An Architecture for Linguistic and Semantic Analysis on the ARXMLIV Corpus

D. Ginev, C. Jucovschi, S. Anca, M. Grigore, C. David, M. Kohlhase
http://kwarc.info/projects/lamapun/
Jacobs University Bremen, Germany

Abstract:

The ARXMLIV corpus is a remarkable collection of text containing scientific mathematical discourse. With more than half a million documents, it is an ambitious target for large scale linguistic and semantic analysis, requiring a generalized and distributed approach. In this paper we implement an architecture which solves and automates the issues of knowledge representation and knowledge management, providing an abstraction layer for distributed development of semantic analysis tools. Furthermore, we enable document interaction and visualization and present current implementations of semantic tools and follow-up applications using this architecture.

We identify five different stages, or purposes, which such architecture needs to address, encapsulating each in an independent module. These stages are determined by the different properties of the document formats used, as well as the state of processing and linguistic enrichment introduced so far. We discuss the need of migration between XML representations and the challenges it would pose on our system, revealing the benefits and trade-off of each format we employ.

In the heart of the architecture lies the Semantic Blackboard module. The Semantic Blackboard comprises a system based on a centralized RDF database which can facilitate distributed corpus analysis of arbitrary applications, or analysis modules. This is achieved by providing a document abstraction layer and a mechanism for storing, reusing and communicating results via RDF stand-off annotations deposited in the central database.

Achieving a properly encapsulated and automated pipeline from the input corpus document to a semantically enriched output in a state-of-the-art representation is the task of the Preprocessing, Semantic Result and Output Generation modules. Each of them addresses the task of format migration and enhances the document for further semantic enrichment or aggregation. The fifth module, targeting Visualization and Feedback, enables user interaction and display of different stages of processing.

The overall architecture purpose is to facilitate the development and execution of semantic analysis tools for the ARXMLIV corpus, automating the migration of knowledge representation and establishing a complete pipeline to both a presentation and content enriched document representation.

Additionally, we present three applications based on this architecture. Mathematical Formula Disambiguation (MFD) embodies an analysis module that uses heuristic pattern matching to disambiguate symbol and structure semantics. Context Based Formula Understanding (CBFU) is another Semantic Blackboard module which in turn focuses on establishing context relationships between symbols, helping to disambiguate their semantics. We also present the Applicable Theorem Search (ATS) system, a follow-up application that performs search functions, retrieving theorem preconditions for the user.

1 Introduction

The "Language and Mathematics Processing and Understanding" (LAMAPUN) project is a recent effort of the KWARC research group at Jacobs University. We investigate semantic enrichment, structural semantics and ambiguity resolution in mathematical corpora. Long term goals include applications in areas such as Information Retrieval, Document Clustering, Management of Change and Verification. The architecture described in this paper provides a workbench for various analysis tools on large corpora. Since different representations of the same documents allow different types of analysis, our architecture automates the transition in between the different formats, allowing integration of multi-purpose tools and establishing a complete input/output pipeline. It is based on state-of-the art Semantic Web services, XML formats, as well as Computational Semantics and Computational Linguistics tools and techniques.

The LAMAPUN work focuses on the ARXMLIV[SK08, arX09b] corpus and is based on the contributions of a group of Jacobs University graduate students, making it a long-term, distributed effort of alternating developers. Two of the most fundamental components needed for any large-scale analysis of informal mathematical discourse are a sizable collection of documents and a comprehensive analysis framework. The ARXMLIV corpus is an XML representation of Cornell's pre-print ARXIV [arX09a] of scientific articles, with more than half a million converted papers in 37 scientific subfields. Even though the XML representation is much more convenient for processing tasks than the LaTeX sources, it still contains a lot of information (such as styling tags) that will not be required for many semantic processing tasks but which nevertheless should not be removed entirely. Our analysis framework gives a comprehensive and high-level abstract layer over the data.

The converted nature of the ARXMLIV corpus allows great customizability of its documents, but at the price of a rather involved low-level interaction. Hence, there is a need for a stable backbone which utilizes the power behind the corpus conversion mechanism and automates the different conversion and analysis stages. Furthermore, different applications on top of the corpus demand different emphases on knowledge representation, state of processing and inferred structure. The architecture needs to encapsulate the different representation stages. It must also allow easy interaction with external tools, motivating a modular design of stand-alone components, each dealing with a particular intermediate representation and state of the document data.

Existing general-purpose annotation frameworks, such as GATE [CMBT02], Heart of Gold (HoG) [Sch05] or UIMA [FL04], already provide parts of the functionality we need for our system. They focus on providing a setting for creating analysis pipelines, oriented towards linguistic analysis and information extraction. However, none of them is ready for direct deployment on a large body of XML documents, or can be easily extended to support various knowledge representations. In the context of analyzing the ARXMLIV corpus and the Semantic Web in general, an intuitive and standardized support of hypertext data is vital for a successful and efficient application development and deployment. The LAMAPUN work already focuses on understanding mathematical discourse, demanding support for different XML formats for mathematics and an accessible document representation for our semantic analysis tools. We contribute to the current state-of-the-art with a

framework that is quickly deployable, representation-aware, enables an intuitive application development and natively supports Semantic Web mathematics.

