Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 2204–2212
Marseille, 11–16 May 2020

MathAlign: Linking Formula Identifiers to their Contextual Natural Language Descriptions

Maria Alexeeva*, Rebecca Sharp†, Marco A. Valenzuela-Escárcega†, Jennifer Kadowaki+, Adarsh Pyarelal‡, and Clayton Morrison‡
Dept. of Linguistics*, Dept. of Computer Science†, Dept. of Astronomy+, School of Information‡
University of Arizona, Tucson, AZ
{alexeeva, bsharp, marcov, jkadowaki, adarsh, claytonm}@email.arizona.edu

Abstract

Extending machine reading approaches to extract mathematical concepts and their descriptions is useful for a variety of tasks, ranging from mathematical information retrieval to increasing accessibility of scientific documents for the visually impaired. This entails segmenting mathematical formulae into identifiers and linking them to their natural language descriptions. We propose a rule-based approach for this task, which extracts LaTeX representations of formula identifiers and links them to their in-text descriptions, given only the original PDF and the location of the formula of interest. We also present a novel evaluation dataset for this task, as well as the tool used to create it. The data and the source code are open source and are available at `https://osf.io/bdxmr/` and `https://github.com/ml4ai/automates`, respectively.

Keywords: machine reading, relation extraction, math information retrieval, corpus creation, tool creation

1. Introduction

Automatic reading of scientific literature has received attention due to the proliferation of publications, as well as the importance of their content. A crucial aspect of understanding scientific publications is understanding the mathematical *formulae* expressed in those publications, since they provide an explicit, concise representation of the key relations between the relevant concepts (Schubotz et al., 2018). Extending machine reading to extract relations from mathematical formulae requires reading the formulae into internal representations, segmenting them into their component terms, and linking the terms to their textual descriptions.

The proposed work stands to impact many different tasks that rely on understanding scientific content, including math information retrieval (Schubotz et al., 2016; Kristianto et al., 2017), scientific keyphrase and relation extraction (Sterckx et al., 2016; Augenstein and Søgaard, 2017; Marsi and Öztürk, 2015), mathematical formula format conversion (Schubotz et al., 2018), and even accessibility efforts to make written content available to the visually impaired (Pontelli et al., 2009; Alajarmeh and Pontelli, 2015). Beyond these tasks, there have also been recent efforts to apply machine reading to automatically analyze, ground, and compare executable models (Pyarelal et al., 2019). For this task, formulae found in papers describing these models can illuminate important semantics of how model variables interact with each other.

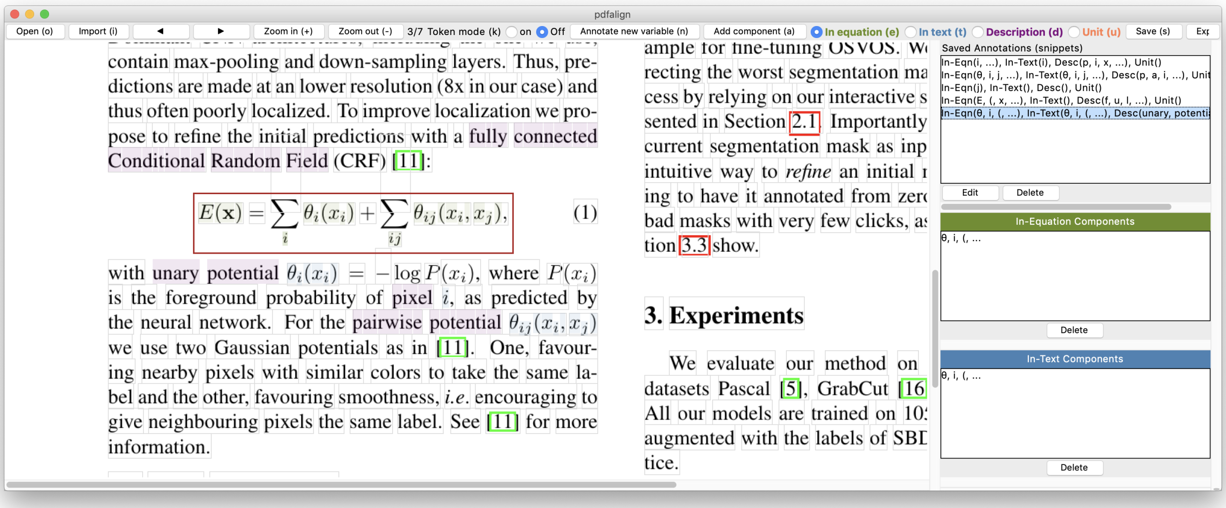
Specifically, given a formula such as the one shown in Figure 1 (Details of the annotation tool interface depicted in Figure 1 will be described below in Section 5.3.), there are several mathematical identifiers including simple identifiers (i.e., x_i), compound identifiers ($\theta_{ij}(x_i, x_j)$) and even subscripts (i). Each of these identifiers is referred to in the text, where the author expresses what they represent. Our goal is to find these identifiers in the equation and link them to their in-text descriptions.

Further, for several reasons, we argue that approaches to this task should ideally rely only on the text of the paper describing the formula. First, it is less expensive, as human annotation for this task is time-consuming, requiring training and great attention to detail. Second, not all domains have readily available comprehensive ontologies of mathematical identifiers and variables to refer to, and usage of specific identifiers varies widely across disciplinary subdomains and authors. Even when these resources do exist, segmenting the formulae and considering the in-context descriptions is a necessary step prior to any grounding to existing ontologies. Finally - and perhaps most importantly - authors of scientific papers usually describe the formula identifiers at the specific level of detail most relevant to the discussion at hand.

