Syntactic Role Identification of Mathematical Expressions

Xing Wang, Jason Lin, Ryan Vrecenar, Jyh-Charn Liu*
Department of Computer Science and Engineering
Texas A&M University
College Station, USA
{wxadb, senyalin, ryanvrecenar, jcliu}@tamu.edu

Abstract*— This paper presents a prediction algorithm to infer the syntactic role (SR) of mathematical expressions (ME), or SR$_{ME}$, in ME-plaintext mixed sentences. SR$_{ME}$ is a predicted syntax label of ME, which could be integrated into any constituent parser to improve their accuracy in sentence parsing. SR$_{ME}$ is based upon three features of ME placement in a sentence:** properness of **S**entence structure (feature F3), properties of **M**E (feature F2), and PoS of the **L**ocal neighbor plain text (feature F1). ***An inside-outside inspired algorithm is proposed for SR$_{ME}$ by maximizing the probability of a relaxed parsing tree. Features in F2 was found to fit into both exponential and Poisson distributions, which could fuse with other features to re-weight the prediction rule that improves the prediction precision for SR$_{ME}$ as a noun phrase (noun modifier) by 3.6% (18.7%). F1, F2, and F3 were found to complement each other. Significant discriminative patterns on the part-of-speech (PoS) of the neighbor plaintext are adopted to build a Naïve Bayesian classifier, which is fused with the F3 baseline that improved the precision of the prediction of SR$_{ME}$ as a sentence by 10%. The overall error rate of the SR$_{ME}$ prediction algorithm was found to be 15.1% based on an experiment using a public ME-plaintext mixed parsing tree data set provided by Elsevier.

Keywords—NLP; parsing; syntactic role; mathematical expressions; ME-plaintext mixed.

I. Introduction

Mathematical expressions (ME) are organic ingredients of many scientific publications. As one part of mixed sentences, they are used in certain specific ways to deliver technical substances not necessarily conformed to the non-technical writing practices. In parsing of a sentence s to generate a parsing tree for s, an ME needs to be assigned a non-terminal node label from the set of nonterminal node labels $\mathcal{M}$ in a grammar $\mathcal{G}$. With much of the mainstream natural language processing (NLP) literatures and tools devoted to general articles, a missing opportunity is to improve the accuracy of parsing algorithms by identification of the *syntactic roles* (SR) of MEs based on technical writing practices. Using the features and their statistic models trained from a set of hand labelled dataset to represent the technical writing practices, SR is a probabilistic assertion of the most likely label(s) in $\mathcal{M}$ for an ME in sentence under analysis. The modeling process also takes into account the syntactic properness of the sentence based on probability maximization over the relaxed parsing tree using a pre-trained PCFG grammar [7]. This way, SR can help higher level NLP parsing mechanisms more accurately perform their functions in ME-plaintext mixed technical articles.

While largely following regular writing styles technical writers regularly create field related expressions, ranging from terminologies, symbols, phrases to deliver the technical contents. Among those complex issues, we focus on the syntactical roles of MEs, or how may MEs be used in the construction of a sentence. For instance, in the dataset [17] an ME is often used as a *noun phrase* (label: **NP**), but it can also be used as a *sentence* (label: **S**) or *noun modifier* (label: **NML**), yet in regular writing a word/phrase is rarely labeled as an **S**.

Knowing that writing of an ME-plaintext mixed sentence is involved with multiple factors in how to convey the technical contents, which then affect the SR of an ME (SR$_{ME}$), we propose the SML model based upon properness of **S**entence structure (feature F3), properties of **M**E (feature F2), and PoS of the **L**ocal neighbor plain text (feature F1). F1 is the frequency of an ME's neighbors in the Part of Speech (PoS) of a sentence. F2 is the structural complexity properties of an ME, including its variable count, operator count, and the depth of the operation tree represented in MathML format. And F3 is a sentence level, grammar based probabilistic assertion of the SR$_{ME}$. We note that our design process can be readily tailored to different numbers and types of features that can be used to characterize SR$_{ME}$ for different fields.

The statistical patterns of the three features are extracted from the hand labeled dataset [17][21] for the design of the prediction rules. Succinctly put, F1 produced useable performance result for the prediction of SR$_{ME}$ → **NP,** and F2 is most effective in the prediction of SR$_{ME}$ ↛ **NML**. F3 had similar performance as F1 in SR$_{ME}$ → **NP,** and it has significantly better performance in prediction of SR$_{ME}$→**S**. The final prediction outcome fused from individual prediction results produced 15.8% of performance gain. The final prediction scheme by fusing three models achieved an error rate of 15.1% for classifying ME as **S** or **NP**/**NML**. The prediction outcomes can be used for initialization of most likely ME role in constituent parsers.