In order to illustrate the operation of all components of the framework, the running example of the LaTeX source of a document containing elementary trigonometry will be used throughout the paper. The document contains a sloppily written sentence about the area of a triangle:

```
If $T$ is a scalene triangle with sides
$a, b, c,\ then\ Area(T)=\frac{1}{2} ab\cdot sin(C)$.
```

The resulting compiled LaTeX output of this sentence will be normal (see Appendix 5.1.2 [app]), but the mixture of text and formulae in LaTeX math mode is semantically flawed and leads to processing errors in a conversion to XML. Our architecture sets out to correct these mistakes and present a semantically correct output.

An introduction into the motivation behind the current framework and related work has been given in the current Section 1. The backbone of the architecture, the central semantic blackboard and the modules facilitating it are described in Section 2. The semantic analysis modules which operate on the blackboard are outlined in Section 3 and the conclusion makes up Section 4.

2 The architecture

We implement a modular architecture that provides a stand-off RDF abstraction of the source documents and automates the migration in between the underlying XML representations, which are essential for the arXMLiv corpus with an outlook to added-on services. The modules encapsulate *preprocessing*, a "*Semantic Blackboard*" for distributed semantic analysis, a representation of the *semantic results*, appropriate *generation of output formats*, as well as *user interaction and visualization*, as outlined in Fig.1. We proceed with a detailed review of the system components.

2.1 The LaTeXML Backbone

The LaTeX to XML conversion that effectively created the arXMLiv corpus, has been performed by Bruce R. Miller's LaTeXML system [Mil07]. LaTeXML is a highly customizable tool released in the Public Domain, which supports the conversion from LaTeX to a custom XML format. Consecutively, its postprocessor, LaTeXMLpost, can drive the conversion to XHTML and potentially any other representation via a customized XSLT style sheet. The chief difference between LaTeXML's representations resides in the structural semantics of mathematical fragments. LaTeXML is currently able to generate both Presentation and Content MathML [ABC+03], as well as an OpenMath [BCC+04] representation of mathematics from its intermediate XMath format. In this paper, each representation of interest will be distinguished via an appropriate file type of the document:

- **.noparse.xml** - Contains a representation linguistically equivalent to the LaTeX source document. Mathematical formulas are represented via a linear sequence of atomic components, i.e. tokens, without creating any semantic parse tree (unless explicitly stated otherwise in the LaTeX source). This custom LaTeXML XML format, as shown in Appendix 5.1.3[app] is achieved by an explicit demand on LaTeXML to not parse any mathematical structures beyond the atomic token level.

