Master ATAL

H. Mouchère / Harold.Mouchere@univ-nantes.fr

Reconnaissance multi-modale (X3ITM40)

IPI team

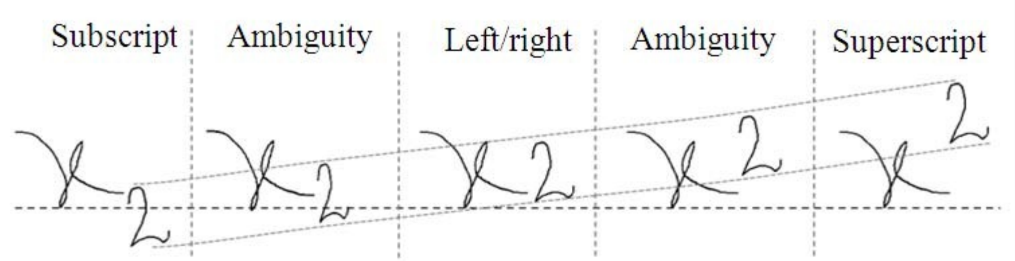
Image, Perception et Interaction

ML for Document Analysis

- Several computer vision tasks are not simple classification problems
  - Several sub tasks and sub steps
- Document Analysis is one of these complex topics
  - Character / word recognition
  - Text line segmentation and recognition
  - Word spotting in handwritten text / natural images
  - Document structure analysis (reading path, image / text segmentation)
  - Math recognition
  - ...

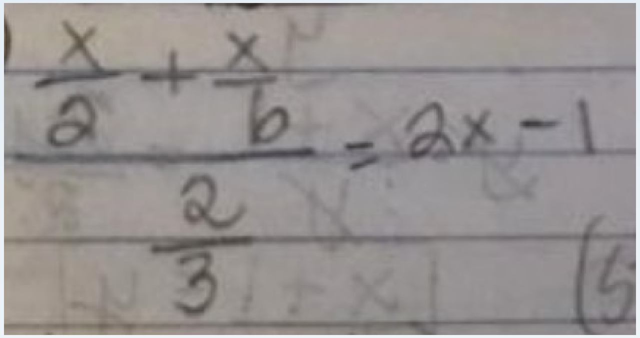
Math Recognition

- Why Math Reco ?
  - Widespread use of math expressions (universality)
  - Very interesting scientific challenges during automatic processing
    - Very high number of symbols (≈ 250)
    - Possibility of very complex 2D symbol organization
  - Conventional tools (LaTex, MathML, MathType,...) ⇒ High input time and know-how required

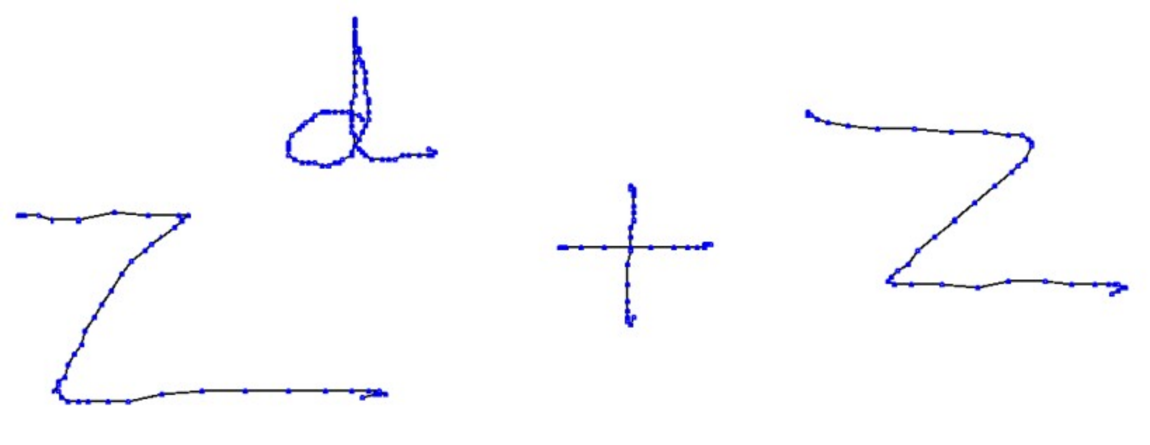


Math Recognition : On/Off line

- Off-line = image = 2D set of pixels
  - Scanned document, photo, …



- On-line = time sequence of (x,y) = strokes



Math Recognition

Three main sub-tasks in ME recognition

Symbol Segmentation

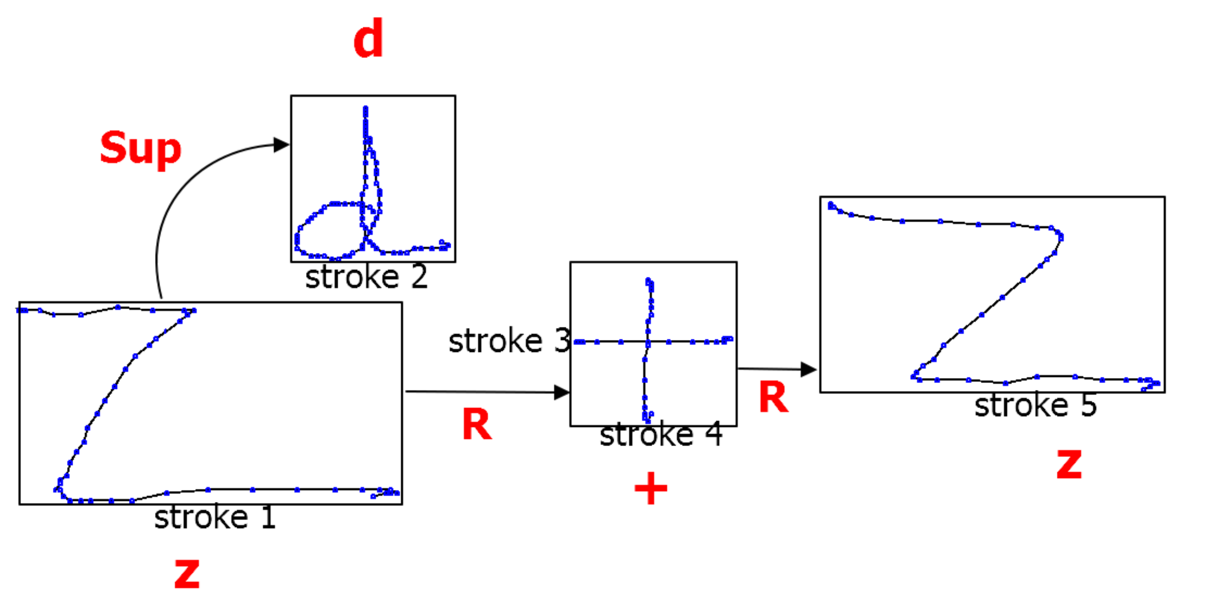
group strokes that belong to the same symbol.

Symbol Classification

assign each symbol candidate a class

Structural analysis

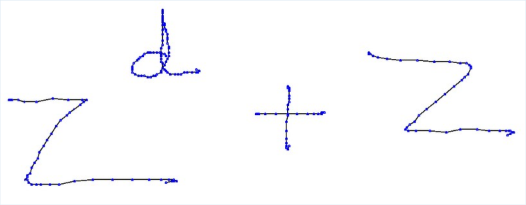
identify spatial relations between symbols and with the help of a 2-D language model to produce a math terpretation



- 5 strokes
- 4 symbols
  - z, d, + z
- 3 relations
  - Superscript
  - Right

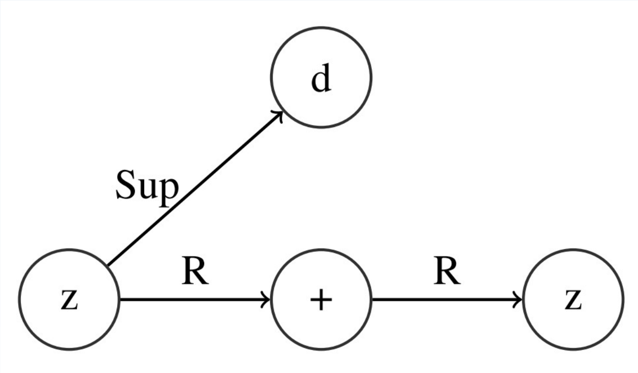
Math representation

- Graph representations at 2 levels



Symbol level

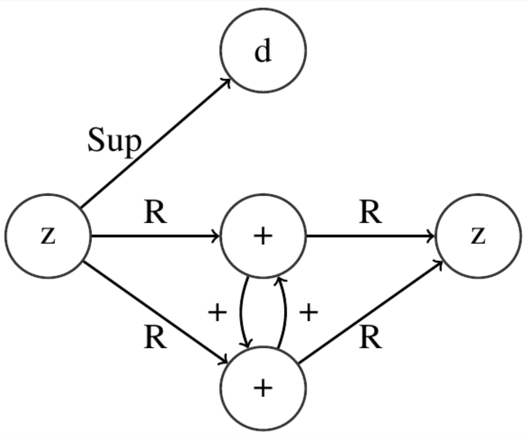
- Symbol relation (layout) tree



Node: symbol
Edge: symbols relationships

Stroke level

- Stroke label graph(SLG)