Here we propose an approach for the automated segmentation of a given formula and the linking of the segmented identifiers to in-text natural language descriptions, given only the original PDF and the location of the formula of interest. In particular, our main contributions are:

1. An approach to scientific information extraction that extracts mathematical identifiers from formulae and links them to their descriptions in surrounding text. Importantly, our approach operates over papers in PDF format, assuming only that the location of the formula of interest is provided. Our rule-based approach for extracting identifier descriptions is readily interpretable, maintainable, and domain-agnostic.

2. A development and evaluation dataset for empirically evaluating approaches to mathematical information extraction. For each identifier in a given formula, we annotate their in-text mentions, descriptions, and units in the surrounding context from the scientific paper. We provide a detailed description of the annotation protocol and an analysis of the annotation quality and consistency for this complex annotation task. Additionally, for each annotated identifier, we retrieve the



Token mode (k) on [✓ Off]

[✓ In equation (e)] In text (t) Description (d) Unit (u)

contain max-pooling and down-sampling layers. Thus, predictions are made at an lower resolution (8x in our case) and thus often poorly localized. To improve localization we propose to refine the initial predictions with a fully connected Conditional Random Field (CRF) [11]:

$$E(\mathbf{x}) = \sum_i \theta_i(x_i) + \sum_{ij} \theta_{ij}(x_i, x_j), \quad (1)$$

with unary potential $\theta_i(x_i) = -\log P(x_i)$, where $P(x_i)$ is the foreground probability of pixel i, as predicted by the neural network. For the pairwise potential $\theta_{ij}(x_i, x_j)$ we use two Gaussian potentials as in [11]. One, favouring nearby pixels with similar colors to take the same label and the other, favouring smoothness, *i.e.* encouraging to give neighbouring pixels the same label. See [11] for more information.

ample for fine-tuning OSVOS. W
recting the worst segmentation ma
cess by relying on our interactive s
sented in Section 2.1. Importantly
current segmentation mask as inp
intuitive way to *refine* an initial r
ing to have it annotated from zer
bad masks with very few clicks, as
tion 3.3 show.

3. Experiments

We evaluate our method on
datasets Pascal [5], GrabCut [16
All our models are trained on 10
augmented with the labels of SBD
tice.

Saved Annotations (snippets)

```
In-Eqn(i, ...), In-Text(i), Desc(p, i, x, ...), Unit()
In-Eqn(θ, i, j, ...), In-Text(θ, i, j, ...), Desc(p, a, i, ...), Unit
In-Eqn(j), In-Text(), Desc(), Unit()
In-Eqn(E, (, x, ...), In-Text(), Desc(f, u, l, ...), Unit()
In-Eqn(θ, i, (, ...), In-Text(θ, i, (, ...), Desc(unary, potenti
```

In-Equation Components

θ, i, (, ...

Delete

In-Text Components

θ, i, (, ...

Delete

Figure 1: Snapshot of the our annotation tool, PDFAlign, currently part way through annotation the first mathematical formula of arXiv paper 1801.00269. Details of the annotation schema are provided in Section 5., and details of the annotation tool itself are in Section 5.3..

sequence of the original LaTeX tokens that produced it, facilitating future approaches that use LaTeX as an intermediate representation.

3. An open-source annotation tool for mathematical information extraction from PDFs. Our proposed tool can annotate at both token and glyph level, allowing for both speed and precision. While this tool can be immediately used to annotate additional formulae, it is also designed to be extensible, so that it may be adapted to annotate PDFs for other mathematical or textual relations within a particular domain of interest. The tool, along with all of the code and data necessary to replicate the proposed work, is freely available.[1]

2. Related Work

Approaches to information extraction (IE) tend to fall into one of two broad categories - approaches based on machine learning (ML) and approaches based on rules. While ML-based approaches are currently favored in academia (e.g. (Zeng et al., 2015; Nguyen and Grishman, 2015; Jat et al., 2018)), rule-based approaches (Appelt et al., 1993; Cunningham et al., 2002; Chang and Manning, 2014; Kluegl et al., 2016) are well received in industry because of their interpretability and consequent maintainability (Chiticariu et al., 2013). Furthermore, rule-based approaches do not have the high data requirements of ML, making them a good fit for our purposes. For these reasons, we propose a rule-based approach to mathematical relation extraction (Section 4.).

Specifically relevant to our approach, several systems have been proposed for extracting natural language descriptions of mathematical expressions from free text. Quoc et al. (2010) use pattern matching for this purpose, and their goal of mining coreference relations between mathematical formulae and their descriptions is similar to what we refer to as linking here. However, they only consider concepts that describe the entire formula itself, whereas we are largely concerned with segmenting the formula to enable the extraction of the descriptions of individual identifiers. Similar to what we propose here, Yokoi et al. (2011) use a combination of pattern matching and machine learning to extract mathematical mentions (these correspond to mathematical identifiers in our paper) and their descriptions. However, unlike what we propose here, they do not link them to formulae that have these mentions as components. This, along with the fact that they work on scientific publications in Japanese, makes their dataset impossible for us to use in the evaluation of our system. Kristianto et al. (2014) also use an ML-based approach to extract mathematical identifiers and their descriptions, including descriptions that go beyond a single noun phrase, and account for multiple and discontinuous descriptions. However, the extracted mathematical expressions were not segmented, and the expression-description pairs were not linked to any formulae, while with our system and our dataset, we attempt to do that.