The rest of the paper is organized as follows: related work is presented in section II; our model is explained in section III; experiment and analysis are given in section IV; the paper is concluded in section V.

[*] Correspondence author.

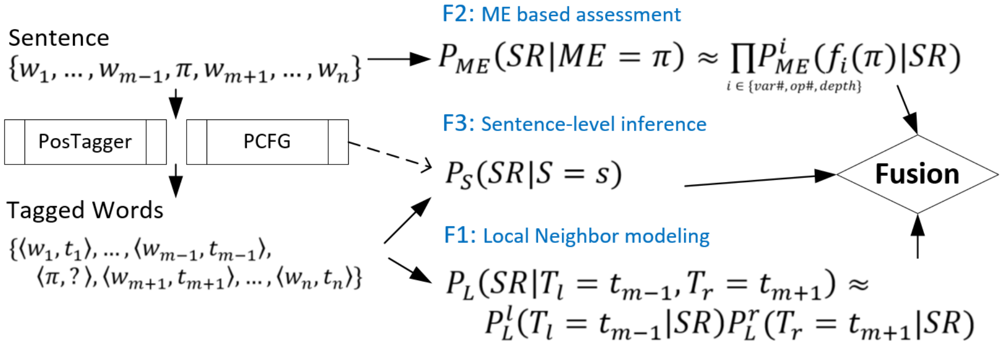

F2: ME based assessment
Sentence
$\{w_1, \dots, w_{m-1}, \pi, w_{m+1}, \dots, w_n\}$
$P_{ME}(SR|ME = \pi) \approx \prod_{i \in \{var\#, op\#, depth\}} P^i_{ME}(f_i(\pi)|SR)$
PosTagger
PCFG
F3: Sentence-level inference
$P_S(SR|S = s)$
Fusion
Tagged Words
$\{\langle w_1, t_1\rangle, \dots, \langle w_{m-1}, t_{m-1}\rangle, \langle \pi, ?\rangle, \langle w_{m+1}, t_{m+1}\rangle, \dots, \langle w_n, t_n\rangle\}$
F1: Local Neighbor modeling
$P_L(SR|T_l = t_{m-1}, T_r = t_{m+1}) \approx P^l_L(T_l = t_{m-1}|SR) P^r_L(T_r = t_{m+1}|SR)$

Fig. 1. System digram for the SML SR prediction model

II. Related Work

There are several works on the mathematical discourse analysis. However, most of the existing approaches treat the parsing as a black box and apply conventional natural language parser on sentences with mixed ME-plaintext. Kristianto [1] and Moritz [2] replace the ME with a random string and feed the sentences to Stanford parser. Magdalena [3] presents a qualitative study on the usage of a mixture of natural language and symbols by studying the chatting log of geometry discussion. Mohan [4] adopt a typed-feature grammar approach to solve the ambiguity during parsing, which emphasized the usage of type ontology as consistency constraints. But they don't handle the different syntactic roles of ME.

Parsing has been one major topic of NLP for more than 20 years, and the formulations could be categorized as shallow parsing (chunking) [5], constituent parsing [6][7], and dependency parsing [8][9]. For the task of parsing ME-plaintext mixed sentences, constituent parsing is more suitable for the following reasons. First, scientific elaborations are expressing a complex relationship that is hard to capture by shallow parsing as it only gives chunks of subsequence such as noun phrase. As for the dependency parsing, there are not enough training data to train the rule of actions as [9] and no existing well-defined dependency relation ontology acknowledge by the community beforehand. The constituent parsing is easy to understand and could be easily modified and extended. One possible extension direction is the categorical combinatorial grammar (CCG), which is commonly used to transform the natural language into first order logic or lambda calculus [10].

Although the existing parser is not perfect, it has already shown potential to enhance the semantic of ME [11]. There have been a few efforts in extracting the variable definition or description [1][2][12]. A better parser for the ME-plaintext mixed sentences would be beneficial to boost the research on machine reading [13] and scientific knowledge management.

III. SML Model

In this section, we will first introduce the basic knowledge of parsing in subsection III.A, followed by an explanation of the SR of ME according to the dataset by Elsevier (subsection III.B). Then we will discuss the SML model, which is illustrated in Fig.1, based on its three features: properness of **S**entence structure (subsection III.D), properties of **M**E, and PoS of the **L**ocal neighbor plain text (subsection III.C).

A. Basics of ME-plaintext Parsing

In this subsection, we will briefly introduce the relationship among the Probabilistic Context Free Grammar (PCFG), parsing tree, and the relation between a sentence and parsing tree for ME-plaintext mixed sentences. A sentence s composed of a linear sequence of words $\{w_1, \dots, w_{m-1}, \pi, w_{m+1}, \dots, w_n\}$ with ME π in position m. There could be multiple MEs in a sentence, and we only show one in the example for convenience. For each word w_i, $i \neq m$, its corresponding PoS tag t_i will be identified, e.g., noun (NN), determinate (DT), and verb (VB).