Node: stroke
Edge: stroke relationships

Math representation

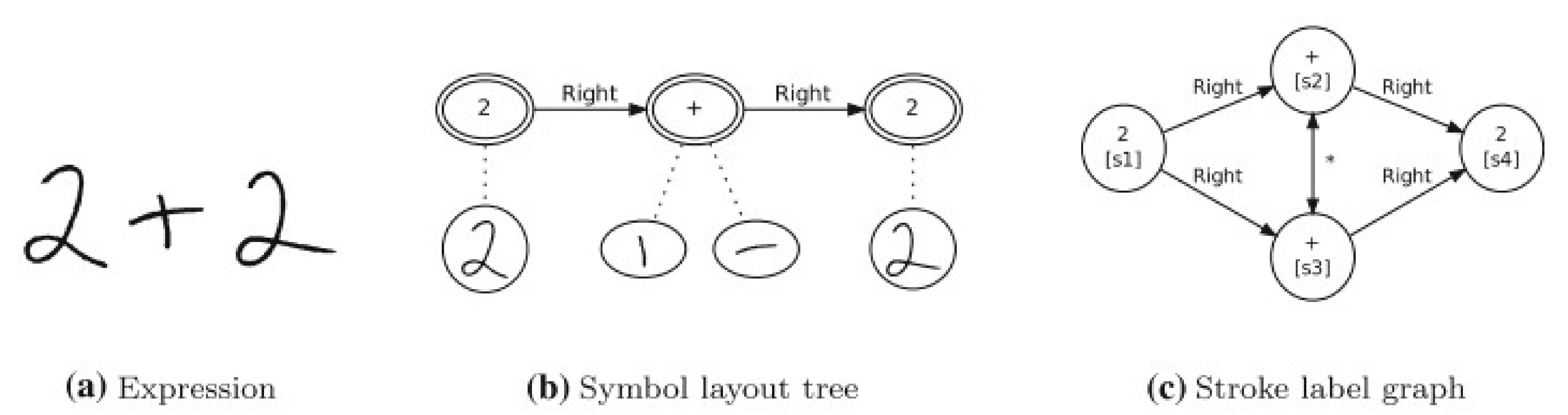
Graph representations at 2 levels

Symbol level

- Symbol relation (layout) tree

Stroke level

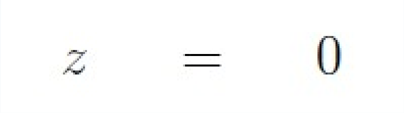
- Stroke label graph(SLG)



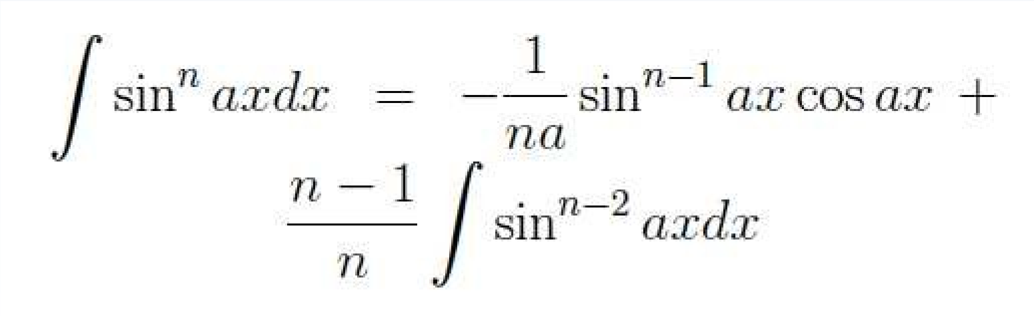
(a) Expression (b) Symbol layout tree (c) Stroke label graph

How to build a Math datasets

- Expression selection (corpus)
  - Realistic expressions
    - Different levels of difficulty
    - According to vocabulary (number of symbols)
    - According to grammar (number of rules)



$$z = 0$$



$$\int \sin^n ax dx = -\frac{1}{na} \sin^{n-1} ax \cos ax + \frac{n-1}{n} \int \sin^{n-2} ax dx$$

How to build a Math datasets

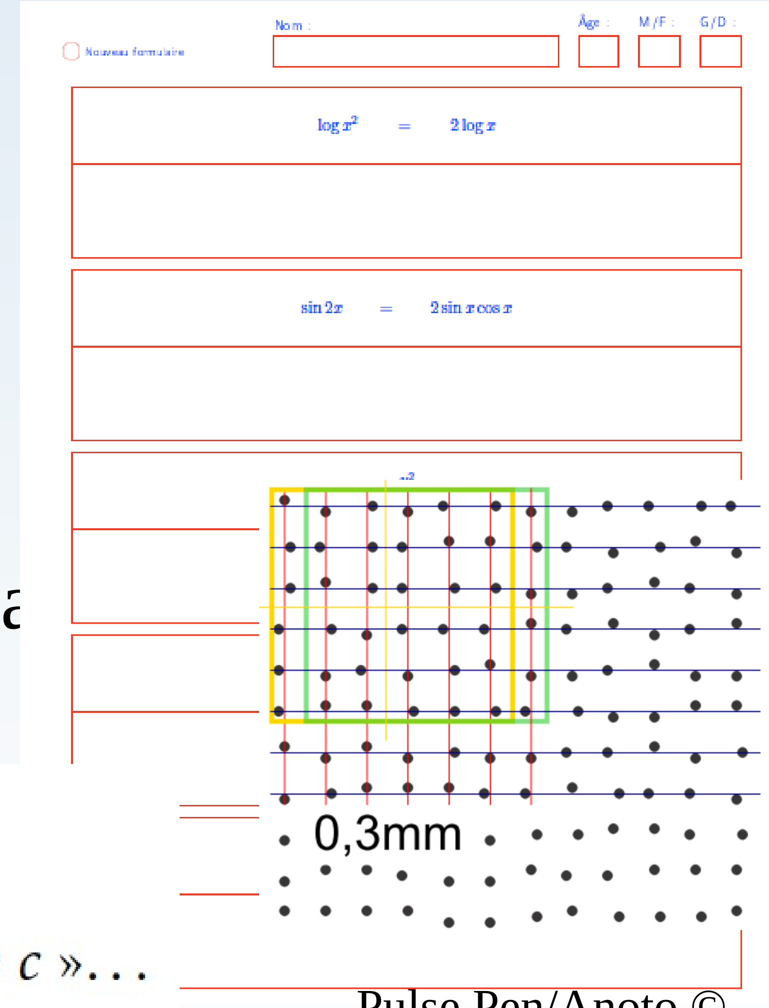
- How to obtain a large corpus ?
  - Write a grammar and generate
    - Not so realistic, but ok in specific domain
  - Collect from open sources
    - Wikipedia (<math>), ArXiv.org (tex sources $ $)
    - removing duplicate expressions
    - Filtering by string length
    - Filtering with a valid symbol set
    - Validating with a grammar parsing tool
    - Controlling the symbol frequency compared to an existing set

$$C_{c,r}(e) = \sum_{t_i \in e} \mathrm{cost}_{c,r}(t_i)$$

$$\mathrm{cost}_{c,r}(t_i) = \log(f_r(t_i)) - \log(f_c(t_i))$$

How to build a Math datasets

- Ink Data collection
  - Using e-paper and forms
  - Using GUI on tabletpc
- Speech data collection
  - dictation without constraints
  - Micro LEM et enregistreur Mara


○ Nouveau formulaire

Nom : Âge : M/F : G/D :

$\log x^2 \quad = \quad 2\log x$

$\sin 2x \quad = \quad 2\sin x\cos x$

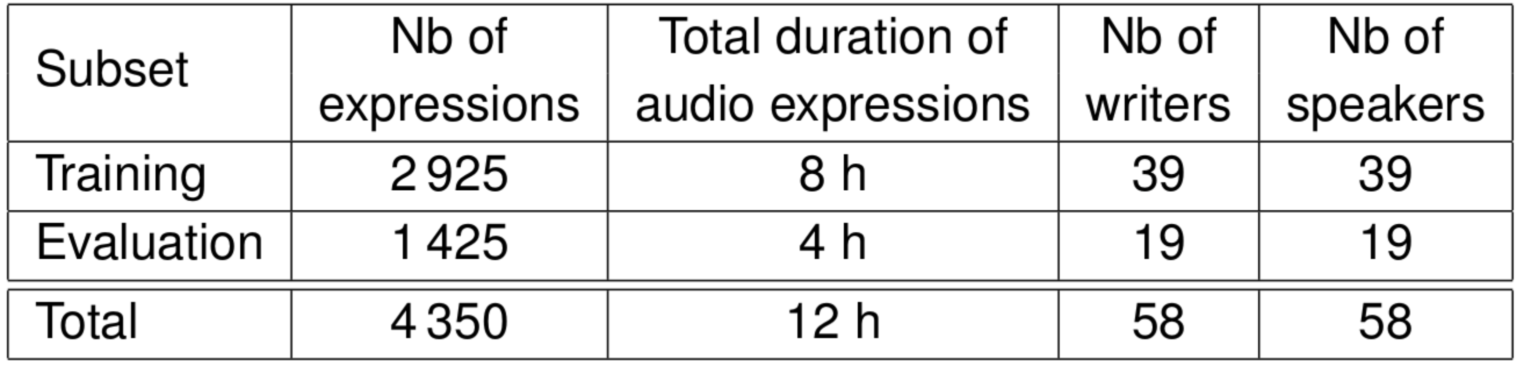
Pulse Pen/Anoto ©

$(a+b)\times c$ « *a plus b fois c* »

« *a plus b le tout multiplié par c* »

« *a plus b le tout entre parenthèses fois c* »...