Closely related to our approach, Schubotz et al. (2016) propose a method for extracting identifiers from formulae and linking them to descriptions. Further, they provide a gold set of annotations of individual mathematical identifiers from formulae and their corresponding descriptions. However, we are unable to use this dataset for our evaluation because they include gold descriptions from a variety of sources, including the context of the paper as well as external expert annotations and web resources. While this is useful information, it is not well-aligned with our goal (i.e., a system which can associate mathematical identifiers with their author-provided descriptions). Further, they limit their identifiers to a single variable which may have one or more subscripts, and their descriptions to noun-(preposition)-noun or adjective-noun noun phrases. In our work, we relax these constraints, allowing composite ex-

[1] `https://github.com/ml4ai/automates`

pressions to serve as identifiers, deferring to the authors' provided in-text descriptions, and allowing for more complex descriptions (see Section 5.). All of these systems rely on the formulae having been previously identified, as do we.

3. Approach

In order to parse formulae found in scientific papers into their component identifiers and subsequently link these identifiers to their textual descriptions in the papers, we propose the three-stage approach shown in Figure 2. First, given a PDF of a scientific paper and the location within the paper of an formula of interest, we parse the formula image into a LaTeX intermediate representation (Section 4.1.). LaTeX is an ideal intermediate representation for formulae as it is machine-readable and also preserves the formatting (e.g., bold, italics, super and subscript) which is semantically meaningful for mathematical expressions. Another advantage of using LaTeX is that it allows us to leverage arXiv[2], a preprint repository that contains a vast number of freely available scientific papers from multiple fields along with with their LaTeX source code, making machine learning approaches feasible.

Next, we use our rule-based information extraction system, Odin (Valenzuela-Escárcega et al., 2016), to find and extract in-text mentions of mathematical identifiers and their descriptions (Section 4.2.).

Once we have the LaTeX representation of the image and the extracted text identifier mentions, we segment the formula tokens into the relevant chunks (i.e., individual identifiers and semantically meaningful compositions) and align them to the text extractions. In order to evaluate the quality of our extractions and alignments, we propose a new dataset, MathAlign-Eval, which consists of mathematical formula identifiers, their in-text mentions, and any natural language descriptions and/or units present in a three paragraph context. This dataset is described in detail in Section 5..

4. Models

4.1. Formula Extraction

To convert our formula images to LaTeX, we use the im2markup model of Deng et al. (2017), who frame the task as a sequence-to-sequence task where the source sequence comes from the image of the equation and the destination sequence is the LaTeX tokens that produce it. A high-quality implementation of this model is provided in the OpenNMT toolkit (Klein et al., 2017). Specifically, the model consists of a convolutional neural network followed by a row encoder to encode the image into a representation that can be used as input for an attention-based neural decoder. The pre-trained im2markup model is freely available,[3] but was trained on data from only the high-energy physics domain (Gehrke et al., 2003) and only on images generated with a specific formatting. We found that this model was unable to generalize to the images we generated from arXiv across many domains. Therefore, we retrained the OpenNMT implementation of the model on an order of magnitude more images from diverse domains and also applied data augmentation to help the model be more robust to different image resolutions. Specifically, for each of our training images we generated a second version with either blurring, downsampling, and some morphological operations, with parameters randomly sampled from a specified distribution. This resulted in a total of 1,716,833 images for training and 214,690 images for validation.[4]

4.2. Extracting Identifier Descriptions

As discussed in Section 2., since we do not have sufficient labeled data to both train and evaluate a system for extracting formulae identifiers and their descriptions from unstructured text, we focus on approaches that do not require direct supervision. Specifically, we propose a rule-based information extraction (IE) system that uses an Odin grammar (Valenzuela-Escárcega et al., 2016). Odin is an information extraction framework that includes a declarative language supporting both surface and syntactic patterns along with a runtime system. Our system is based on Eidos (Sharp et al., 2019), an open-domain causal IE system. Unlike in a typical bottom-up IE system, where the first step involves finding entities, followed by identifying events that operate over the entities, Eidos was designed so as to not be limited to a fixed set of concepts (i.e., entities and events) to make it domain agnostic. Here we use a *hybrid* approach. For our extraction of mathematical identifiers and their descriptions, we do not limit ourselves to a fixed set of identifiers so that our approach is similarly domain agnostic. However, we do attempt to identify them whenever possible using a set of nine rules.Then we have two rules that are designed to extract descriptions of these previously found identifiers. However, we also use a set of six rules that do not rely on previously found identifiers. For these top-down rules, we first find trigger words signaling a *Description* relation, and then look for the concepts that participate in the relation (i.e., the identifier and the description itself). Similar to Eidos, we also expand our initial descriptions using certain outgoing dependencies to capture content that extends beyond a single noun phrase (see Figure 3). Notably, our approach was developed on crop modeling papers for the DARPA ASKE program[5], but in an effort to not overfit to the task, the rules were not adjusted or tuned for our new dataset.

5. Dataset

We present a development and evaluation dataset[6] for evaluating automatic reading of formulae in scientific publications, as well as the open-source annotation tool that we developed for its creation.

For each formula, we annotate all the discernible mathematical identifiers inside the formula along with the associated in-text instances, descriptions, and units of measure if present. Specifically, we annotate identifier frames, where

[2]https://arxiv.org

[3]https://github.com/harvardnlp/im2markup

[4]To prevent overfitting, we exclude all images from papers that were present in the set of annotation candidates for our MathAlign-Eval dataset, Section 5..