A PCFG is defined by a quintuple $\mathcal{G} = (\mathcal{M}, \mathcal{T}, \mathcal{R}, \mathbb{s}, P)$, where $\mathcal{M}$ is the set of non-terminal symbols, $\mathcal{T}$ is the set of terminal symbols, $\mathcal{R}$ is the set of production rules, $\mathbb{s} \in \mathcal{M}$ is the starting symbol, and $P: \mathcal{R} \to [0,1]$ is the probability of each rule. $\mathcal{M}$ contains all PoS tag and some other larger syntactic structure such as noun phrase (NP), verbal phrase (VP), and preposition phrase (PP). Each rule $r \in \mathcal{R}$ is composed of a left-hand side lhs_r and a right-hand side rhs_r, where $lhs_r \in \mathcal{M}$, and the rhs_r is a sequence of symbols in the product space $\{\mathcal{M} \cup \mathcal{T}\}^{|rhs_r|}$. There is a constraint over the probability that the sum of probability of rules with the same left-hand side non-terminal node m equals 1, i.e., $\sum_{r:lhs_r = m} P(r) = 1$.

A parsing tree t is composed of non-leaf nodes $\bar{\mathcal{L}}(t)$ and leaf nodes $\mathcal{L}(t)$, which should satisfy the following condition with respect to the PCFG $\mathcal{G}$:

1) $n.val \in \mathcal{M}, \forall n \in \bar{\mathcal{L}}(t)$.

2) $n.val \in \mathcal{T}, \forall n \in \mathcal{L}(t)$,.

3) For every node n, there is a production rule $r \in \mathcal{R}$ with $lhs_r = n.val$ and the $rhs_r = \{c.val | c \in n.children\}$, which is denoted as r_n the production rule mentioned above. Take the node "VP" with children node "VBZ" and "NP" in Fig. 3 for example, it satisfies the condition because there is a rule $VP \to VBZ\ NP$.

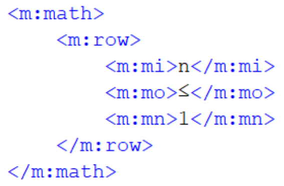
4) The value of the root node $n^{root}_t.val = \mathbb{s}$.

The sentence generated from a parsing tree t is obtained by left first traverse of the leaf nodes. For any parsing tree t, the probability of observing it is $P(t) = \prod_{n \in t} P(r_n)$.

B. SR of ME

TABLE I. ME-PLAINTEXT SENTENCE EXAMPLES AND THEIR SR

SR	Sentences
S	"Fix a prime and let $\boldsymbol{n} \in \boldsymbol{N}$." "Note that $[\boldsymbol{f}]\boldsymbol{p} = [\boldsymbol{f_0}]\boldsymbol{p}$ and $[\boldsymbol{f}']\boldsymbol{p} = [\boldsymbol{f}'_0 - \boldsymbol{f_1}]\boldsymbol{p}$."
NP	"Let $\boldsymbol{G} = \boldsymbol{limG_n}$ be the projective limit of this system." "We are given a graph $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$. "
NML	"This happens $\boldsymbol{lgn}$ times by repeating squaring. "



```
<m:math>
    <m:row>
        <m:mi>n</m:mi>
        <m:mo>≤</m:mo>
        <m:mn>1</m:mn>
    </m:row>
</m:math>
```

Fig. 2. An example of MathML representation

For prediction of the SR of ME π among the nonterminal symbols $\mathcal{M}$, denoted as SR_π, we follow the convention from the label of the Elsevier data, which categorized the ME role into three categories: **S** (sentence or subordinate clause), **NP** (noun phrase), **NML** (noun modifier). Some examples of these labels are shown in Table I. A hint on the complexity of the label assignment problem for SR_π can be illustrated by these examples. For instance, the equal sign could be interpreted as a verb "equal" in the second case for **S**, where $[f]p = [f_0]p$ is translated into "$[f]p$ equals $[f_0]p$". It could also be interpreted as a subordinate clause in the first case of **NP**, where $G = limG_n$ is translated into "G which is lim G_n". In another example, the first case of S and NP, the left words are the same, but the SR differ due to words after the ME. Besides acting as a noun or noun phrase, the ME could also be part of the compounded noun acting as a modifier. In the last row of table I, lgn is a quantity that modifying the noun "times".