HAMEX dataset



Subset	Nb of expressions	Total duration of audio expressions	Nb of writers	Nb of speakers
Training	2 925	8 h	39	39
Evaluation	1 425	4 h	19	19
Total	4 350	12 h	58	58

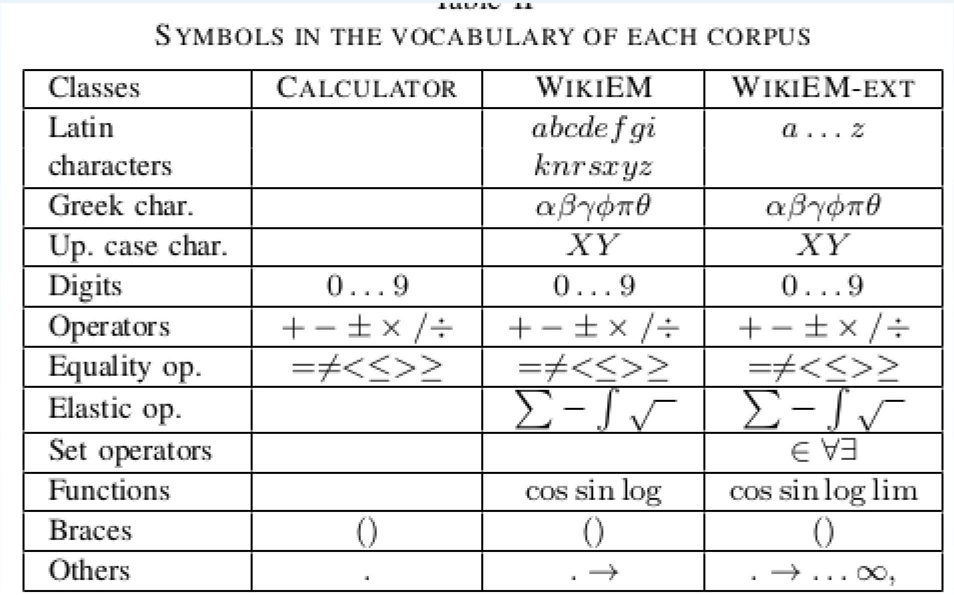


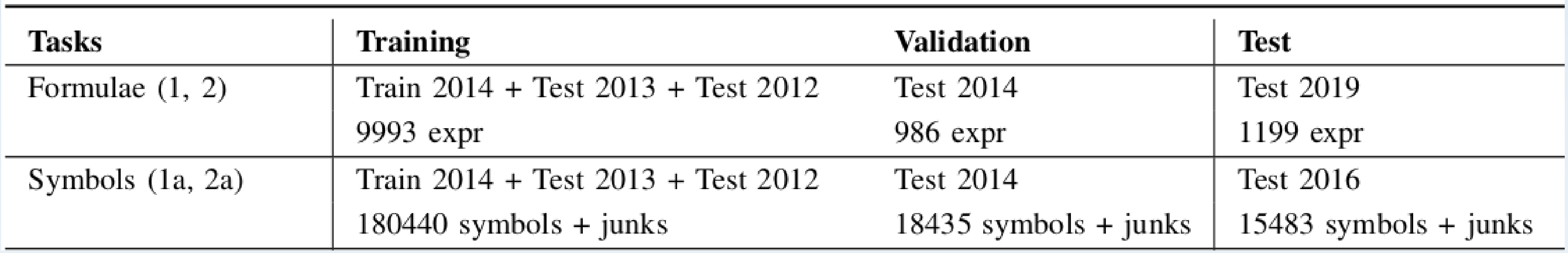
Table II

SYMBOLS IN THE VOCABULARY OF EACH CORPUS

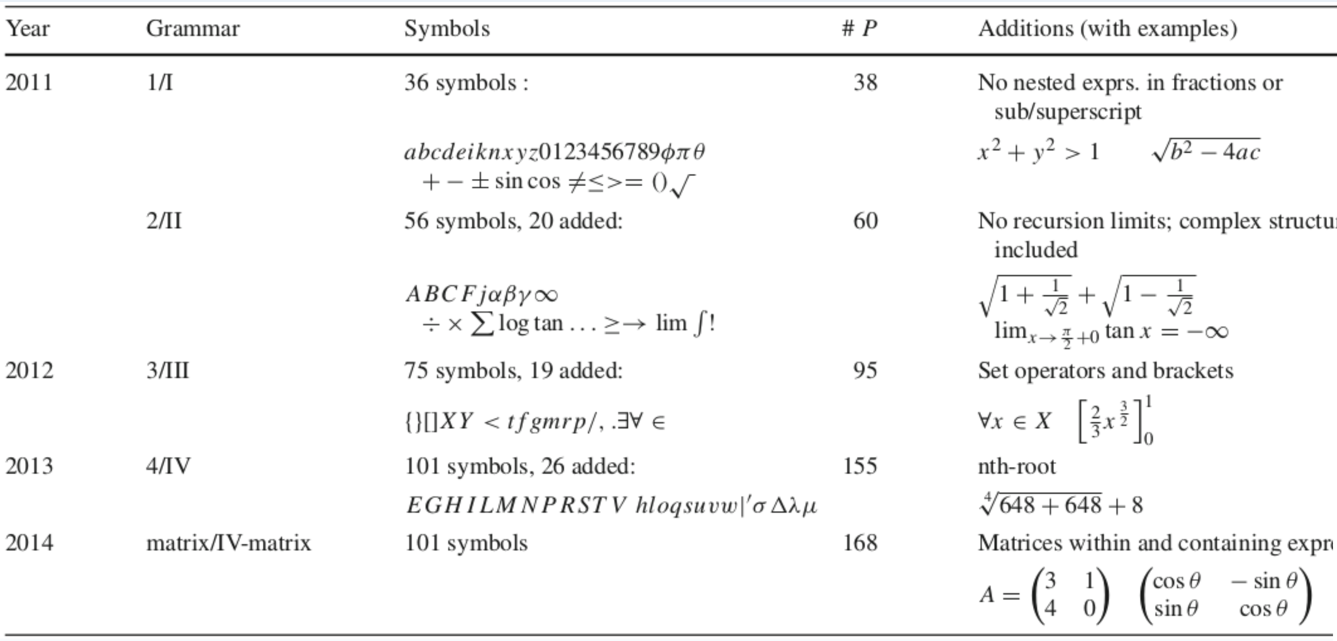
Classes	CALCULATOR	WIKIEM	WIKIEM-EXT
Latin characters		$abcdefgi$ $knrsxyz$	$a \ldots z$
Greek char.		$\alpha\beta\gamma\phi\pi\theta$	$\alpha\beta\gamma\phi\pi\theta$
Up. case char.		XY	XY
Digits	$0 \ldots 9$	$0 \ldots 9$	$0 \ldots 9$
Operators	$+ - \pm \times / \div$	$+ - \pm \times / \div$	$+ - \pm \times / \div$
Equality op.	$= \neq < \leq > \geq$	$= \neq < \leq > \geq$	$= \neq < \leq > \geq$
Elastic op.		$\sum - \int \sqrt{}$	$\sum - \int \sqrt{}$
Set operators			$\in \forall \exists$
Functions		cos sin log	cos sin log lim
Braces	()	()	()
Others	.	. →	. → . . . ∞,

CROHME dataset

Data size in 2019



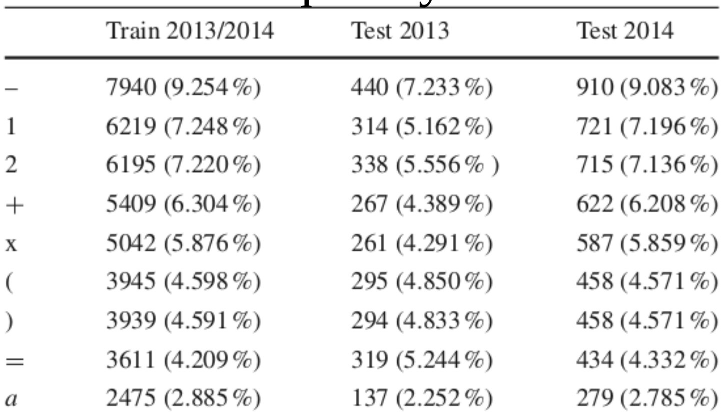
Tasks	Training	Validation	Test
Formulae (1, 2)	Train 2014 + Test 2013 + Test 2012 9993 expr	Test 2014 986 expr	Test 2019 1199 expr
Symbols (1a, 2a)	Train 2014 + Test 2013 + Test 2012 180440 symbols + junks	Test 2014 18435 symbols + junks	Test 2016 15483 symbols + junks