[5]https://www.darpa.mil/program/automating-scientific-knowledge-extraction

[6]https://osf.io/bdxmr/

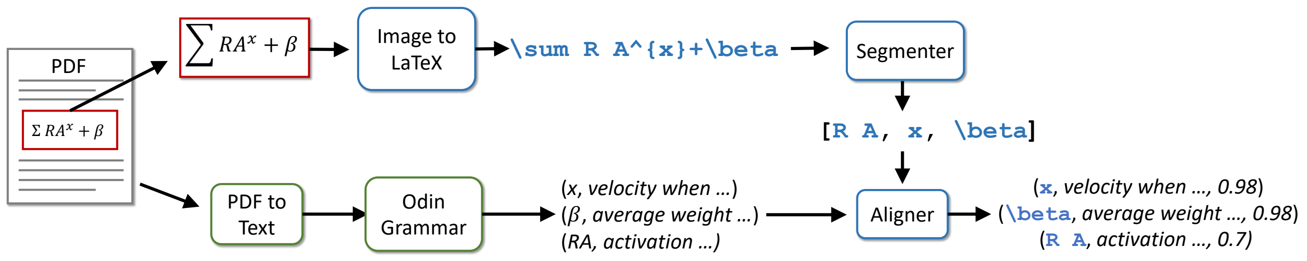


Figure 2: Architecture of our approach. Given a scientific paper in PDF format, we process it in two ways. First, we extract the images of the formulae and convert them to LaTeX, a machine-readable intermediate representation. Then, we extract mentions of mathematical identifiers and their descriptions from the text in the PDF using a rule-based system built upon the Odin information extraction framework (Valenzuela-Escárcega et al., 2016). We then segment the LaTeX and align the component identifiers in the segments to their extracted textual descriptions.

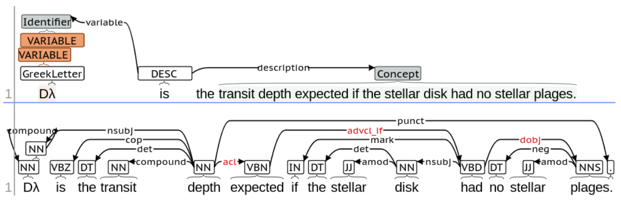

Identifier
variable
VARIABLE
VARIABLE
GreekLetter
DESC
description
Concept
Dλ is the transit depth expected if the stellar disk had no stellar plages.
compound
nsubj
cop
det
punct
advcl_if
mark
dobj
neg
acl
amod
NN VBZ DT NN NN VBN IN DT JJ NN VBD DT JJ NNS .
Dλ is the transit depth expected if the stellar disk had no stellar plages.

Figure 3: An example of how expanding an initial description along syntactic dependencies (here, `acl`, `advcl_if` and `dobj`, shown in red) can help capture crucial additional information about the identifier.

each frame contains four properties as they are expressed in a formula and text: (1) in-formula mentions of the identifier (for example, $\theta_i(x_i)$ in the formula shown in Figure 1), (2) in-text mentions of the identifier, (3) natural language descriptions (e.g., *unary potential*), and (4) units of measurement. Each of these properties, in turn, has associated with it a set of zero or more mentions, where a given mention does not need to be contiguous. For example, in the description text, *where L and V are characteristic length and velocity scales for the system*[7], the description of L is the non-contiguous mention, *characteristic length ... scales for the system*. A single annotation frame constitutes the relation between these four elements for a single identifier. There is always a value for (1), but the others may be absent, depending on the formula and the context.

In an attempt to overcome the limitations discussed in Schubotz et al. (2016), in addition to annotating identifiers consisting of one variable with a subscript, we also annotate multi-variable identifiers, identifiers containing superscripts, and identifiers that are only present in the sub- or superscripts. While we annotate every single-variable identifier, for the more complicated cases (e.g., compound identifiers such as $\theta_{ij}(x_i, x_j)$), we only annotate identifiers that are explicitly mentioned or described in text. Similar to Kristianto et al. (2014), we distinguish between mathematical descriptions and other mathematical text, e.g., properties, and only include as descriptions the spans of text that provide definitions of identifiers (referred to as *definiens* in Schubotz et al. (2016)); and allow for multiple and discontinuous descriptions.

[7]Example taken from arXiv paper 1808.08122.

In addition to providing the string values of every component annotated, we provide the information on the location of each component in the original PDF, including the page number and the bounding boxes of the glyphs. Further, we provide the approximate location of each annotation element within the original LaTeX source, along with the original LaTeX expression for the annotated identifiers.[8]

With all the identifiers in each formula annotated, regardless of whether or not their descriptions are present in the surrounding text, we hope that this dataset can be used to evaluate text reading, formula segmentation, and text-formula linking systems.

5.1. Data Source

We curated our dataset from open-sourced papers on arXiv, due to its richness in text-formula alignment data. After retrieving all papers submitted in 2018 and their respective arXiv metadata, we parsed the LaTeX files to search for formula environments (e.g., equation, align, gather) to aggregate a list of formulae from each paper.

To reduce the annotation burden and to ensure that as many of the formula descriptions as possible can be found within their respective contexts, we carefully selected formulae by filtering papers on formula and context statistics, where we define the context as the three paragraphs before and after the formula. Specifically, we excluded all formulae from papers that contained more than 5 formulae, and also those whose contexts contained an insufficient amount of natural language content (or too many in-text mathematical expressions). That is, we eliminated expressions whose context had more than 50 characters contained within math environments. These would be both overly difficult to annotate fully and less useful for our goal of evaluating links between identifiers and their textual descriptions. Furthermore, we considered only formulae with 200 or fewer LaTeX tokens, to prevent annotating overly complicated formulae

[8]A notable exception is that if there were author-defined macros in the original LaTeX expression, we provide the expanded version in the dataset, i.e., the version that compiles in isolation.