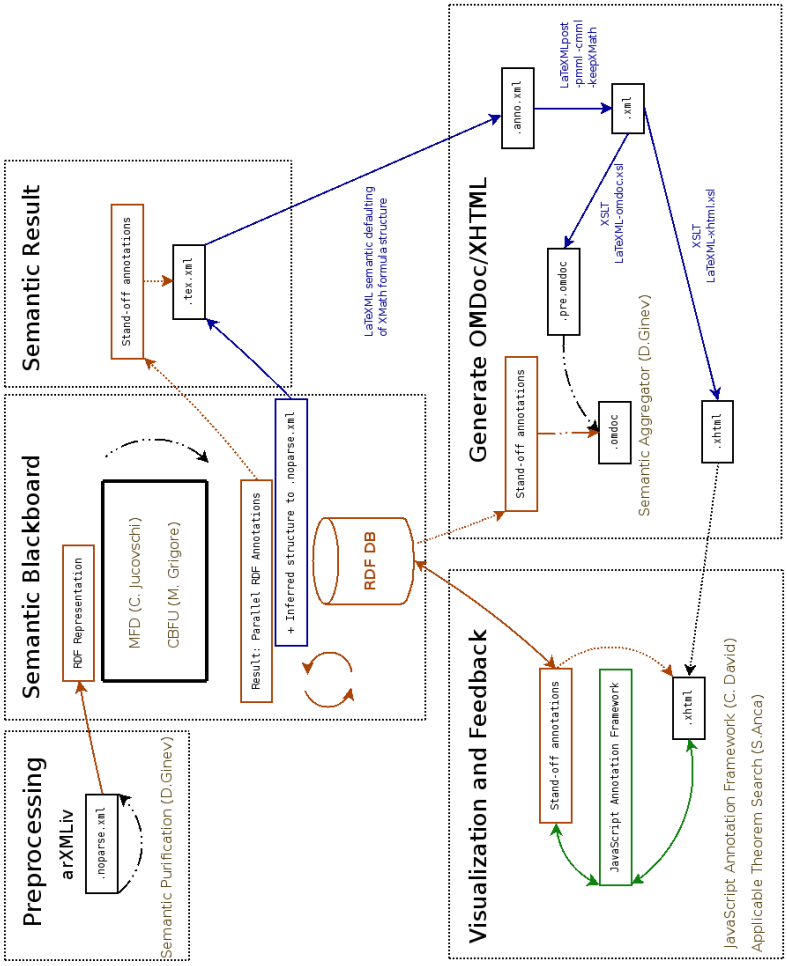

Figure 1: A high-end overview of an ARXMLIV analysis architecture

- **.tex.xml** - Equivalent to **.noparse.xml** with the exception of parsing mathematical fragments and creating a formula derivation tree. The semantics of the mathematics is changed, as the formula structure is achieved via a predefined grammar and the use of simple heuristics, which implies defaulting of both symbol and structural semantics. This often leads to wrong semantics of the respective augmented fragment. However, the only way to assure a valid conversion of the math fragments to a content representation is via such treatments. This is the case since otherwise any subsequent processor will have to deal with partially linearized mathematics, leftover from the **.noparse.xml** predecessor. Such structure is clearly ambiguous and malformed, hence needing further analysis to be resolved. Still, **.tex.xml** is generated solely to enable postprocessing, which is an analysis-free stage. As a framework

default, creating a full derivation tree guarantees a successful pass through the different representation conversions, producing valid XML output. An example, also purified as explained in Section 2.2, can be found in Appendix 5.1.5[app].

- **.xml** - Provides additional MATHML and/or OPENMATH representation of the math fragments, optionally using parallel markup and keeping the original XMATH. The rest of the XML DOM is still the same as of the previous **.tex.xml** and **.noparse.xml**.
- **.xhtml** - Achieved via a native XSLT style sheet which transforms the **.xml** into XHTML. An example can be observed in Appendix 5.1.6 [app].

As LATEXML already facilitates the transition between the different intermediate stages, incorporating it as a backbone of the architecture is an obvious choice. The LATEXML developers have contributed to the effort in a fruitful collaboration which gave further power to LATEXML's DOM and postprocessing module. Integrating the different representations together, could now be achieved almost out of the box. A set of low-level Perl scripts and customized XSLT stylesheets manage the consistent transition between intermediate formats, accommodating their proper interpretation by LATEXML and LATEXMLPOST and assuring the preservation of the XML hooks which would be used for stand-off annotations. Remarkably, most of the processing is performed by already existing capabilities of the LATEXML software, which makes the architecture design lighter and provides a very intuitive conversion pipeline.

2.2 Preprocessing Module

The ARXIV corpus contains almost twenty years of good and bad practice of writing TEX and LATEX documents. However, as TEX/LATEX is presentation-oriented, people have only cared whether the result "looks right", ignoring any semantic implications of their work. Furthermore, TEX/LATEX gives the user abundant possibilities to achieve the desired look and feel of their document. However, some of these are "semantically adequate" while some are not. The preprocessing module tries its best to convert the latter into the former from the ARXMLIV perspective, as the XML translation of ARXIV propagates these issues. Currently, we focus on "purifying" the semantics of the mathematical fragments in the documents, which we describe in detail below.

Based on the **.noparse.xml** document representation, the primary goal of the purification procedure is to enhance the existing XML modularity of the natural language and mathematics. In the intermediate **.noparse.xml** format, a mathematical fragment is encapsulated into an XMATH element, while natural language resides in regular TEXT elements. Due to the LATEX origin of the documents, however, we often have semantic "noise" in this modularity.

TEX/LATEX distinguishes two processing modes: **text mode** and **math mode**. From the perspective of in-sentence, or "natural language near", linguistic modality. However, LATEX is originally presentation oriented, having the purpose of "typesetting ink on paper", which could easily lead to semantically void structures, originally motivated by purely presentational reasons.
The most basic examples for switching the specific mode only for the purpose of a nicer appearance would be "1^{st}" and "{\bf x} - {\bf y}" (without $).

This problem is bidirectional - mathematics in text mode as well as text in math mode can both be observed in the ARXMLIV corpus. The former is relatively easy to spot and adjust, as LaTeX uses ASCII. Any symbolically involved mathematical construct would require specific typesetting techniques that are only accessible in LaTeX's math mode. Hence, the primary purpose of this direction of purification is to extend existing math segments by nearby adjacent scalars, simple variables and ASCII operators, incorrectly given in text mode. In order to detect the constructs of interest we use simple, yet unambiguous, heuristics that recognize simple mathematical discourse in plain text.

Spotting natural language in XMATH is a task of slightly greater difficulty. A word in math mode would get converted by LATEXML into an atomic token sequence of its letters and is no longer immediately recognizable. Hence, we use WORDNET [Fel98], a comprehensive list of LaTeX symbols and a sieved statistics of the top occurrences of non-WORDNET words in the corpus, as a partial recognition heuristic. This already achieves an auxiliary purpose of spotting complex structural tokens, something that LATEXML is currently not supporting natively for constructs without additional markup. For example, $\$last \neq first\$$ would be interpreted as a sequence of 4 tokens, representing $last$, an operator token representing $\neq$, followed by 5 tokens representing $first$. This is bizarre from the perspective of a human reviewer, but is a good example of the linguistic neutrality a **.noparse.xml** document representation produces. Our purification procedure succeeds in detecting any unambiguous complex token that is named after a natural language construct and proceeds with detecting those of them which are semantically not a part of the math construct. Currently, a single heuristic is employed, which tests for spacing around any WORDNET-derived token and if such spacing exists it transfers the segment back to the text modality.