Year	Grammar	Symbols	# P	Additions (with examples)
2011	1/I	36 symbols :	38	No nested exprs. in fractions or sub/superscript
		$abcdeiknxyz0123456789\phi\pi\theta$ $+ - \pm \sin\cos \neq\leq>= ()\sqrt{}$		$x^2+y^2>1$ $\quad\sqrt{b^2-4ac}$
	2/II	56 symbols, 20 added:	60	No recursion limits; complex structures included
		$ABCFj\alpha\beta\gamma\infty$ $\div\times\sum\log\tan\ldots\geq\rightarrow\lim\int!$		$\sqrt{1+\frac{1}{\sqrt{2}}}+\sqrt{1-\frac{1}{\sqrt{2}}}$ $\lim_{x\rightarrow\frac{\pi}{2}+0}\tan x=-\infty$
2012	3/III	75 symbols, 19 added:	95	Set operators and brackets
		$\{\}[]XY<tfgmrp/,.\exists\forall\in$		$\forall x\in X\quad\left[\frac{2}{3}x^{\frac{3}{2}}\right]_0^1$
2013	4/IV	101 symbols, 26 added:	155	nth-root
		$EGHILMNPRSTV\ hloqsuvw|'\sigma\Delta\lambda\mu$		$\sqrt[4]{648+648}+8$
2014	matrix/IV-matrix	101 symbols	168	Matrices within and containing expr
				$A=\begin{pmatrix}3&1\\4&0\end{pmatrix}\quad\begin{pmatrix}\cos\theta&-\sin\theta\\\sin\theta&\cos\theta\end{pmatrix}$

Symbols, grammars, rules

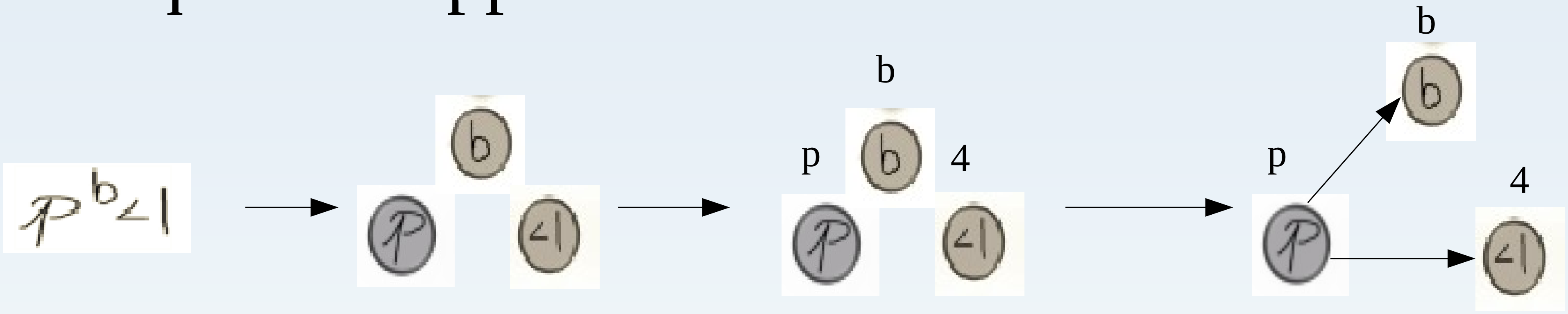
Freq. of symbols



	Train 2013/2014	Test 2013	Test 2014
–	7940 (9.254 %)	440 (7.233 %)	910 (9.083 %)
1	6219 (7.248 %)	314 (5.162 %)	721 (7.196 %)
2	6195 (7.220 %)	338 (5.556 %)	715 (7.136 %)
+	5409 (6.304 %)	267 (4.389 %)	622 (6.208 %)
x	5042 (5.876 %)	261 (4.291 %)	587 (5.859 %)
(	3945 (4.598 %)	295 (4.850 %)	458 (4.571 %)
)	3939 (4.591 %)	294 (4.833 %)	458 (4.571 %)
=	3611 (4.209 %)	319 (5.244 %)	434 (4.332 %)
a	2475 (2.885 %)	137 (2.252 %)	279 (2.785 %)

Math recognition : seq. solution

- Sequential approach

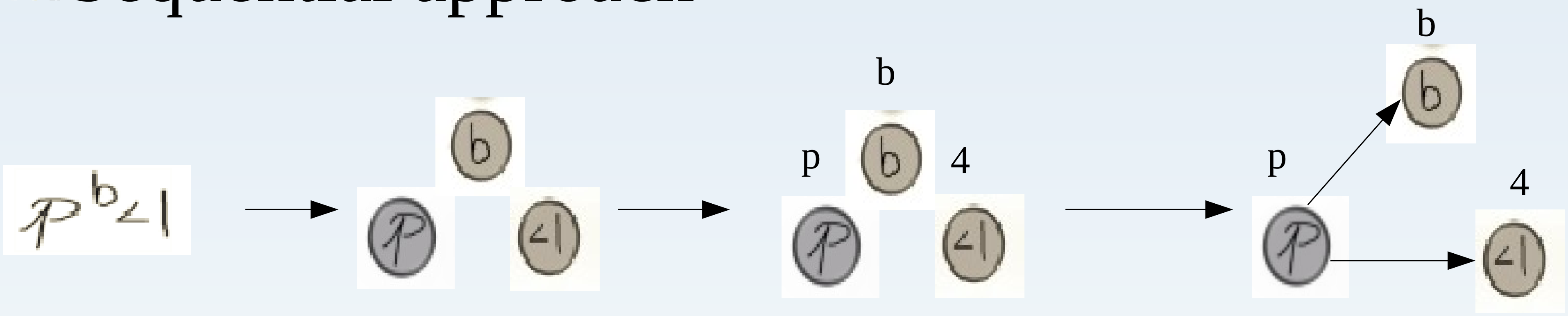


Seq of strokes → Segmentation → Classification → Structural analysis

- 3 steps = 3 classification tasks ?
  - Segmentation: symbol or junk ?
  - Classification: 101 classes
  - Relation recognition: 6 relations

Math recognition : seq. solution

- Sequential approach



Seq of strokes

Segmentation

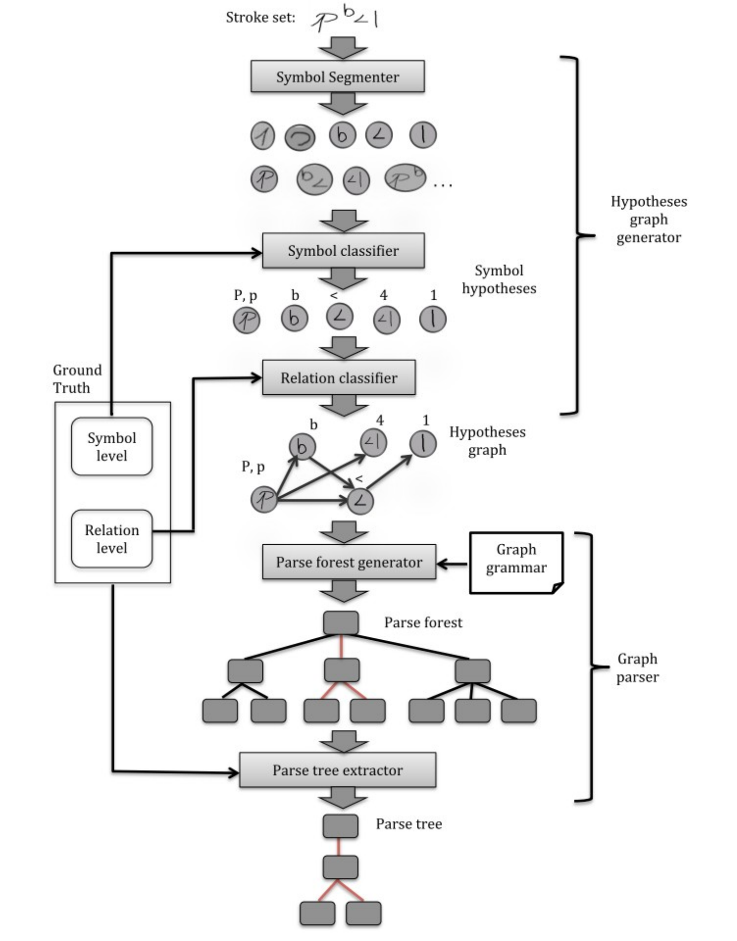
Classification

Structural analysis

- Error propagation

$$p^b < 1 \longrightarrow p^b 4$$

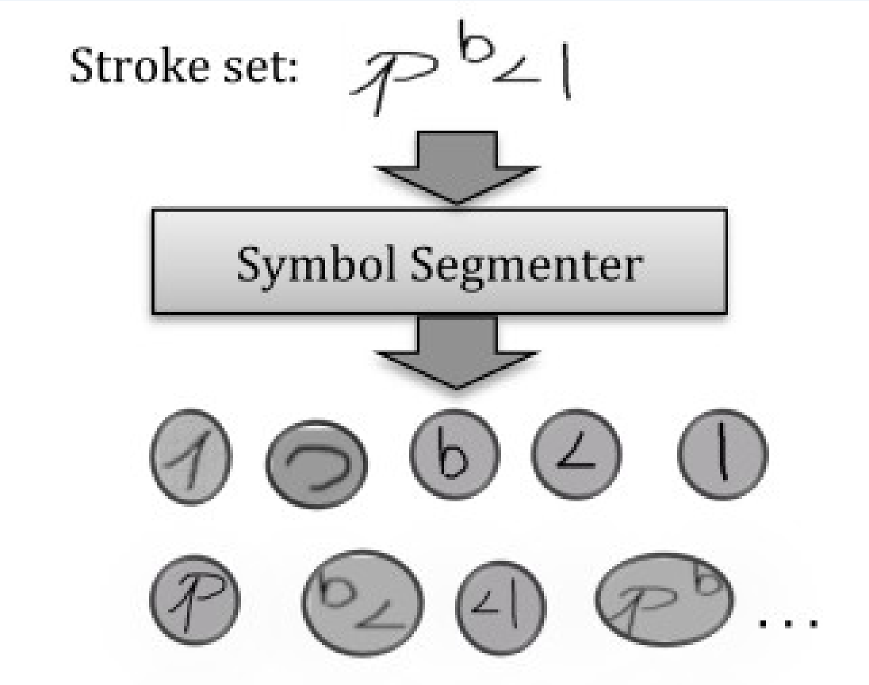
Math Reco: Integrated solution



- Do not take any final decision at each step
  - Hypothesis graph
  - Selection of the best solution
  - Cost function to minimize