	Count		
Total number of papers	116		
Total number of formulae	145		
Total number of identifiers	757		
Number of identifiers with descriptions	584		
	Min	Max	Avg
Number of identifiers per formula	2	15	5.2
Number of formula mentions per identifier	1	7	1.5
Number of tokens per description	1	21	4.0

Table 1: Statistics for our MathAlign-Eval dataset. We show the overall number of formulae and identifiers in the dataset as well as the total number of papers represented. Also shown are the distribution statistics for the formulae, identifiers, and descriptions in the dataset.

arXiv Domain	Paper Count	Formula Count
cs	42	50
econ	2	4
physics:astro-ph	24	33
physics:cond-mat	20	23
physics:hep-ex	2	2
physics:hep-ph	3	4
physics:physics	10	13
physics:quant-ph	4	5
physics:nlin	1	1
physics:nucl-ex	1	1
physics:nucl-th	2	2
q-bio	2	3
stat	3	4

Table 2: Domain statistics for our MathAlign-Eval dataset. We show the breakdown of the number of papers and formulae per arXiv domain.

and identifiers. [9]
Once we had selected a set of mathematical formulae to annotate, we gathered the formula content and rendered it in isolation to generate an image of the formula. Then, we used OpenCV's template matching[10] to locate the page and bounding box for the formula in the original PDF. This process gives us (a) the gold sequence of latex tokens, (b) the position of the formulae in the paper, and (c) the cropped image of each formula.

5.2. Dataset Statistics and Analysis

Following a three month process of developing the annotation guidelines, the annotations were completed over the course of a month by a group of researchers, including graduate students and senior researchers. The process of creating the guidelines included three rounds of small scale annotation (two annotators each round) and discussions of the results based on qualitative assessment of annotations produced in each round. This process allowed us to decide on the information to be annotated, improve the comprehensibility and establish the scope of the annotation guidelines, and make key decisions about the features to include in the annotation tool. During the annotation exercise, the annotators completed both the segmentation task and the linking task jointly, as opposed to first doing the segmentation and later the linking. The decision to follow this procedure was based on the fact that for our task, segmentation is largely dependent on linking—multi-variable identifiers and identifiers in the subscripts and superscipts are only annotated if they have associated in-text descriptions.

The statistics for the dataset, including the total number of formulae and identifiers included, are given in Table 1. Fifty-seven of the annotated formulae were double annotated in order to calculate inter-annotator agreement; however six formulae were excluded from the dataset because they were not suitable for the task. The remaining 51 double annotated formulae comprise the development set for the pipeline proposed in the paper. The test partition has an additional 94 annotated formulae.

The domain representation statistics for the entire dataset are provided in Table 2. Our dataset is comprised of 145 equations from 116 papers: 92 papers contributing 1 equation each, 19 papers contributing 2, and 5 papers contributing 3. While the dataset is strongly represented by the disciplines of computer science, astrophysics, and condensed matter physics, the dataset spans 13 unique arXiv domains. Of these 13 domains, 9 are physics subfields and the remaining 4 domains include computer science, economics, quantitative biology, and statistics.

To evaluate the quality of the dataset, we calculated the inter-annotator agreement score for a subset of the annotations. To the best of our knowledge, there is no established procedure for evaluating the type of data produced for this task. However, since for every distinct formula identifier we produce a set of associated components, which could include additional instances of the identifier within the formula as well as the associated in-text instances, descriptions, and units, our task appears to be most similar to coreference annotation. Passonneau (2004) provides a flexible framework for adapting Krippendorff's alpha reliability score for coreference annotation in a way that allows for an evaluation that takes into account the severity of errors. We follow the procedure similar to that in Passonneau (2004), with the formula identifier being the unit coded and the rest of the components together with the coded unit making up an equivalence class. Since under our approach it is possible for formula identifiers to overlap, we do not allow for lenient evaluation of formula identifiers, that is, the coded units have to match exactly, while for the other components, evaluation is lenient. The code for calculating inter-annotator reliability score was adapted from an exist-

[9]In rare cases, we note that some papers in our dataset have superficially passed the filtering criteria due to the use of rare keywords to generate LaTeX formula environments. In such cases, we asked annotators to disregard overly complicated or long formulae.

[10]https://docs.opencv.org/2.4.13.7/doc/tutorials/imgproc/histograms/template_matching/template_matching.html

ing Python implementation of Krippendorff's alpha [11].
While the scores for half of the annotated formulae were over the threshold established for this inter-annotator reliability measure (0.67, Passonneau (2004)), a number of annotated formulae had a much lower score bringing the overall score for 57 equations down to 0.59. An error analysis uncovered the following recurring issues: (1) misuse of the annotation tool, (2) annotating subscripts as separate identifiers without them being defined in text, and (3) inconsistent annotation of cases that did not occur in the data used to develop the guidelines (e.g., the use of natural language as identifiers within the formula). Based on the error analysis, we manually checked the single-annotated formulae for these recurrent errors to ensure the quality of the gold dataset; also, we have updated the annotation guidelines in light of these recurring errors in hopes of better informing future annotators. Another major source of disagreement was the difficulty of establishing equivalence between formula identifiers and in-text identifiers. While the original guidelines instructed annotators to link formula identifiers to descriptions only if the intermediate in-text identifier was exactly identical (aside from the presence or absence of an index in a time-series), based on seeing more data, we found that we cannot always abide by this heuristic. In these cases, annotators had to make a judgment call based on the context provided, and these calls were not always equivalent between annotators.