Another two auxiliary purposes that the semantic purification achieves, are recognizing and enhancing mistyped LaTeX operators (cos vs $\cos$) and merging adjacent XML math blocks. While the former enhances symbol semantics, the latter restores the original formula context scope, achieving broader analysis potential for subsequent formula analyzer modules. As a result, the purified **.noparse.xml** representation achieves a truly semantic modularity between mathematics and natural language, enhanced token scopes and token semantics, as well as expanded formula contexts. This clears the road for the central processing modules, abolishing most of the harmful "semantic noise". The result of purifying our running example is shown in Appendix 5.1.4[app].

2.3 Semantic Blackboard Module

The vision behind a "Semantic Blackboard" is essentially to allow distributed corpus analysis by providing an accessible document representation and the means to store and later use the inferred semantic information from all active analyzers. This module is the core of the architecture, coordinating the analysis process and acting as an interface between the different semantic applications and the rest of the architecture. Below, we describe the general principles behind the design and the implementation of the Semantic Blackboard and show two fundamental analysis tools which build on top of its implementation.

2.3.1 Knowledge Representation

The idea of introducing semantics into a very large online corpus like the ARXIV is very ambitious. Clearly, it is a long term project and hopefully more research groups will join our efforts (or vice versa) to accomplish this task. In order to ensure a long life to this project we have to use a knowledge representation system that is easy to understand, use, extend, distribute and share. We want other researchers to quickly grasp the basic concepts and spend their time on hunting for new semantic information. We do not want to limit the users in using a certain tool, hence supporting software based on the knowledge representation of choice should already exist. Also, we want to have a system which gives the possibility to fetch only a subset of the available original data and inferred semantics, process it, and then push new semantic data back to a public database. This will make the system faster and more robust to failures, as each user can work with local data. For these reasons we chose to adhere to the standards and best practices of the Semantic Web.

Consequently, we chose a stand-off annotation system. Through that, we avoid having conflicts in between the efforts of different researchers, the resulting system is faster and more stable, and is also easier to share. Also, as prescribed by best practices from Semantic Web, we represent knowledge in the subject-predicate-object paradigm supported by the W3C Resource Description Framework (RDF) [LSWC98]. This will make sharing new semantics easier and tool independent. These decisions represent the only imposed limitations for describing semantics.

We use the openRDF database Sesame [BKH01] to store semantic annotations. It provides fast storage, SPARQL [PS08] query support as well as a friendly user interface (not of least importance). Having a query language enhances developer experience considerably. Firstly, fetching some data does not mean writing yet another program. Secondly, one can specifically download/work with the data from the server in which he/she is interested in. Also, enabling the use of SPARQL query language is a step forward towards more flexibility in choosing the underlying storage database and hence should be adhered to whenever possible.

As we base our work on the intermediate **.noparse.xml** stage in the corpus conversion which is not publicly accessible, we are compelled to keep this data in the public RDF database as subject-predicate-object statements. Storing the corpus documents in this way might sound suboptimal, however it gives us the option of hiding the complexity of the XML representation by ignoring, for example, formatting tags. This also means that we can introduce them back into the database on demand. Another gain is the expressiveness to group objects of the same type. For example we are free to define a *followed* relationship between consecutive words, even if they do not appear consecutively in the document.

2.4 Semantic Result Module

The Semantic Result Module is a static module that preserves the semantic analysis results in their original stand-off configuration. The final state of the stand-off annotations produced by the Semantic Blackboard analyzers, after all processing has taken place, is considered the analysis result. The primary **.noparse.xml** document which was the subject of analysis is enhanced with unambiguous and consistent inferred structural semantics,

ideally becoming a correct version of LaTeXML's **.tex.xml** representation and in turn changing its own extension to **.tex.xml**.

2.5 Output Generation Module

Over the course of the architecture development, Bruce R. Miller has assisted us in making LaTeXML customizable enough to support the specific needs of the representation migrations for the different architecture modules. The main help of LaTeXML's functionality is in the Output Generation Module, starting with the conversion from **.tex.xml** to **.xml**. At this step we have the option to add parallel MathML (towards **.xhtml**), OpenMath (towards **.omdoc**) and XMath (for annotation visualization and feedback), translating the mathematical fragments into state-of-the-art representations, targeting both human- and computer-oriented applications. Currently there are two supported output formats from this math-enhanced intermediate **.xml** representation, respectively a presentation oriented one and a content oriented one.

As XHTML is the standard for hypertext documents, it is an obvious choice for a presentation-oriented representation. It allows embedding mathematics via the MathML format, which in turn allows for accommodating any alternative representation via annotation-xml elements. In our workflow, we use the global "xml:id" attributes of the ⟨Math⟩ elements as annotation hooks throughout all XML representations, which makes the stand-off annotation process more generalized and maintainable. Preserving these hooks during the **.xml** to **.xhtml** conversion requires a slight deviation from the native LaTeXML to XHTML style sheet, which is the only change we need to introduce to the existing LaTeXML procedure, giving us the workflow to XHTML at almost zero cost. This facilitates a connection between the XHTML representation and the stand-off annotation database, satisfying the prerequisites for the successive Interaction and Visualization Module.

