case of modeling hierarchy of fields, but it is significantly richer for representing mathematical named entities.

Impressive advances in ontology extraction have been achieved across many domains. However, before our work, only a few projects have applied ontology based NLP techniques for scholarly papers in mathematics. The mArachna project [11] focuses on extracting ontologies combining the mathematical knowledge and information about the document structure. However, a comparison of mArachna with our work is problematic, because the project aims for German, and its authors do not provide many details about the specification of the structure, and implementation of the entity extraction techniques to enable a replication of their results. Next, linguistic modules of the arXMLiv project [4] are intended for resolving ambiguities in mathematical notation for texts in English. We are going to conduct a comparative analysis with this work after adding support of English language to our NLP annotation module.

[27] http://bit.ly/cambridge-math-thesaurus

[28] www.ams.org/msc/

Most research insights and tasks, the solutions of which we described here, were stated in [12]. To our knowledge, the present work is first to extract a Linked Data representation of academic papers in mathematics using not only their metadata, but also the text contents, in an automatic way.

7 Conclusion and Outlook

We present a platform prototype for mining a structured standardized representation of scholarly papers in mathematics. The platform aims for automatic publication their contents as well as metadata in the format of LOD-compliant data. The tool has been applied on a collection of over 1 300 mathematical publications to demonstrate feasibility of the solution. We report on evaluation of the most important tasks solved during the development. Finally, we provide several use cases to illustrate utility of the published data. As a future work, we are aiming to integrate all the modules into a full-fledged toolkit, add support of English language, and extend our approach to other natural science domains, such as physics, chemistry, and biology.

Acknowledgments. This work was supported by the Ministry of Education and Science of the Russian Federation in the scope of the Federal Framework Program – Priority Research and Development in Science and Technology Complex of Russia in 2007-2012 (State contract No. 07.524.11.4005), the Russian Foundation for Basic Research (grant No. 11-07-00507-a), and Kazan Federal University (grant No. 0221 09011). The authors would like to thank Valery Solovyev, Anna Kayumova, Evgeny Lipachev, Ilgiz Kayumov, Pyotr Ivanshin, Elena Utkina, and Marjan Matvejchuk, who have contributed a lot to the $OntoMath^{PRO}$ ontology as well as the conducted evaluation. The authors are also very grateful to Christoph Lange and three anonymous reviewers for their useful suggestions.

References

1. Groza, T., Handschuh, S., Möller, K., Decker, S.: SALT – Semantically Annotated Latex for Scientific Publications. In: Franconi, E., Kifer, M., May, W. (eds.) ESWC 2007. LNCS, vol. 4519, pp. 518–532. Springer, Heidelberg (2007)
2. Nevzorova, O., Nevzorov, V.: The Development Support System "OntoIntegrator" for Linguistic Applications. In: International Book Series "Information Science and Computing", vol. 3(13), pp. 78–84. ITHEA, Rzeszow-Sofia (2009)
3. Solovyev, V., Zhiltsov, N.: Logical Structure Analysis of Scientific Publications in Mathematics. In: Proceedings of the International Conference on Web Intelligence, Mining and Semantics (WIMS 2011), pp. 21:1–21:9. ACM (2011)
4. Stamerjohanns, H., Kohlhase, M., Ginev, D.: Transforming Large Collections of Scientific Publications to XML. In: Mathematics in Computer Science, vol. 3, pp. 299–307. Springer (2010)
5. Kamareddine, F., Wells, J.B.: Computerizing mathematical text with MathLang. Electr. Notes Theor. Comput. Sci., 5–30 (2008)

6. Kohlhase, M.: OMDoc – An Open Markup Format for Mathematical Documents [Version 1.2]. Springer (2006)
7. David, C., Kohlhase, M., Lange, C., Rabe, F., Zhiltsov, N., Zholudev, V.: Publishing Math Lecture Notes as Linked Data. In: Aroyo, L., Antoniou, G., Hyvönen, E., ten Teije, A., Stuckenschmidt, H., Cabral, L., Tudorache, T. (eds.) ESWC 2010, Part II. LNCS, vol. 6089, pp. 370–375. Springer, Heidelberg (2010)
8. Gruber, T., Olsen, G.: An Ontology for Engineering Mathematics. In: KR 1994, pp. 258–269 (1994)
9. Aberer, K., Boyarsky, A., Cudr-Mauroux, P., Demartini, G., Ruchayskiy, O.: ScienceWISE: A Web-based Interactive Semantic Platform for Scientific Collaboration. In: 10th International Semantic Web Conference (ISWC 2011 - Demo) (2011)
10. Lange, C., Ion, P., Dimou, A., Bratsas, C., Sperber, W., Kohlhase, M., Antoniou, I.: Bringing Mathematics to the Web of Data: the Case of the Mathematics Subject Classification. In: Simperl, E., Cimiano, P., Polleres, A., Corcho, O., Presutti, V. (eds.) ESWC 2012. LNCS, vol. 7295, pp. 763–777. Springer, Heidelberg (2012)
11. Jeschke, S., Natho, N., Rittau, S., Wilke, M.: mArachna: Automaticall Extracting Ontologies from Mathematical Natural Language Texts. In: IMECS, pp. 958–963 (2007)
12. Birialtsev, V., Elizarov, A., Zhiltsov, N., Ivanov, V., Nevzorova, O., Solovyev, V.: Ontology Based Semantic Search Model for the Collections of Mathematical Documents. In: Proceedings of XII All-Russian Science Conference RCDL, pp. 296–300 (2010) (in Russian)