C. Feature F1 (neighbor words) and F2 (ME properties)

The first prediction heuristic is based on the intuition that the SR of a complex (simple) ME is more likely to be **S** (**NP**/**NML**), where the complexity of an ME is based on 3 features: the number of variables and operators, and the depth of the expression operation tree (EOT), which represents the mathematical operations of an ME.

Among different options, the EOT of an ME can be represented in the Latex, XML or other similar formats, but the widely used Elsevier dataset uses Unicode to represent MEs and the Penn Tree annotation do not support this feature. As such, we manually translated the ME into *Latex* format [21], which is then converted into MathML format using LateXML [15] with one example shown in Fig. 2. The number of variables is the number of "m:mi" tags; The number of operators is the number of "m:mo" tag; The depth of the expression is the depth of the root nodes. For the example in Fig. 2, the value for the three features are 1, 1, 3 respectively. Given an ME π and the corresponding ME property features $\{f_\pi^j\}$, the conditional probability of the SR is estimated as: $P_{ME}\left(SR_\pi \middle| \{f_\pi^j\}\right) \approx \prod_j P_{ME}^j(f_\pi^j|SR_\pi)$.

Next, we discuss how to use the PoS tag of the neighbor words of an ME to infer its SR$_{\text{ME}}$ based on the intuition that the usage of ME in a sentence is not random. For a sentence $\{w_1, \dots \mathrm{w}_{\mathrm{m}-1}, \pi, w_{m+1}, \dots, w_n\}$, we would like to estimate the probability of SR_π based on the neighbor $P_L(SR_\pi|t_{\mathrm{m}-1}, t_{m+1})$, where $t_{\mathrm{m}-1}$ is the PoS tag of the word w_{m-1} if $m - 1 > 0$ else $t_{i-1} = "BEG"$, and t_{m+1} is the PoS tag of the word $w_{\mathrm{m}+1}$ if $m + 1 \leq n$ else $t_{m+1} = "END"$. The assessment is conducted through Bayesian transform and with the assumption of conditional independency:

$$P_L(SR|T_l = t_{m-1}, T_r = t_{m+1}) \\ \approx P_L^l(T_l = t_{m-1}|SR)P_L^r(T_r = t_{m+1}|SR)$$

D. Feature F3 (Sentence based Inference)

The goal of F3 is to find the SR of MEs based on the properness of a sentence structure. We will first present the conventional CYK algorithm [16], which calculates inside probability α, and explain why the missing SR invalid CYK algorithm. Then we show our formulation of SR prediction as an optimal relaxed parsing tree searching problem. A Relaxed Tree probability Maximization (RTM) algorithm is proposed to solve the search problem for the cases when there is a single ME in a sentence.

The grammar is often represented in Chomsky Norm Form (CNF) form, which only allows two types of production rule: $A \to BC$, or $A \to a$, where $A, B, C \in M$ and $a \in T$. The CYK algorithm on CNF grammar builds the α inside probability using dynamic programming (DP) method, where $\alpha(A, i, j)$ stores the highest probability of the parsing tree with root as non-terminal symbol A covering words in range $[i, j]$ of the sentence:

$$\alpha(A, i, j) = \max_{A \to BC, k \in [i,j)} P(A \to BC)\alpha(B, i, k)\alpha(C, k + 1, j). \quad (1)$$

The inside probability is calculated iteratively with increasing range length $j - i$, and $\alpha(A, i, i)$ is initialized based on unary rules.

However, we face a challenging situation that if the SR of ME position is unknown, one cannot build the constituent in a bottom-up approach. Although we could try all states at the ME position, it is inefficient as there are 12976 states even for the Stanford unlexicalized PCFG parser [7]. Moreover, the assignment of the prior probability is still an open problem.

In this work, we define the probability of SR of ME(s) conditioned on the whole sentence s given a PCFG $\mathcal{G}$ by the largest probability of feasible relaxed parsing tree (RPT):

$$P_S(SR = \overrightarrow{sr}|S = s) = \max_{t \in \mathbb{T}_{\overrightarrow{sr}}(s)} P(t),$$

where an RPT is defined to handle the unknown SR$_{\text{ME}}$ by relaxing the second condition of a parsing tree under the grammar PCFG $\mathcal{G}$ discussed in III.A. That is, the value of a leaf node $\boldsymbol{n} \in \mathcal{L}(\boldsymbol{t})$, $\boldsymbol{n.val}$ could be non-terminal, so that we are relieved from the rigor of building a parsing tree before the SR of an ME is known. This way, we have the full freedom in searching for the most likely SR of an ME in the relaxed search space. The actual construction of a parsing tree can be done by the constitute parser using the SR outcomes as its initialization step. In PRT, all leaves are traversed from left to right, and the symbol sequence is the same with the sequence of words w_i in a sentence s except the position where MEs locate. We denote $\mathbb{T}_{\overrightarrow{sr}}(s)$ as all RPTs for a sentence s with constraints on the SR of MEs represented as a sequence $\overrightarrow{sr}$, where sr_i as the SR of the

i^{th} ME after left first traversal. One example of such RPT is shown in Fig. 3, with $\overrightarrow{sr} = \{S\}$.