Math Reco: Integrated solution

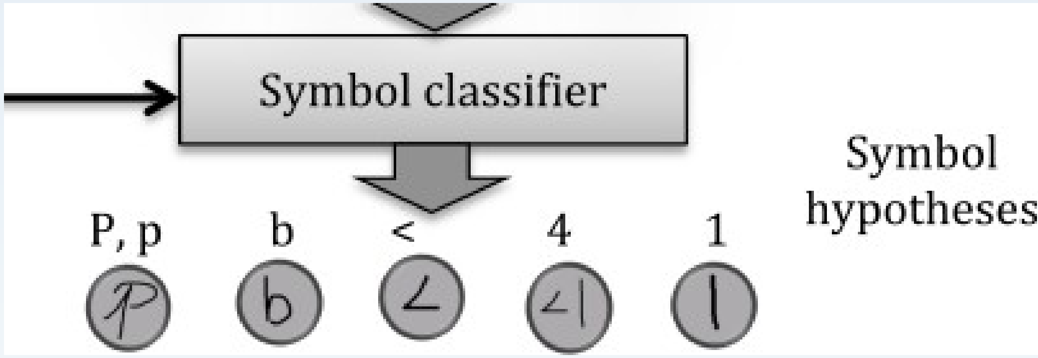
- Symbol segmenter
  - Use local information
  - Time / space continuity
  - 2 class classifier trained with wrong segmentation (junk)
  - Huge number of possible segmentation
    - Bell number
    - #symb is smaller...



$$B_{n+1} = \sum_{k=0}^{n} \binom{n}{k} B_k; \text{ with } B_0 = B_1 = 1$$

Math Reco: Integrated solution

- Symbol classifier
  - Online, Offline, On+Off
  - Keep all meaningful hypothesis
    - Use classifier score
  - One or several classes per hypothesis
  - Recognition cost :

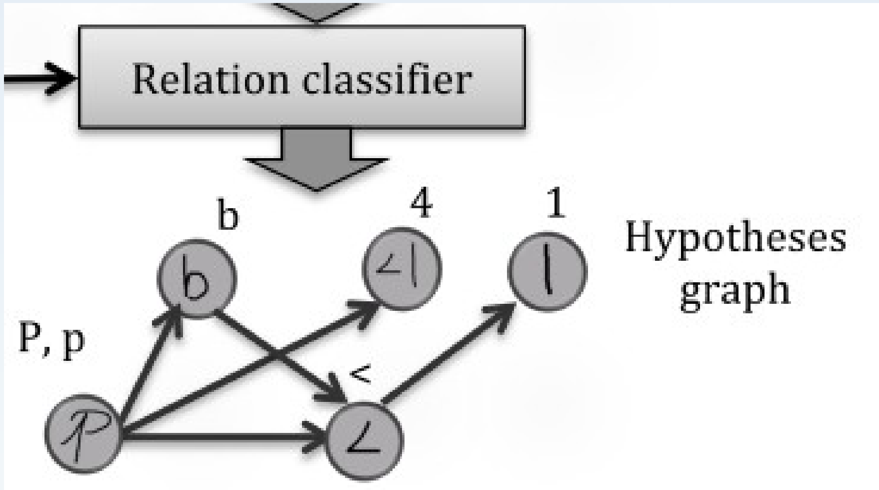


$$\sum_{j=1}^{topN} p(C_j|sh_i) \geqslant k$$

$$C_{reco}(sh_i) = -log(P(c = C_j|sh_i))$$

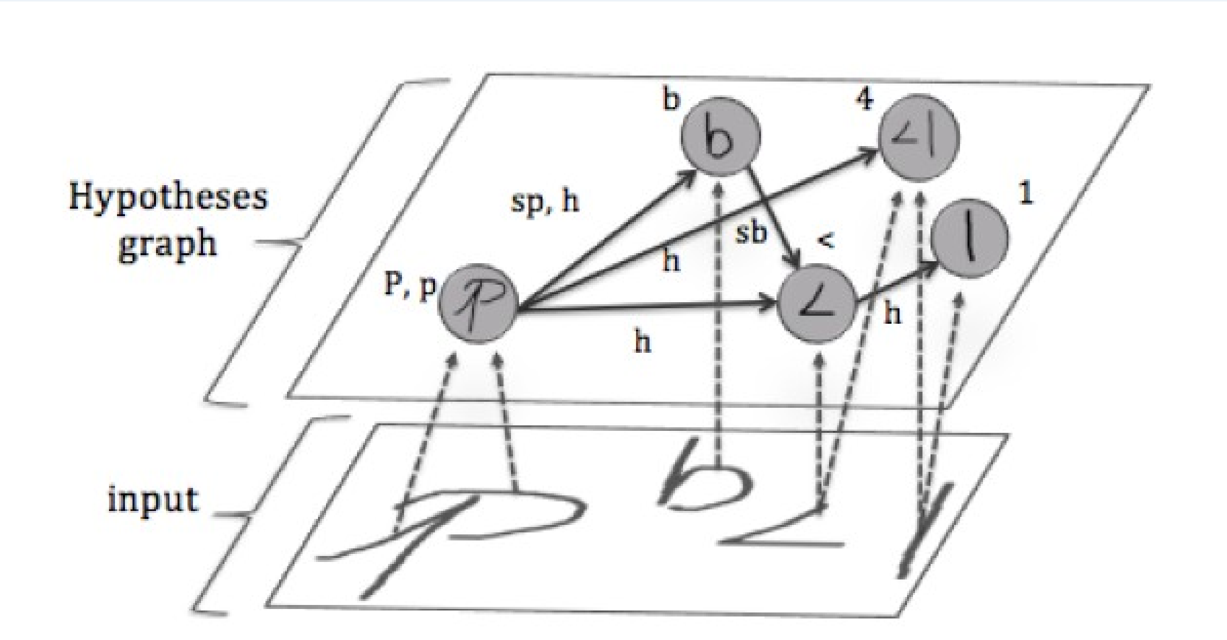
Math Reco: Integrated solution

- Relation classifier
  - Online, Offline, On+Off
  - Keep all meaningful relation hypothesis
    - Use classifier score
  - One or several classes per relation hypothesis
  - Recognition cost