OMDoc [Koh06] is a state-of-the-art content representation format for mathematical documents and is the second supported output by the architecture. As LaTeXML does not directly support an OMDoc representation at the moment, we had to develop our own LaTeXML to OMDoc style sheet supporting the transition. Furthermore, as the OMDoc format is capable of expressing semantics on all document levels, it is a target for the aggregation of the inferred stand-off content. This is achieved via a semantic aggregator which performs consistency checks, resolves conflicts and avoids redundancy on the database annotations, embedding the aggregated results into the OMDoc output. The aggregation procedure is still work in progress and would employ a semantic analysis of its own.

2.6 Interaction and Visualization Module

Usually, people are taken out of the "equation" of a project regarding the annotation of big collections of mathematical documents like the ones found in the arXMLiv corpus. This happens because the sheer enormity of the target data implies a very time consuming annotation process. At the same time, the available resources, such as frameworks on which the annotation can be done, are scarce. However, such a framework would eventually enable people to share their knowledge with a computer (in this case formalized in

an RDF database) and could bring numerous advantages to any disambiguation or supervised process in general, as well as have an immediate impact on the current LaMaPUn project.

One of the most basic applications of human interaction in the field of language processing is "ground truth"-ing. For example, in the equation $f(x) = x + 2$, a human would easily assume that f is a function, x is the variable and that $+$ is the summation operator, only from a simple observation. If we further assume that the people reviewing the corpus articles are acquainted with the field of the article and introduce correct annotations, their involvement would considerably contribute to the disambiguation process and provide reliable data for learning approaches and work on related documents.

Therefore, any semantic analysis module which deals with structural semantics, semi-supervised learning or disambiguation could make use of the existing data (in the form of stand-off RDF annotations) and statistics, and thus improve the disambiguation process to provide more conclusive results. Also, the existence of such a framework will benefit the developers of other analysis modules by providing early feedback regarding the derived semantics of documents annotated using this tool. In other words, this would allow the user to visualize the annotations, to see what the analysis derives from the data and, if needed, help debug the respective tool. Also in the field of visualization, this framework can benefit the development of the LaTeXML software, by offering feedback with regard to possible conversion errors of LaTeXMLpost (a task already being undertaken inside the arXMLiv group at Jacobs University as it is of central relevance to the quality of the corpus articles).

Having already mentioned its benefits, the process of mathematical annotation is a long and tedious one, making it completely unattractive to non-specialists. A solution to making the process more appealing is a web-based design of an annotation framework that would read data from the common knowledge RDF database and, consequently, by interacting with the user, decide on the meaning of certain parts of the formulas, storing them as new and improved annotations back in the database. This procedure should be realized in a pleasant and interesting way for the user, potentially being competitive and stimulating, in order to attract an ever larger user base.

The actual implementation (see Appendix 5.2 [app]) of the project relies on the capability of the Mozilla Firefox browser to parse and correctly display Presentation MathML. We make use of the **.xhtml** representation generated by the Output Module which allows us to immediately provide online document interaction. The next step in the development is utilizing the GreaseMonkey [Gre09] extension for Firefox, which allows users to customize the way web pages look and function. This method of customization is already widely used and users have developed tools for interaction with websites so that the user would enjoy a better web experience. First of all, this extension allows deep HTML modification in appearance, by allowing user created scripts to modify the original HTML code of the document and add certain types of controls. Secondly, changes could also be functional. Via implementing JavaScript functions, the script may invoke the refresh of a page at a certain time or other behavior that enhances the experience of the users. Having this setup, the single major add-on left to implement is the safe and correct communication with the RDF database. The implementation of this module is currently under development

and its client-side approach promises a distributed, secure and efficient user interaction with the semantic results of the Semantic Blackboard.

3 Linguistic Analysis modules

In this section we will present two linguistic analysis modules and an application feeding on the analysis result. This serves as a description of our experiences with the architecture as well as a template for further analysis and application modules. We invite external users to collaborate with us on the conversion of the ARXMLIV corpus by contributing such modules.

3.1 Context Based Formula Understanding

This approach deals with context-based ambiguities that often occur in mathematical notations. It is well known that experienced readers are able to find the proper reading of a mathematical formula by making use of both their intuition and the formula context. Because of the vast number of documents stored in the ARXMLIV corpus, a necessary goal is to minimize the amount of disambiguation work left to readers as much as possible. There are significantly many situations in which solving the ambiguity requires extra-syntactic information. A typical one, is the situation in which the reader deals with the ambiguity by means of context. For example, when the symbol ω occurs in a text, it is necessary to first understand its meaning in order to understand the meaning of the symbol ω^{-1}. In the case when it is known to be a function, then ω^{-1} is obviously the inverse function corresponding to ω. This is completely different than the situation when ω is a scalar value and ω^{-1} should be understood as $1/\omega$. The goal of this module is to automatically retrieve the information that can be deduced from the context, but that is intentionally omitted by mathematicians to improve succinctness.