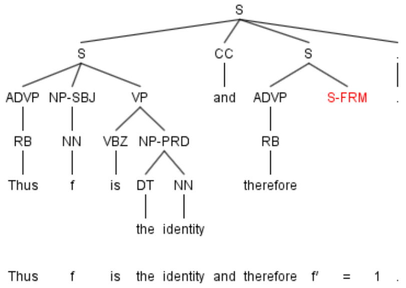


Fig. 3. An example of a relaxed parsing tree visualized by the Tsurgeon tool.

The large number of possibilities for $\overrightarrow{sr}$ still makes it cost prohibitive to search for the optimal relaxed parsing tree $\mathbb{T}_{\overrightarrow{sr}}(s)$ for a sentence s based on enumeration. Due to its complexity, we will focus the case that a single ME is in s in this work. Inspired by the inside-outside algorithm [20], we propose a Relaxed Tree probability Maximization (RTM) algorithm that uses dynamic programming to calculate the *inside* and *outside probability* to infer the probability of candidates syntactic roles.

Given the inside probably $\alpha(.)$ described in (1), the outside probability is defined as $\beta(A,i,j) = \max_{t \in OTS(A,i,j)} P(t)$, where $\beta(.)$ is the outside probability; $OTS(A,i,j)$ the outside tree set which has leaves sequence covering the words in the range $[1, i-1]$ and range $[j+1, n]$; and notation A is the tree node corresponding to words in range $[i,j]$. For the single ME located at position m, $OTS(SR,m,m)$ is equivalent to $\mathbb{T}_{\overrightarrow{sr}}(s)$. And we have the probability of the SR of the ME as $P_S(SR|S=s) = \beta(SR,m,m)$.

Following the DP optimization process, the outside probability is iteratively updated with decreasing of the $j-i$ range, starting with the initialization of $\beta(\mathbb{s}, 1, n) = 1$:

$$\beta(A,i,j) = \max(\\ \max_{B \to AC, k \in [j+1,n]} P(B \to AC)\beta(B,i,k)\alpha(C,j+1,k), \\ \max_{A \to BC, k \in [i,j)} P(A \to BC)\alpha(B,i,k)\alpha(C,k+1,j))$$

, where "$B \to AC$","$A \to BC$" $\in \mathcal{R}$ are production rule in the PCFG $\mathcal{G}$ [7], and $A, B, C \in \mathcal{M}$ are non-terminal symbols.

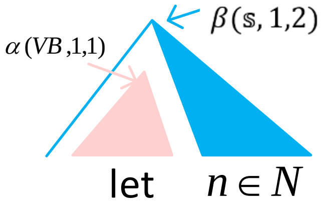


Fig. 4. RTM based inside/outside probability vs search space of a sentence.

One example of the inside-outside process is given in Fig. 4 for the sentence "let $n \in N$". We could first calculate the inside probability in a bottom-up approach. In the example, we could get $\alpha(VB,1,1)$ as the highest probability of the first word with role "VB" for the word "let". Then, we could also get the outside probability $\beta(\mathbb{s},1,2)$ as the highest probability by treating range [1,2] as "S" and remaining words are covered by the tree. Further, if we have a grammar rule $\mathbb{s} \to VB\ SBAR$. We could make inference about the probability of ME as "$SBAR$" as $\beta(\mathbb{s},1,2) * \alpha(VB,1,1) * P(\mathbb{s} \to VB\ SBAR)$.

The prediction outcomes of the RTM algorithm are the states in the PCFG parser, which contains than the three annotated labels of (**NP**, **S**, **NML)** in the Elsevier dataset. As such, we manually map the states in $\mathcal{M}$ into the SRs as follow:

- JJ, JJR, JJS, QP, ADJP ➔ **NML**
- NN, NNS, NNP, NNPS, NP ➔ **NP**
- S, SBAR, SBARQ, SINV ➔ **S**

All other states are removed from the SR candidates. More details on PoS and syntactic tags can be found in [19].

IV. Experiment

We will first present the data set and the evaluation criteria. We will then explore of the relation between the complexity properties and the SR of ME. A histogram based distribution fitting of ME properties will be presented. Third, the PoS tag of the neighbor words are explored to assist the prediction, and a Naïve Bayes classifier is built to show the discriminate ability. Finally, we will show the performance results of the sentence-level assessment and the fusion with the other two features.