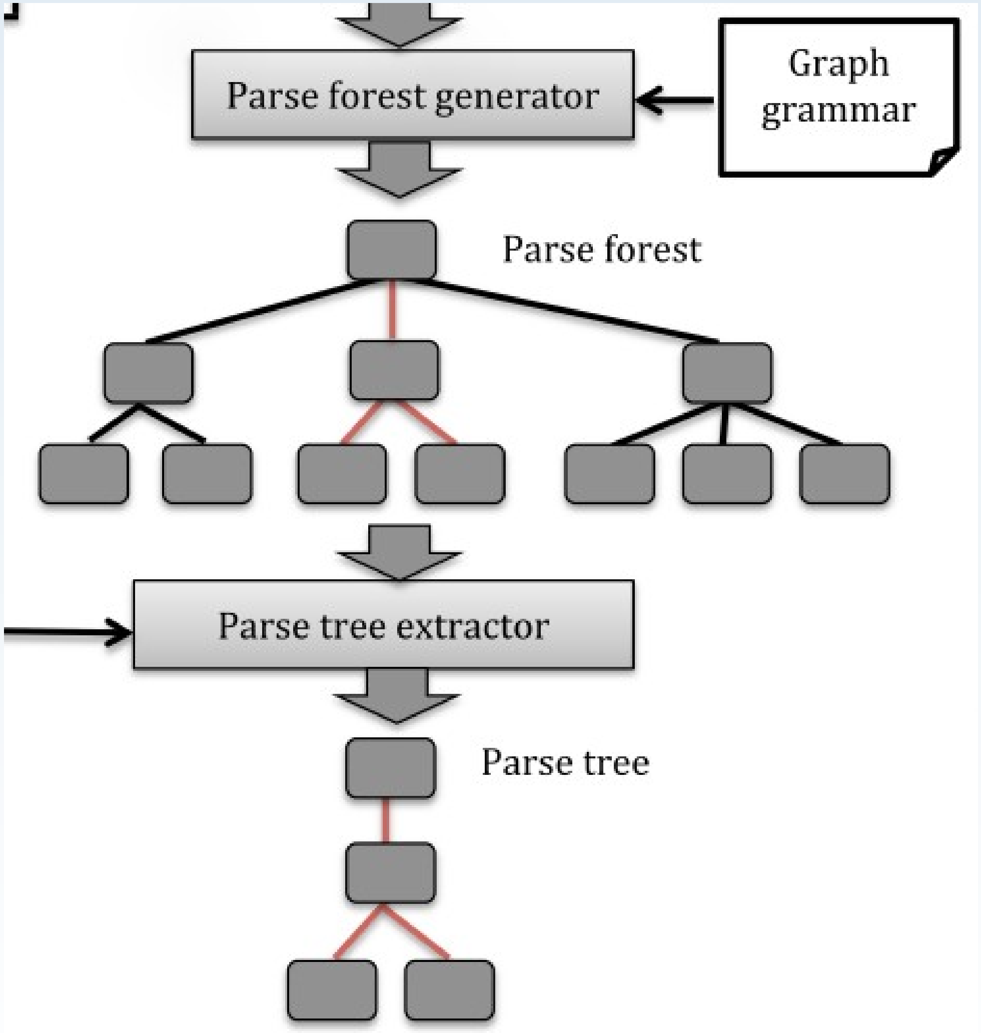
Math Reco: Integrated solution

- Graph of hypothesis
  - Keep all meaningful relation an symbol hypothesis
    - Can be quite complex !



Math Reco: Integrated solution

- Language model
  - Grammar
    - 2D Context Free Grammar
    - Graph grammar
  - Parse algo
    - CYK parsing
  - Select the best solution using a global cost
    - Seg + reco + rel + LM

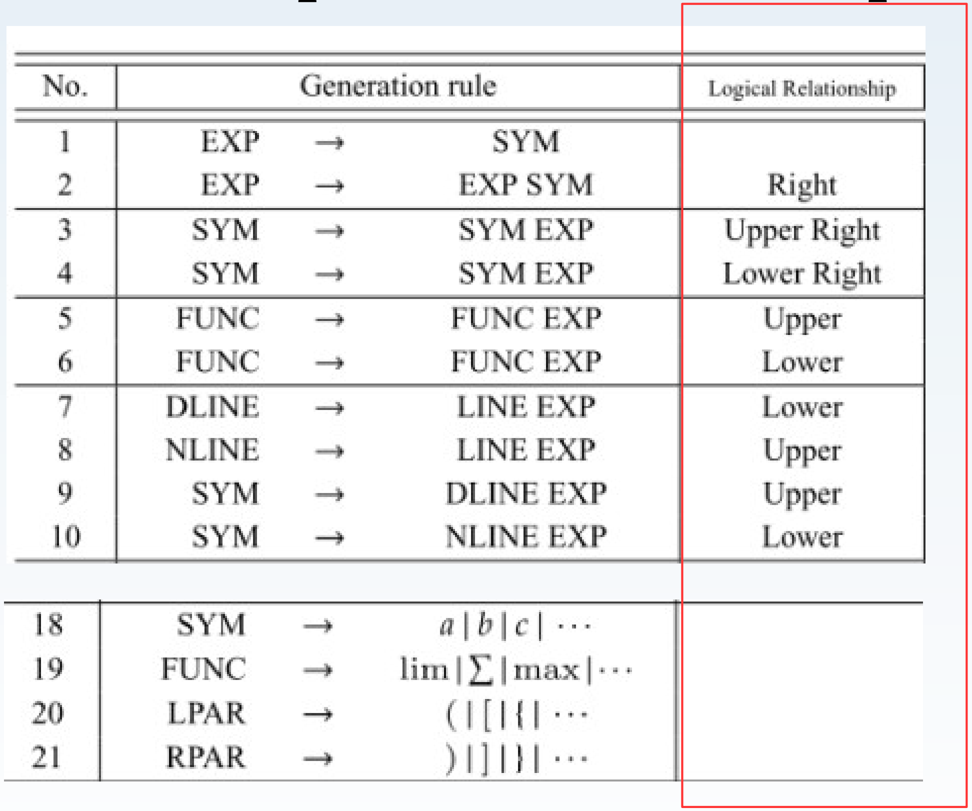


2D Context Free Grammar

- A grammar is a 4-tuple (N, T, I, R) such that:
  - N : finite set of symbols, called non-terminals;
  - T : finite set of symbols, called terminals;
  - $N \cap T = \emptyset$;
  - R : set of pairs (P, Q), production or rewriting rules, such that $P \in (N \cup T)^+$ and $Q \in (N \cup T)^*$;
  - I, $I \in N$, start or initial symbol.
- CNF (Chomsky Normal Form) : single non-terminal on their LHS

2D Context Free Grammar

- 2D CFG ?
  - Add a spatial relation ship in each rule



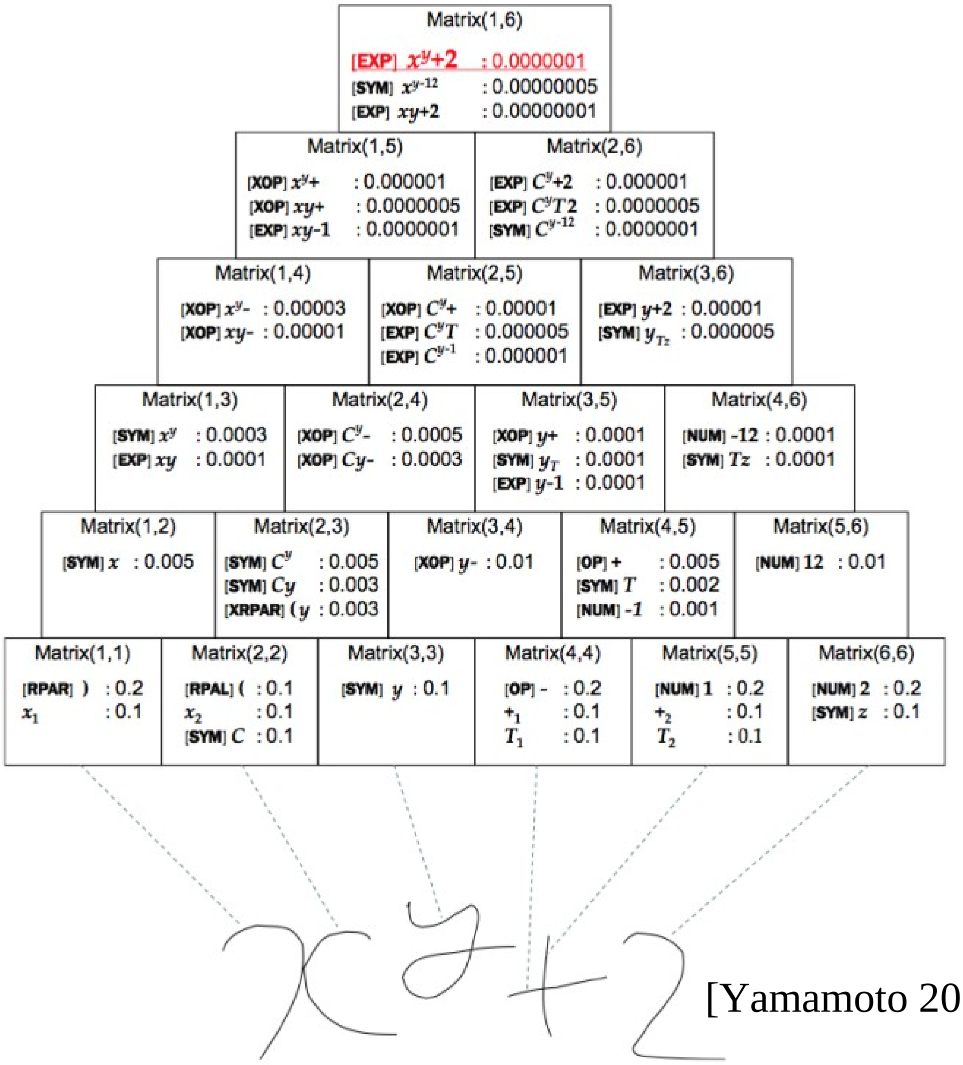
No.	Generation rule			Logical Relationship
1	EXP	→	SYM	
2	EXP	→	EXP SYM	Right
3	SYM	→	SYM EXP	Upper Right
4	SYM	→	SYM EXP	Lower Right
5	FUNC	→	FUNC EXP	Upper
6	FUNC	→	FUNC EXP	Lower
7	DLINE	→	LINE EXP	Lower
8	NLINE	→	LINE EXP	Upper
9	SYM	→	DLINE EXP	Upper
10	SYM	→	NLINE EXP	Lower
18	SYM	→	$a \mid b \mid c \mid \cdots$	
19	FUNC	→	$\lim \mid \sum \mid \max \mid \cdots$	
20	LPAR	→	$(\mid [\mid \{ \mid \cdots$	
21	RPAR	→	$) \mid] \mid \} \mid \cdots$	

… 28 rules ...