This work is designed to make use of Word Sense Disambiguation techniques in order to deal with formula context within the ARXMLIV corpus. More precisely, comprehensive word- and subformula-contexts of a mathematical formula may lead, in significantly many cases, to its partial or total disambiguation. As one of the existing predefined grammatical relations [dMM08], an apposition is defined as a grammatical construction in which two typically adjacent nouns referring to the same person or thing stand in the same syntactical relation to the rest of a sentence. An appositional modifier of a noun phrase (NP) is another NP immediately to the right of the first NP, serving to define or modify it (e.g. *"Heron, the mathematician"*). It also includes parenthesized examples. Looking at the combination of mathematics and text, we discovered that in most of cases, a math formula is like an explanation of the facts that are described in natural language, and so resemble an apposition in terms of English grammatical dependencies. One way of linking the formula to its context as an apposition is to substitute it with a comprehensive mathematical term and then collect the resulting grammatical dependencies. For instance, the input context: *"the value of the characteristic function (Formula)..."* will generate the following representative relations: *amod(function, characteristic)* and *appos(function, Formula)*, which will easily lead us to the conclusion that the considered *Formula* is actually a *(characteristic) function*.

Since most of the mathematical documents contain a lot of recurring formulations, we have derived three universally applicable pattern:[1]

a. $Formula \xrightarrow{appos} NP \xrightarrow{dobj} VP$
b. $Formula \xrightarrow{appos} NP \xrightarrow{nsubj} VP$
c. $Formula \xrightarrow{appos} NP_1 \xrightarrow{nsubj} NP_2$

where *dobj* denotes the direct object of the verb and *nsubj* denotes the noun phrase which is the syntactic subject of a clause.

Let us consider the given sentence: *"The cumulative density (Formula) is the probability to ..."*, with the following dependency flow (as matching the third pattern):

$$Formula \xrightarrow{abbrev} (cumulative)density \xrightarrow{nsubj} probability$$

Distinguishing which of the two NPs is actually more related to the formula raises the need to build a lexicon for mathematics. We use OPENMATH [BCC+03] for math symbols, while the extraction of meaningful terms (keywords) from our corpus statistics (i.e. *function*) together with their (adjectival) modifiers (i.e. *composite functions, inverse function, linear function, monotonic functions, periodic function, scaling function*) is achieved by using the term frequency-inverse document frequency (tf-idf [TFI]) weighting scheme applied to ARXMLIV documents.

Because there are infinitely many combinations of mathematical constructions, one can find many situations in which a formula cannot be explained as a whole, but in which it is possible to match different parts of it. Let us consider the formula $S^{-B}f(C \ln S)$, which we assume to represent a scaling function, and also assume the subformula f is a function. In this case, one can reveal that S^{-B} is actually a scalar that is multiplied by $f(C \ln S)$, which furthermore is a function application.

Using the RDF document representation allows an intuitive tokenized sentence format, having the mathematical fragments decomposed to the symbol level. This allows the system to process the full underspecification of the formulas, which gives best results in detecting subformulae and in relating to context. The results of the disambiguation process are stored back as stand-off annotations and are fully accessible for subsequent use by other applications.

3.2 Mathematical Formula Disambiguation

As mentioned in section 3.1, mathematical formulas contain ambiguities, some of them requiring deep context understanding. However, most of the mathematical ambiguities spotted by computers do not represent an ambiguity for a human reader. An example is $f(x) = x - 5$, where f could be multiplied by x or function f evaluated at x. Technically both are possible, but a reader with any experience in reading mathematical texts would see the formula as unambiguous straight away, due to highly standardized symbol conventions and the simple context.

[1] Grammatical relations appearing in this example follow the standard described in [dMM08]

The aim of the MFD module is to disambiguate the parts of formulas which require little or no context information. Our starting point is the creation of disambiguation rules like: *if "(" is followed by "symbol" followed by ")", then "(symbol)" is an argument to a function*. Or *if "symbol1" is followed by "symbol2" in a subscript, then separate them by ","* as in $F_{ij} = F_{i,j}$. For now, these rules are created manually by observing certain common patterns in the documents and analyzing their effect after applying the rules. It is clear that the order of rule application changes the final result. So we must define a strict order of application. This also gives the possibility of having so called "correcting" rules which again change the role of a certain set of symbols.

This method is very similar to the rule-based approaches employed to solve the Part of Speech Tagging (POST) problem [Bri95]. The rule generation in POST is however unsupervised. Since there is a strong connection between the Mathematical Formula Disambiguation problem and POST, we hope that eventually we will be able to make rule generation in our case unsupervised as well. But since we do not yet know the properties of mathematical language, we have begun with supervised rule generation .

3.3 Applicable Theorem Search

The goal of our architecture is to provide a convenient framework and abstraction layer for linguistic semantic analysis and for systems which use the resulting semantically-enriched documents. The first system operating on the documents resulting from the architecture is the proposed Applicable Theorem Search (ATS), described below. As was already mentioned, the corpus is composed of documents containing both mathematics and text in mixed discourse. Introducing mathematics-oriented semantics to the corpus through the Semantic Blackboard described in Section 2.3 and using a hybrid of existing NLP tools and adapted algorithms, one of the first applications targeting the corpus is semantic information extraction. One specific starting point in this area is idiom extraction, the search for *fixed-structure sentences* containing both text and mathematics (which we define as *idioms*). The Applicable Theorem Search engine is based on identifying "theorem-like" *idioms* and indexing their conclusions.