A. Dataset and evaluation criteria

OS-STM-Elsevier [17] provides 10 papers with parsing tree annotated in Penn tree format [18]. Each paper is from a different technical field. Each sentence is annotated with a parsing tree exemplified by Fig. 3, in which the ME is labeled as "*-FRM", where the "*" could be replaced with a more specific label based on its SR, i.e., **NP**, **S**, **NML**. Among 2757 sentences, there are 346 sentences with a mixture of ME and plaintext. Sentences containing only one MEs are removed from experiment as they represent the trivial case that their only viable SR is **S**. For the study related to ME properties and local neighbor features; all the 330 sentences are used. For experiment related with the sentence based inference, we only use the 196 sentences containing a single ME due to the RTM model limitation on the outside probability formulation. The statistic for different SRs in the dataset [17] is shown in Table II.

TABLE II. Statistics of the SR in the dataset [17]

	S	NP	NML
All ME	75	448	77
One ME	30	145	24

The prediction results are presented in the *primitive confusion matrix* (CM) format shown in Table V and VI. A CM can be easily translated into other evaluation criteria such as precision, recall, and error rate. In CM, each row represents the samples labeled as the SR in the row header in the ground truth, and each column represents the samples predicted by our algorithm as the SR in the column header.

An ME with the label **NP** or **NML** could be used alternatively to construct compounded noun phrases. This means for MEs playing a decorative function in noun phrase such as the last row in Table I, predicting the SR of ME as **NP** or **NML** will not affect the overall sentence parsing tree. Given the reason above, we are more concerned about the error rate of **S** versus

NP/**NML**. The numbers highlighted in bold font in Table V and VI show a few significantly improved performance numbers, whose details will be explained shortly.

B. *ME properties exploration and distribution fitting*

Figure 5 shows the cumulative bar charts of SR labels with respect to elements of the ME complexity feature, i.e., F2, the number of variables and operators, and the structural depth. The blue/red/ light blue bars indicate the bin counts of MEs with the SR of **NP**/**S**/**NML**. While it is infeasible to perform discriminant analysis due to the overall significant overlaps of the three SR labels, we observe two useful patterns: First, the feature value for **NML** are mostly smaller than that of the other roles. Second, the number of variables and operators can fit the exponential distribution, while the depth value the Poisson distribution. As such, we use the distribution fitting technique to estimate the parameter explained in the rest of this section. The probability density/mass function (PDF/PMF) will be used to calculate the posterior probability of each SR based on the ME feature values.

For fitting of the exponential distribution for the numbers of variables and operators, the PDF of the exponential distribution of feature j under a SR value sr is $p_{sr,j}^{exp}(\mathrm{x}) = \lambda_{sr}^{j} e^{-\lambda_{sr}^{j} x}$, where the parameter λ_{sr}^{j} is estimated as $|\{\pi: SR_\pi = sr\}| / \sum_{\pi: SR_\pi = sr} f_\pi^j$, where π is an ME instance, SR_π is the SR of the ME in the sentence, and f_π^j is the value of feature j for the ME π. The $|\cdot|$ operator is the size of a set.

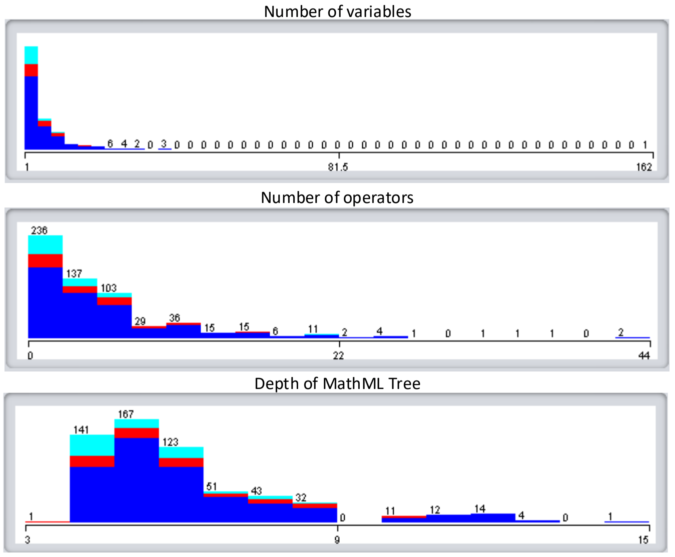


Fig. 5. Cumulative histograms for ME features

For fitting of the Poisson distribution for the structural depths, the PMF of the Poisson distribution of the feature j under a SR value sr is $p_{sr,j}^{poi}(k) = \frac{{\gamma_{sr}^{j}}^{k} e^{-\gamma_{sr}^{j}}}{k!}$, where the parameter γ_{sr}^{j} is estimated as:$\sum_{\pi: SR_\pi = sr} f_\pi^j / |\{\pi: SR_\pi = sr\}|$.