R. Yamamoto 2006, On-line recognition of handwritten mathematical expressions based on stroke-based stochastic context-free grammar. ICFHR

2D Context Free Grammar

- 2D CFG parsing : modified CYK
  - J. Cocke, D.H. Younger, and T. Kasami. (Younger 1967)
  - Add 2D complexity in the token combination



Matrix(1,6)
[EXP] x^y+2 : 0.0000001
[SYM] $x^{y\text{-}12}$: 0.00000005
[EXP] xy+2 : 0.00000001

Matrix(1,5)
[XOP] x^y+ : 0.000001
[XOP] xy+ : 0.0000005
[EXP] xy-1 : 0.0000001

Matrix(2,6)
[EXP] C^y+2 : 0.000001
[EXP] C^yT2 : 0.0000005
[SYM] $C^{y\text{-}12}$: 0.0000001

Matrix(1,4)
[XOP] x^y- : 0.00003
[XOP] xy- : 0.00001

Matrix(2,5)
[XOP] C^y+ : 0.00001
[EXP] C^yT : 0.000005
[EXP] $C^{y\text{-}1}$: 0.000001

Matrix(3,6)
[EXP] y+2 : 0.00001
[SYM] y_{T_z} : 0.000005

Matrix(1,3)
[SYM] x^y : 0.0003
[EXP] xy : 0.0001

Matrix(2,4)
[XOP] C^y- : 0.0005
[XOP] Cy- : 0.0003

Matrix(3,5)
[XOP] y+ : 0.0001
[SYM] y_T : 0.0001
[EXP] y-1 : 0.0001

Matrix(4,6)
[NUM] -12 : 0.0001
[SYM] Tz : 0.0001

Matrix(1,2)
[SYM] x : 0.005

Matrix(2,3)
[SYM] C^y : 0.005
[SYM] Cy : 0.003
[XRPAR] (y : 0.003

Matrix(3,4)
[XOP] y- : 0.01

Matrix(4,5)
[OP] + : 0.005
[SYM] T : 0.002
[NUM] -1 : 0.001

Matrix(5,6)
[NUM] 12 : 0.01

Matrix(1,1)
[RPAR]) : 0.2
x_1 : 0.1

Matrix(2,2)
[RPAL] (: 0.1
x_2 : 0.1
[SYM] C : 0.1

Matrix(3,3)
[SYM] y : 0.1

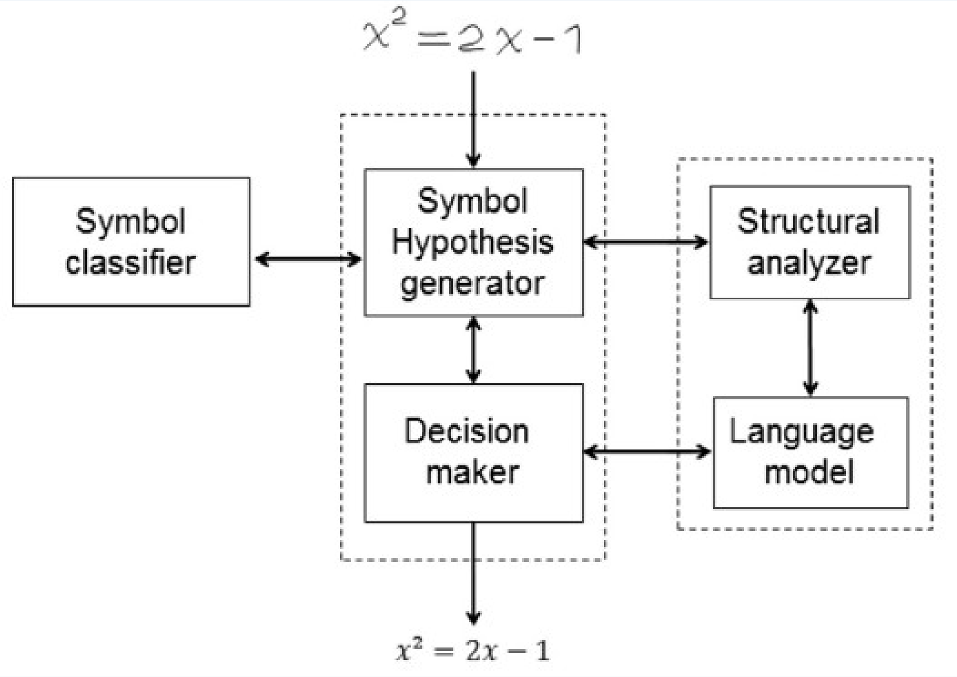
Matrix(4,4)
[OP] - : 0.2
$+_1$: 0.1
T_1 : 0.1

Matrix(5,5)
[NUM] 1 : 0.2
$+_2$: 0.1
T_2 : 0.1

Matrix(6,6)
[NUM] 2 : 0.2
[SYM] z : 0.1

[Yamamoto 2006]

Math Reco: Integrated solution



- Needed
  - Symbol classifier
  - Relation classifier
  - Grammar (handmade)
  - Parser which build a graph (Stroke LG)
- Very costly !

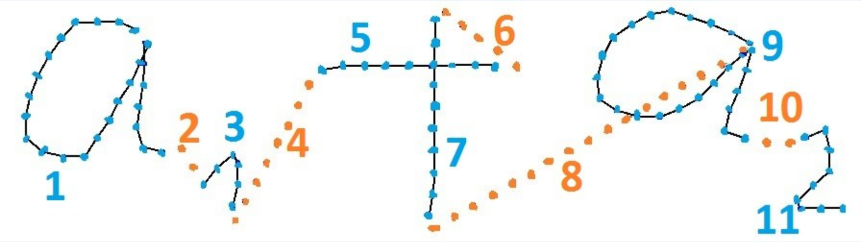
Math Reco: Merged solution

- Merge recognition steps
- Build directly a stroke label graph
- Use deep learning

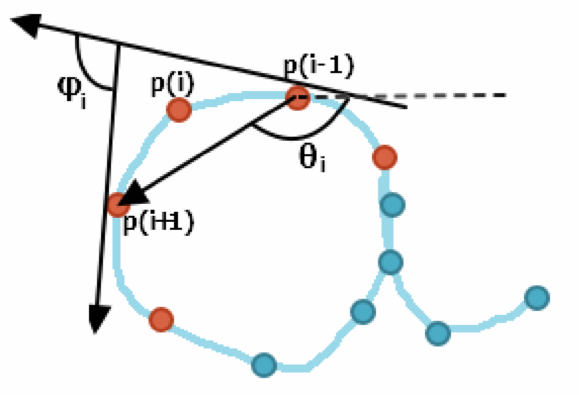
- PhD thesis of Ting Zhang (2017)
  - New Architectures for Handwritten Mathematical Expressions Recognition
  - Using Tree-BLSTM on on-line signal