The ATS system uses the existing MATHWEBSEARCH [Mat09, KŞ06, KAJ+08] engine to index the mathematics in the hypothesis part of the idiom. Since the MATHWEBSEARCH system can only index mathematical formulae in Content MATHML format, it is important that the documents provided by the architecture are in the right format and contain the correct information. This is the part where the architecture plays a very important role in enriching the documents with the correct semantics (of mathematical formulae) and providing the right representation format. The ATS system uses crawlers similar to the ones used by MATHWEBSEARCH, which search for XHTML pages containing ContentMATHML and add the relevant found idioms to the index.

Simple examples of analyzed idioms are sentences of the form: "*X is defined as Y*", or "*If X then Y*". They are formed from fixed words or *keywords*, like "*defined*" or "*if*" and *placeholders* like *X* or *Y* arranged in a given pattern. A language idiom actually expresses a semantic relation between the placeholders, for example "*We define X as Y*" translates to *X* relates to *Y* by the equality relation. In order to differentiate the type of placeholders, the terminology of *hypothesis* and *conclusion* is used. The hypothesis is considered to be the

term that receives the property imposed by the relation that the idiom defines. For example, in the case of a definition, the *hypothesis* is the definiendum, while the *conclusion* is the definiens. The running "area of a triangle" example, found in Appendix 5.1.2 [app], fits the "*If X then Y*" idiom. In this case, the *hypothesis* is "T *is a scalene triangle with sides* a, b, c", while the conclusion is "$Area(T) = \frac{1}{2}ab \cdot sin(C)$".

In order to find and index the *conclusions* found in such sentences, the idioms need to be "spotted" in or retrieved from the scientific texts. The ATS system makes use of a comparison between three different approaches to mathematical idiom spotting, all based on NLP tools: a heuristic pattern-matching approach based on predefined cleartext patterns, a syntactical analysis approach based on syntax parsing and syntax tree fragment matching and a Discourse Representation Theory [Kam95] analysis based on matching patterns in resulting DRS structures [Cur07]. The three methods are run on the common ARXMLIV corpus described above and compared against each other for the purpose of finding the best idiom recall rate. Currently, the system is running based on the first idiom spotting approach and it will be updated if the latter 2 analyses prove to provide better recall rates of correct idioms. The **heuristic pattern matching** approach looks at ordered keywords. If the set of keywords and their order in a sentence matches a particular pattern, the sentence is then analyzed and the relevant conclusions and hypotheses are extracted, as raw text or math formulas found in between the keywords (or replacing a placeholder in the idiom pattern). Once the idioms are found, the mathematical formula part of their conclusion is added to the index. The hypotheses corresponding to each conclusion are also stored in the database, allowing for retrieval at query time.

4 Conclusion and Outlook

We have presented a large-scale analysis framework working on top of the ARXMLIV corpus. Our well-motivated modular design promises scalability and easy maintainability in the long-term, while harnessing the power of existing semantic tools and platforms. Based on the LATEXML system, the process of migration in between knowledge representations, while simultaneously preserving inferred semantics, becomes stable, fully-automated and encapsulated from the rest of the system. A data abstraction of the corpus documents, which stores them in the context of an online database of stand-off annotations in the W3C RDF format, provides an intuitive and distributed platform for potential developers, at a very small learning curve, as well as a rapid implementation and deployment time frame. Additionally, we provide a set of preprocessing and post-processing tools that increase the quality of explicit document semantics and support multi-purpose output formats for successive applications. In the Interaction and Visualization module we facilitate multiple purpose user interaction for various supervised techniques and, as a means to ease the development process, display inferred annotations and enable their creation and editing.

While the design of the architecture has stabilized, the development of the different modules is still ongoing and the maturity of the components varies. In particular, the Semantic Blackboard, the OMDOC generation module and the visualization framework are still under development and are yet to be properly tested. Tasks which are yet to be completed are the development of stand-off annotation conventions, an aggregation procedure for importing the revealed semantics to OMDOC and a standard for feedback annotations.

Hence, the described applications building on our architecture are yet to be completely integrated and made coherent with each other, justifying the lack of reported results in this paper. The preprocessing and post-processing modules are also currently being further developed and improved and there are plans for novel applications utilizing the power of our framework. We envision the design of an ontology for mathematical discourse relations, formalizing the RDF representation employed by the document abstraction layer. Nevertheless, we plan to deploy a publicly accessible server by the end of the current year, which will demonstrate the complete pipeline and functionality of the system, as described in this paper.

Furthermore, we are looking for collaborators in creating analysis modules that infer semantics from the ARXMLIV corpus, using the techniques from Computational Linguistics and Computational Semantics. The promise of the architecture and the aim of the future work of the LAMAPUN project is to achieve a large-scale formalization pipeline which performs full semantic enrichment of informal mathematical discourse, creating a consistent, unambiguous, formal representation.