5.3. PDF Annotation Tool

While we first tried to use brat (Stenetorp et al., 2012) to annotate our data, we found that there were a large number of links that spanned several lines, even paragraphs, and the LaTeX source formulas were hard for annotators to segment. As a result, the annotation process was cumbersome, and it was difficult to ensure coverage of all identifiers. For this reason, to better facilitate the human annotation needed for our purposes, we developed an open-source tool that makes use of the original LaTeX source for the paper as well as the glyph-level bounding boxes. For the user's convenience, the annotation is done directly on the PDF file, and the annotations are propagated to the LaTeX source. To prepare the PDF to be annotated, the tool first uses poppler[12] to render the pages of the PDF into images. It then uses pdfminer[13] to find the bounding box of each glyph in the document, which we combine with a set of heuristics, e.g., split at spaces, to generate token bounding boxes. We then use SyncTeX (Laurens, 2008) to align each bounding box to the corresponding line in the original LaTeX source.
A challenge with using SyncTeX is that it only gives the line number for the LaTeX, and may return the line of the formula itself or the line where the formula's environment is ended, but here we are interested in knowing the exact portion of the LaTeX which corresponds to a particular formula identifier. Therefore, in order to find the sequence of LaTeX tokens corresponding to each identifier, we implemented a LaTeX tokenizer based on the description of Eijkhout (1991), and used this to tokenize the formula. We then generated all valid subsequences of these tokens and used colorization to select the subsequence that best matched the manually annotated bounding boxes corresponding to the identifier. This is done by calculating the precision and recall of the colorized pixels with respect to the gold bounding box (i.e., how many of the colorized pixels were inside the bounding box versus outside, and how many of the foreground pixels inside the bounding box were colorized). The selected subsequence of LaTeX tokens was the one which maximized the corresponding F1 score.
We obtain the text representation of each of our mentions directly from the PDF being annotated. Each glyph inside a PDF can have an extraction value associated with it, i.e., the unicode representation of the glyph. For example, with ligatures the character sequence "fi" may be rendered as a single glyph but the extraction value (used for search and copying content) is `fi`. These extraction values are typically included in the PDF as a character mapping table, or CMap.[14] However, PDFs generated with LaTeX do not include these values for all fonts (particularly relevant for us, for certain math fonts). Therefore, in order to have access to the text content of the annotated glyph-level bounding boxes, we gathered the mappings from glyph name to unicode sequence for several of the font families most commonly used for math expressions. We then used this mapping to extend the internal lookup table used by pdfminer[15] so that it has access to the correct unicode value.
The tool is open source and available as a stand-alone application.[16]

6. Experiments and Results

In order to evaluate our segmentation and linking approach, we compare our extracted formula identifiers, and any descriptions they have, with the gold set. In particular, we compare the unicode value of the annotated bounding boxes with the unicode value of our identifier LaTeX and the descriptions. To get a unicode value for the LaTeX predicted by our approach, we first tokenized it with our LaTeX tokenizer. Then, we use a set of heuristics to produce a canonical version of the identifier. Next, we replace some LaTeX control sequences with their corresponding unicode character, using a manually built lookup table that is meant to contain most math symbols and Greek letters. Finally, we remove elements that provide layout information, but not content (as these are not represented in the unicode).
As our task is substantially different from any we were able to find in previous work (see Section 2.), we are unable to compare against other datasets and approaches. However, as our approach is the first for this dataset, we anticipate future comparisons.
We consider three levels of evaluation: strict, lenient, and soft. For the *strict* evaluation, when comparing predicted

[11] `https://github.com/grrrr/krippendorff-alpha`
[12] `https://poppler.freedesktop.org`
[13] `https://github.com/pdfminer/pdfminer.six`
[14] `https://www.adobe.com/content/dam/acom/en/devnet/acrobat/pdfs/pdf_reference_1-7.pdf`
[15] Our fork with this modification is available at `https://github.com/ml4ai/pdfminer.six`.
[16] `https://github.com/ml4ai/automates/tree/master/pdfalign`

Eval Level	Precision	Recall	F1
Segmentation and Full Description			
Strict	0.17	0.23	0.19
Lenient	0.22	0.30	0.26
Soft	0.23	0.32	0.27
Segmentation and Description without Expansion			
Strict	0.16	0.22	0.18
Lenient	0.22	0.31	0.26
Soft	0.23	0.32	0.27
Segmentation Only			
Strict	0.56	0.68	0.62
Lenient	0.60	0.74	0.66

Table 3: Results of our approach on the tasks of segmenting a mathematical formula and linking the segmented identifiers to in-text descriptions. We provide the precision, recall, and F1 score using three different evaluation levels: *strict*, which requires exact string match of both identifier and description; *lenient*, which allows for certain specific forms of subsumption; and *soft*, which allows descriptions which overlap. We show results for our full system, the system without expanding the extracted text decriptions (see Section 4.2.), and also the segmentation only.