C. *Local Neighbor Features*

The PoS tags of the left hand side (LHS) and right hand side (RHS) of an ME offer valuable hints on the likely SR value of an ME, due to the constraining nature of writing. Based on statistic reported in Table III and IV, we came up with several useful predictive heuristics:

PoS tag	SR of ME
VBZ/VBP/ NN/NNS (LHS)	never **S**
DT (LHS)	always NML
VBZ, DT, NN, NNS (RHS)	likely to be NP or NML

Based on these rules, we design a Naïve Bayesian (NB) classifier, which produced the confusion matrix of classification on all MEs shown in Table V. Overall, the classifier is found to be effective for the majority of the class **NP**. The precision for **S** is very low, with a significant number of miss prediction of **NP**. The recall rate for the **S** role is lower than 50%. As such, it necessitates the sentence level RTM algorithm to make final prediction, whose results are reported next.

TABLE III. CO-OCCURRENCE OF SR AND POS OF **LEFT** NEIGHBOR

	VBZ	*NN*	*DT*	*VB*	*RB*	*VBP*	*NNS*
NP	48	32	0	8	15	11	11
S	0	0	0	13	5	0	0
NML	7	7	26	0	0	7	6

TABLE IV. CO-OCCURRENCE OF SR AND POS OF **RIGHT** NEIGHBOR

	VBZ	*)*	*(*	*WRB*	*NN*	*DT*	*NNS*
NP	38	22	19	17	1	14	0
S	0	1	2	2	0	0	0
NML	4	5	1	0	17	0	10

TABLE V. CONFUSION MATRIX OF NB CLASSIFIER USING LOCAL NEIGHBOR FEATURES

	NP	**S**	**NML**
NP	**250**	98	58
S	44	31	0
NML	16	8	53

D. *Sentence-level inference and fusion*

Table VI showed the SR prediction result for the sentence-level RTM algorithm and its fusion with other two features. The table is based on 192 successfully tested sentences out of the 196 sentence with only one ME. In particular, we find that the sentence level inference significantly improves the recall rate of the SR **S**, with improved the precision.

TABLE VI. CONFUSION MATRIX OF SENTENCE-LEVEL ANALYSIS

	Sentence only			**Sentence+local**			**Sent.+ local+ME**		
	NP	**S**	**NML**	**NP**	**S**	**NML**	**NP**	**S**	**NML**
NP	**89**	39	11	94	**25**	21	**105**	24	**11**
S	6	21	1	6	22	0	5	23	0
NML	18	2	4	19	2	3	17	2	5

The improvement of the precision of **S** prediction is likely attributed to the complemtary nature of different local features. When integrated at the sentence level model, prediction blind spots of low level predictors were effectively compensated by other modules to produce overall much better prediction results that otherwise cannot be produced individually. Specifically, fusion of F1 and F3 still has low performance for **NML** prediction. Knowing that the F2 are effective in predicting

$\mathrm{SR}_{\mathrm{ME}} \nrightarrow$ **NML** when the feature value is very high (see Fig. 5). We fused F2 with the F1 and F3 that resulted in better precision for **NML** prediction. It also leads to better precision and recall for **NP** prediction. The error rate of the fusion model of three features is 30.7%. If we aggregate **NP** and **NML** into one group as discussed in the evaluation criteria (IV.A), the error rate is cut to 15.1%.

The SML model and the RTM algorithm is not perfect. To gain more insights on the nature of incorrect predictions, we made some selective case studies. The first quick observation is that hand labelled tags are not error free. For the 23 false-positive samples in **S** prediction, 13 of them have been incorrectly hand mislabelled as **NP** but they really should be assigned the label **S.** For instance, some of these cases are in the form of "[Therefore | otherwise | so], ME", where the "ME" should be labeled as role **S**, not the incorrectly labeled **NP**. Some other cases also demonstrated the limitation of the SML model. In the three common patterns of "let ME" (2 cases), "if ME" (1 case), "*** that ME" (1 case), the ME was falsely predicted to be NP while their hand labels are **S.** The error suggests that one may want to create additional features in the decision rule to make corrections: if there is only one ME, it should be labelled as **S**; if there are other VP structure after the ME, the ME should be treated as a subject and labelled as **NP**, and so on.

E. Implementation Detail

We built Java code based on the open source Stanford CoreNLP framework. The unlexicalized PCFG [7] model (updated at 2016-10-31) is used as the grammar input for our RTM algorithm. With no loss of its accuracy, we did not initialize the $\alpha(A, i, i)$ of position i for an ME in the alpha probability calculation to reduce the search space.