References

[ABC+03] Ron Ausbrooks, Stephen Buswell, David Carlisle, Stéphane Dalmas, Stan Devitt, Angel Diaz, Max Froumentin, Roger Hunter, Patrick Ion, Michael Kohlhase, Robert Miner, Nico Poppelier, Bruce Smith, Neil Soiffer, Robert Sutor, and Stephen Watt. Mathematical Markup Language (MathML) Version 2.0 (second edition). W3C recommendation, World Wide Web Consortium, 2003.

[app] An Architecture for Linguistic and Semantic Analysis on the arXMLiv Corpus, Appendix supplement, http://www.kwarc.info/projects/lamapun/pubs/AST09_LaMaPUn+appendix.pdf.

[arX09a] ARXIV.ORG. Cornell University Library at http://arxiv.org/, seen March 2009.

[arX09b] ARXMLIV. Project home page at http://arxmliv.kwarc.info/, seen March 2009.

[BCC+03] S. Buswell, O. Caprotti, D. P. Carlisle, M. C. Dewar, and M. Gaetano. The OpenMath Standard iii, 2003.

[BCC+04] Stephen Buswell, Olga Caprotti, David P. Carlisle, Michael C. Dewar, Marc Gaetano, and Michael Kohlhase. The Open Math Standard, Version 2.0. Technical report, The Open Math Society, 2004.

[BKH01] Jeen Broekstra, Arjohn Kampman, and Frank Van Harmelen. Sesame: An Architecture for Storing and Querying RDF Data and Schema Information. In *Semantics for the WWW*. MIT Press, 2001.

[Bri95] Eric Brill. Unsupervised Learning of Disambiguation Rules for Part of Speech Tagging, 1995.

[CMBT02] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Annual Meeting of the ACL*, 2002.

[Cur07] James R. Curran. Linguistically motivated large-scale NLP with C&C and Boxer. In *Proceedings of the Demonstrations Session of the 45th Annual Meeting of the Association for Computational Linguistics (ACL*, pages 29–32, 2007.

[CZ04] Claudio Sacerdoti Coen and Stefano Zacchiroli. Efficient Ambiguous Parsing of Mathematical Formulae, 2004.

[dMM08] Marie-Catherine de Marneffe and Christopher D. Manning. Stanford typed dependencies manual, 2008.

[Fel98] Christiane Fellbaum, editor. *WordNet: an electronic lexical database*. MIT Press, 1998.

[FL04] David Ferrucci and Adam Lally. UIMA: an architectural approach to unstructured information processing in the corporate research environment. *Nat. Lang. Eng.*, 10(3-4):327–348, 2004.

[Gre09] GreaseMonkey - a Firefox extension for customizing web pages. http://diveintogreasemonkey.org, seen March 2009.

[KAJ+08] Michael Kohlhase, Ştefan Anca, Constantin Jucovschi, Alberto González Palomo, and Ioan A. Şucan. MathWebSearch 0.4, A Semantic Search Engine for Mathematics. manuscript, 2008.

[Kam95] Hans Kamp. Discourse Representation Theory. In J. Verschueren, J.-O. stman, and J. Blommaert, editors, *Handbook of Pragmatics*, pages 253–257. Benjamins, 1995.

[Koh06] Michael Kohlhase. OMDoc – *An open markup format for mathematical documents [Version 1.2]*. Number 4180 in LNAI. Springer Verlag, 2006.

[KŞ06] Michael Kohlhase and Ioan Şucan. A Search Engine for Mathematical Formulae. In Tetsuo Ida, Jacques Calmet, and Dongming Wang, editors, *Proceedings of Artificial Intelligence and Symbolic Computation, AISC'2006*, number 4120 in LNAI, pages 241–253. Springer Verlag, 2006.

[LSWC98] Ora Lassila, Ralph R. Swick, World Wide, and Web Consortium. Resource Description Framework (RDF) Model and Syntax Specification, 1998.

[Mat09] Math Web Search. http://kwarc.info/projects/mws/, seen Feb. 2009.

[Mil07] Bruce Miller. LaTeXML: A LaTeX to XML Converter. Web Manual at http://dlmf.nist.gov/LaTeXML/, seen September2007.

[PS08] Eric Prud'hommeaux and Andy Seaborne. SPARQL Query Language for RDF. Technical report, W3C, January 2008.

[RJF04] Eylon Caspi Richard J. Fateman. Parsing TEX into Mathematics, 2004.

[Sch05] Ulrich Schäfer. *Heart of Gold – an XML-based middleware for the integration of deep and shallow natural language processing components, User and Developer Documentation*. DFKI Language Technology Lab, Saarbrücken, Germany, 2005.

[SK08] Heinrich Stamerjohanns and Michael Kohlhase. Transforming the arχiv to XML. In Serge Autexier, John Campbell, Julio Rubio, Volker Sorge, Masakazu Suzuki, and Freek Wiedijk, editors, *Intelligent Computer Mathematics, 9th International Conference, AISC 2008 15th Symposium, Calculemus 2008 7th International Conference, MKM 2008 Birmingham, UK, July 28 - August 1, 2008, Proceedings*, number 5144 in LNAI, pages 574–582. Springer Verlag, 2008.

[TFI] TF-IDF measurement, http://nlp.cs.swarthmore.edu/~richardw/papers/sparckjones2004-idf.pdf.