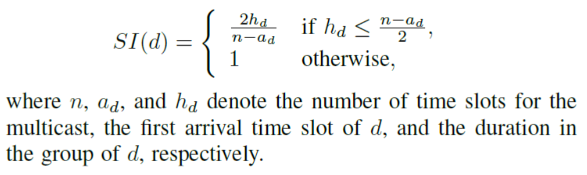
and gold identifiers and descriptions, we consider it a match if there is an exact string match for one of the identifier mentions and one of the description mentions, where a special token is used to denote the lack of any description for the identifier. With this level of evaluation, any extra word in a description results in a failure to match. We additionally include a *lenient* evaluation. This is similar to the strict evaluation, except that we (a) allow identifiers corresponding to functions (e.g., `f(x,y)`) to match with only the function "name" (here, `f`) and (b) allow descriptions to match if one is fully subsumed within the other. Finally, similar to Kristianto et al. (2014), we also include a *soft* matching, which extends the lenient description matching to also include overlapping descriptions. For this, we find the longest common substring and ensure that it contains words that are not stopwords (to avoid counting matches whose substring is simply *the*, etc.).

As we hope that the proposed dataset can additionally be used to evaluate the task of formula segmentation, we also provide the performance of our approach in only the segmentation task, i.e., ignoring the descriptions. We report the results of our approach, with both the full task as well as the segmentation only task, using all three evaluation levels in Table 3.

The task of segmenting the formula is simpler than the compound task of both segmenting and linking, and unsurprisingly the results reflect this difference. The performance of our approach on the segmentation task (0.62 F1 for strict and 0.66 for lenient) is much higher than for the full task (F1 scores from 0.19 to 0.27). Also unsurprisingly, the lenient and soft scores are much higher than the strict (0.26 and 0.27 versus 0.19). Notably, the soft measure is not much higher than the lenient (0.27 versus 0.26, respectively), though it is far less restrictive. This suggests that when the system is able to locate the general location of the description of an identifier, but doesn't extract the exact span annotated, the missing or extra information is on the edges of the extracted description (e.g., as with an additional clause).

Some previous work has limited the descriptions to single noun phrases (Schubotz et al., 2016), while others have allowed for larger text spans (Kristianto et al., 2014). We find that our choice to allow for expansion of descriptions along syntactic dependencies (such as certain prepositional phrases) does not adversely affect the performance. The fact that removing the expansion does not noticeably raise or lower performance suggests that there are descriptions of both types (i.e., of simple noun phrases as well as longer clauses), and ideally a future system would find these longer phrases, but only when necessary.

Though a direct comparison is not possible, we note that our performance is lower than that of Kristianto et al. (2014); however, their task is limited to linking descriptions to mathematical identifiers in the same sentence, while we must find descriptions of identifiers in a much larger, three paragraph context (see Section 5.). That said, the gains they see in their task when using a machine learning approach, as compared with pattern-based approaches, motivate gathering a much larger set of annotations to facilitate supervised training, which we defer to future work.



$$SI(d) = \begin{cases} \frac{2h_d}{n-a_d} & \text{if } h_d \leq \frac{n-a_d}{2}, \\ 1 & \text{otherwise,} \end{cases}$$

where n, a_d, and h_d denote the number of time slots for the multicast, the first arrival time slot of d, and the duration in the group of d, respectively.

Figure 4: An example of in-text identifiers (n, a_d, and h_d) and their descriptions appearing in a coordinate construction, i.e., there are other identifiers present between the target identifier (e.g., n) and its description (*the number of time slots for the multicast*). Example taken from arXiv paper 1801.00110.

6.1. Error Analysis and Future Work

We conducted a lightweight manual error analysis on the entire development dataset, which uncovered several future directions of improving the proposed pipeline. In several cases, the false negative was the result of the formula identifier and its in-text equivalent not being exactly identical, but differing in terms of presence or absence of subscripts. The outstanding issue that has to do with this class of errors is the difficulty for non-experts to establish equivalency between identifiers in scientific publications, which was also referenced in section 5.2.; the issue will require further analysis and a more sophisticated procedure for matching formula identifiers with their in-text counterparts. A major source of false negatives, which will also need to be addressed in the future, was the fact that many in-text identifiers and their descriptions appear in coordinate constructions (see Figure 4). This construction is quite common, and our current rule-based approach is unable to handle it. We also found that in several cases, what was counted as a false positive due to the extracted description not matching the gold data was in fact a true positive; however, the description for the identifier was extracted from a portion of the paper outside of the annotation window. This could be addressed by allowing annotators to annotate descriptions throughout the paper.

In addition to the future directions discussed above, we see the value in applying machine learning techniques to this task. In order to support supervised machine learning approaches, we envision gathering training data through a large-scale annotation exercise, enabled by a web-based version of the annotation tool, which is currently under development.

7. Conclusions

Here we have presented an approach to the joint tasks of segmenting mathematical formulae into their component identifiers and linking these identifiers to natural language descriptions. Specifically, given only a PDF document and the location of the formula of interest, we are able to convert the image of the formula into an intermediate LaTeX representation, which we segment and link to identifiers and descriptions extracted from the text of the PDF. Our rule-based approach is interpretable and extendable, allowing for modifications for future use on new segmentation and linking tasks. For evaluating our approach, as well as facilitating future work on this task, we provide an evaluation dataset, MathAlign-Eval, which consists of over 700 identifiers and their descriptions, as well as the original span of LaTeX used to generate the identifier. Further, we provide our annotation tool, PDFAlign, to enable extension of the dataset, and all of our code, data, and resources are open-source and readily available.

8. Acknowledgements

We thank the anonymous reviewers for their constructive feedback. This work is supported by the Defense Advanced Research Projects Agency (DARPA) as part of the Automated Scientific Knowledge Extraction (ASKE) program under agreement number HR00111990011. Marco Valenzuela-Escárcega declares a financial interest in LUM.AI. This interest has been properly disclosed to the University of Arizona Institutional Review Committee and is managed in accordance with its conflict of interest policies.