For F3, the PoS tag for each word is either looked up in the lexicon in the Stanford Unlexicalized Parser [7], or identified through the Stanford PoS tagger [14]. For F1, we first used the hand labelled PoS tags to infer the conditional probabilities that produced results in Tables V and VI. We then further used the predicted PoS tags from the Stanford PosTagger [14] to compare their effects on the RTM algorithm. The difference of the two approaches is negligible.

V. Conclusion and Discussion

In this work, we propose the SML predictive model for $\mathrm{SR}_{\mathrm{ME}}$ to predict the syntactical role of an ME in a ME-plaintext mixed sentence. Results show that the effectiveness of the sentence-level modeling in prediction the SR. The ME property and local neighbor are helpful in enhancing the performance. The error rate of SR prediction under **NP**/**NML** vs. **S** classification is 15.1%, which is reliable enough to integrate with the existing parsing tools.

Since this is a preliminary study, more experiments are required to validate the conclusion of this work in a larger data set. The experiments in this paper are not done with cross-validation due to the high computational cost of the RTM algorithm. The future algorithm will need to consider handling sentence with multiple MEs and accelerating the process speed for deployment in the real-world applications.

References

[1] G. Y. Kristianto, M.-Q. Nghiem, Y. Matsubayashi, and A. Aizawa, "Extracting definitions of mathematical expressions in scientific papers," Proc. of the 26th Annual Conference of JSAI, 2012.

[2] R. Pagael and M. Schubotz, "Mathematical language processing project," arXiv preprint arXiv:1407.0167, 2014.

[3] M. Wolska and K.-K. Ivana, "Analysis of mixed natural and symbolic language input in mathematical dialogs," Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics, 2004.

[4] M. Ganesalingam, "The Language of Mathematics," Springer, Berlin, 2013.

[5] L.A. Ramshaw and P.M. Mitchell, "Text chunking using transformation-based learning," Natural Language Processing Using Very Large Corpora, Springer Netherlands, 1999, pp. 157-176.

[6] E. Charniak and J. Mark, "Coarse-to-fine n-best parsing and MaxEnt discriminative reranking," Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, 2005, pp. 173-180.

[7] D. Klein and C.D. Manning, "Accurate Unlexicalized Parsing," Proceedings of the 41st Meeting of the Association for Computational Linguistics, 2003, pp. 423-430.

[8] J. Nivre, J. Hall, J. Nilsson, A. Chanev, G. Eryigit, S. Kübler, S. Marinov, and E. Marsi, "MaltParser: A language-independent system for data-driven dependency parsing," Natural Language Engineering 13.2 (2007): 95-135.

[9] D. Chen and C.D. Manning, "A fast and accurate dependency parser using neural networks," Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 2014.

[10] L.S. Zettlemoyer and M. Collins, "Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars," arXiv preprint arXiv:1207.1420, 2012. Kristianto, Giovanni Yoko, et al. "Extracting definitions of mathematical expressions in scientific papers." *Proc. of the 26th Annual Conference of JSAI*. 2012.

[11] M. Schubotz, et al. "Semantification of identifiers in mathematics for better math information retrieval," Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval. ACM, 2016.

[12] K. Yokoi, et al. "Contextual analysis of mathematical expressions for advanced mathematical search," Polibits 43 (2011): 81-86.

[13] R. Cohen. "DARPA's Big Mechanism program," Physical biology 12.4 (2015): 045008.

[14] K. Toutanova and C.D. Manning. "Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger, " In Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000), pp. 63-70.

[15] B. Miller. "LaTeXML: ALatex to xml converter," Web Manual at http://dlmf. nist. gov/LaTeXML/, seen September2007 (2010).

[16] T. Kasami (1965). "An efficient recognition and syntax-analysis algorithm for context-free languages (Technical report)," AFCRL 1965. pp. 65-758

[17] https://github.com/elsevierlabs/OA-STM-Corpus/

[18] E. Charniak. "Tree-bank grammars," Proceedings of the National Conference on Artificial Intelligence. 1996. APA

[19] M. Marcus, M. Marcinkiewicz, and B. Santorini. "Building a large annotated corpus of English: The Penn Treebank," Computational linguistics 19.2 (1993): 313-330.

[20] M. Collins. "The Inside-Outside Algorithm," Lecture Notes (2013).

[21] Xing Wang, Jason Lin, Ryan Vrecenar, "Manual transcription from unicode value to Latex for ME in Elsevier OT-STM-Corpus TreeBank, in https://github.com/elsevierlabs/OA-STM-Corpus", web posting: http://rtds.cse.tamu.edu/resources/, posted at Aug. 2017.