9. Bibliographical References

Alajarmeh, N. and Pontelli, E. (2015). Visual disabilities, information technology, and the learning of mathematics. In *Encyclopedia of Information Science and Technology, Third Edition*, pages 345–353. IGI Global.

Appelt, D. E., Hobbs, J. R., Bear, J., Israel, D., and Tyson, M. (1993). Fastus: A finite-state processor for information extraction from real-world text. In *IJCAI*, volume 93, pages 1172–1178.

Augenstein, I. and Søgaard, A. (2017). Multi-task learning of keyphrase boundary classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 341–346, Vancouver, Canada, July. Association for Computational Linguistics.

Chang, A. X. and Manning, C. D. (2014). TokensRegex: Defining cascaded regular expressions over tokens. Technical Report CSTR 2014-02, Computer Science, Stanford.

Chiticariu, L., Li, Y., and Reiss, F. R. (2013). Rule-based information extraction is dead! long live rule-based information extraction systems! In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 827–832.

Cunningham, H., Maynard, D., Bontcheva, K., and Tablan, V. (2002). A framework and graphical development environment for robust nlp tools and applications. In *ACL*, pages 168–175.

Deng, Y., Kanervisto, A., Ling, J., and Rush, A. M. (2017). Image-to-markup generation with coarse-to-fine attention. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pages 980–989. JMLR.org.

Eijkhout, V. (1991). *TEX by topic: a TEXnician's reference*. Addison-Wesley UK. Version 1.1, 2007.

Gehrke, J., Ginsparg, P., and Kleinberg, J. (2003). Overview of the 2003 kdd cup. *Acm SIGKDD Explorations Newsletter*, 5(2):149–151.

Jat, S., Khandelwal, S., and Talukdar, P. (2018). Improving distantly supervised relation extraction using word and entity based attention. *arXiv preprint arXiv:1804.06987*.

Klein, G., Kim, Y., Deng, Y., Senellart, J., and Rush, A. (2017). OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada, July. Association for Computational Linguistics.

Kluegl, P., Toepfer, M., Beck, P.-D., Fette, G., and Puppe, F. (2016). Uima ruta: Rapid development of rule-based information extraction applications. *Natural Language Engineering*, 22(1):1–40.

Kristianto, G. Y., Aizawa, A., et al. (2014). Extracting textual descriptions of mathematical expressions in scientific papers. *D-Lib Magazine*, 20(11):9.

Kristianto, G. Y., Topić, G., and Aizawa, A. (2017). Utilizing dependency relationships between math expressions in math ir. *Information Retrieval Journal*, 20(2):132–167, Apr.

Laurens, J. (2008). Direct and reverse synchronization with synctex. *TUGBoat*, 29:365–371.

Marsi, E. and Öztürk, P. (2015). Extraction and generalisation of variables from scientific publications. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 505–511, Lisbon, Portugal, September. Association for Computational Linguistics.

Nguyen, T. H. and Grishman, R. (2015). Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 39–48.

Passonneau, R. (2004). Computing reliability for coreference annotation. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*.

Pontelli, E., Gupta, G., and Karshmer, A. I. (2009). Mathematics and accessibility: a survey. In *Universal Access Handbook*. CRC Press.

Pyarelal, A., Valenzuela-Escárcega, M. A., Sharp, R., Hein, P. D., Stephens, J., Bhandari, P., Lim, H., Debray, S., and Morrison, C. T. (2019). Automates: Automated model assembly from text, equations, and software. *Modeling the World's Systems*.

Quoc, M. N., Yokoi, K., Matsubayashi, Y., and Aizawa, A. (2010). Mining coreference relations between formulas and text using wikipedia. In *Proceedings of the Second Workshop on NLP Challenges in the Information Explosion Era (NLPIX 2010)*, pages 69–74.

Schubotz, M., Grigorev, A., Leich, M., Cohl, H. S., Meuschke, N., Gipp, B., Youssef, A. S., and Markl, V. (2016). Semantification of identifiers in mathematics for better math information retrieval. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 135–144. ACM.

Schubotz, M., Greiner-Petter, A., Scharpf, P., Meuschke, N., Cohl, H. S., and Gipp, B. (2018). Improving the representation and conversion of mathematical formulae by considering the textual context. *TUGBoat*, 39(3):228–240.

Sharp, R., Pyarelal, A., Gyori, B., Alcock, K., Laparra, E., Valenzuela-Escárcega, M. A., Nagesh, A., Yadav, V., Bachman, J., Tang, Z., et al. (2019). Eidos, indra, & delphi: From free text to executable causal models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 42–47.

Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S., and Tsujii, J. (2012). brat: a web-based tool for nlp-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107. Association for Computational Linguistics.

Sterckx, L., Caragea, C., Demeester, T., and Develder, C. (2016). Supervised keyphrase extraction as positive unlabeled learning. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1924–1929, Austin, Texas, November. Association for Computational Linguistics.

Valenzuela-Escárcega, M. A., Hahn-Powell, G., and Surdeanu, M. (2016). Odin's runes: A rule language for information extraction. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 322–329.

Yokoi, K., Nghiem, M.-Q., Matsubayashi, Y., and Aizawa, A. (2011). Contextual Analysis of Mathematical Expressions for Advanced Mathematical Search. *Polibits*, pages 81 – 86, 06.

Zeng, D., Liu, K., Chen, Y., and Zhao, J. (2015). Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1753–1762.