Local on-line features

- Re-sample strokes



- Connect some pen-up strokes
- Extract 5 local features for each point

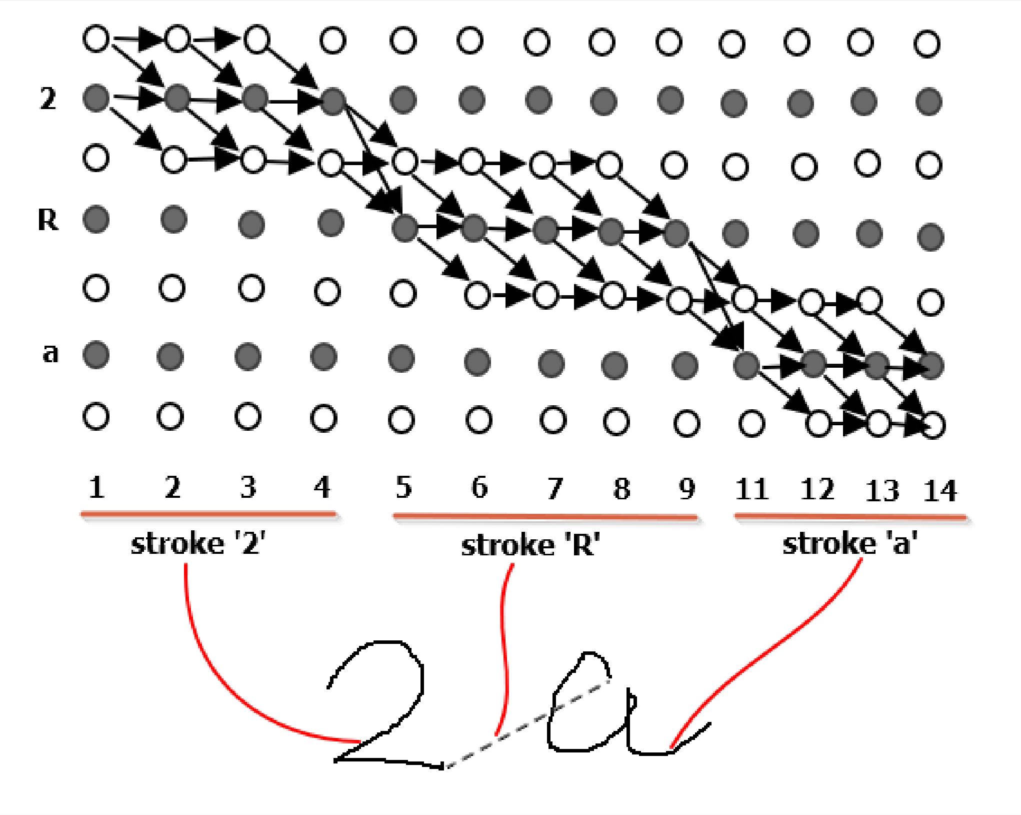


$$[\sin\theta_i, \cos\theta_i, \sin\phi_i, \cos\phi_i, PenUD_i]$$

BLSTM training / recognition

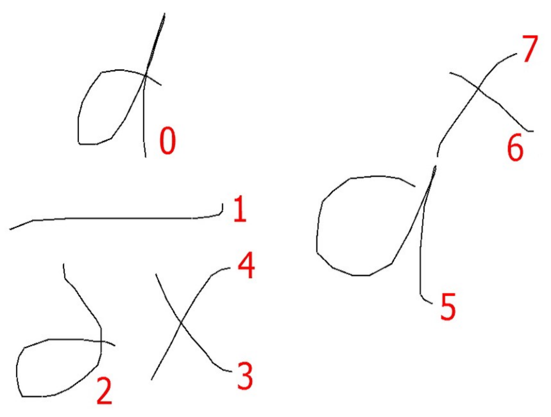
- Use local CTC to constraint local decision
  - 1 stroke = 1 label
    - Symbol label
    - Relation label
- Recognition with max rule
  - Class with the best score in the stroke

$$P_c^i = \sum_{t=1}^{|s_i|} p_{ct}^i$$

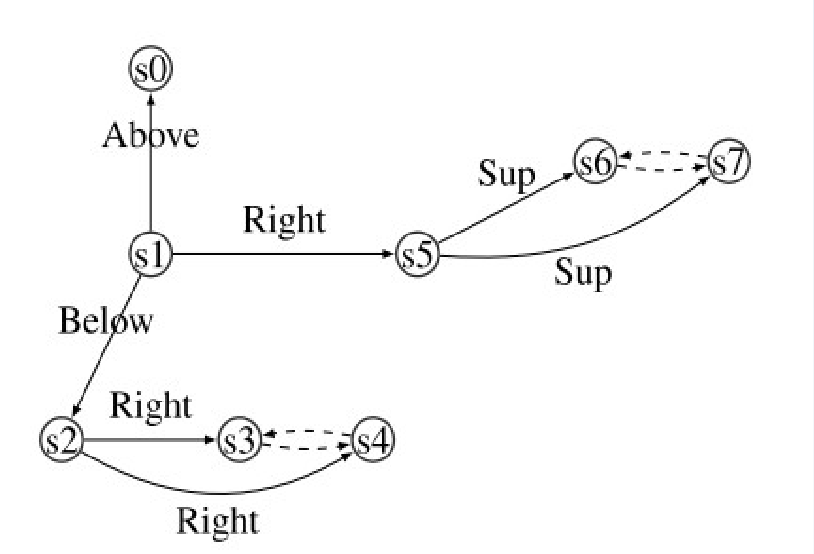


Graph of possible connections

- Math Expr are not linear structures: build the graph of all possible connections



(a) the input

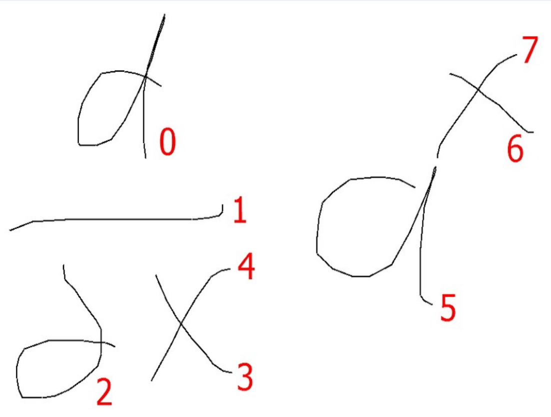


(b) the ground truth SLG

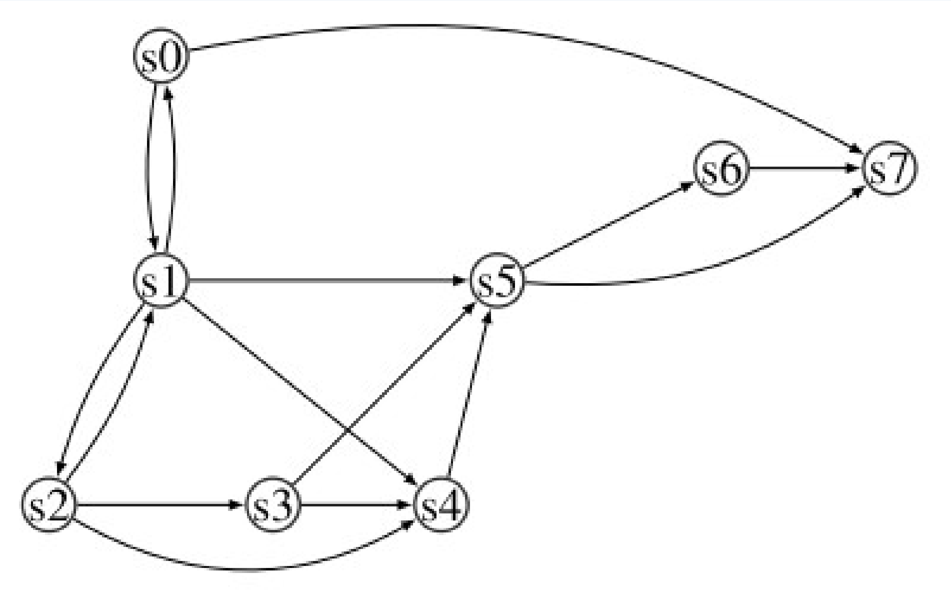
Graph of possible connections

- Math Expr are not linear structures: build the graph of all possible connections

•crossing, strokes
•visibility between strokes
•time order



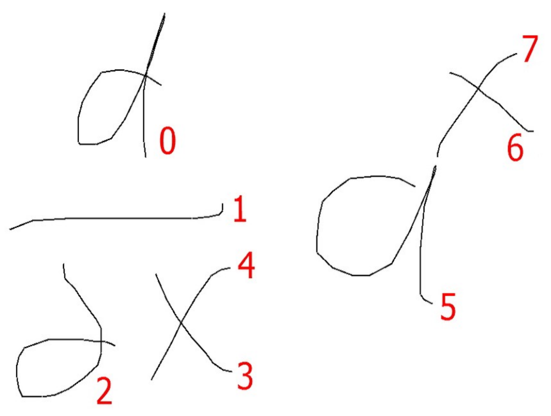
(a) the input



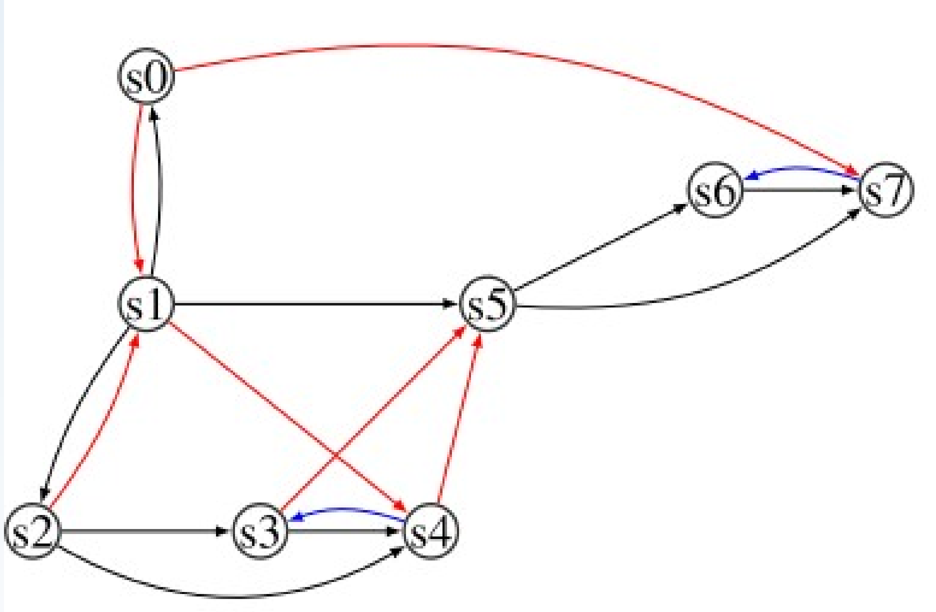
(c) the derived graph G

Graph of possible connections

- Math Expr are not linear structures: build the graph of all possible connections



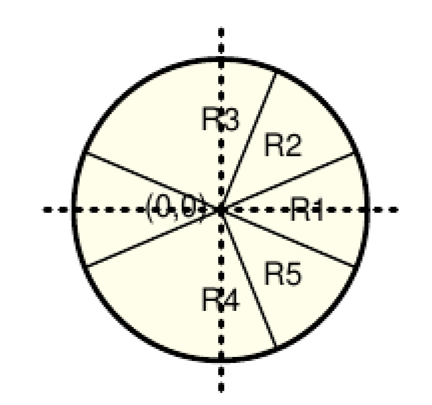
(a) the input

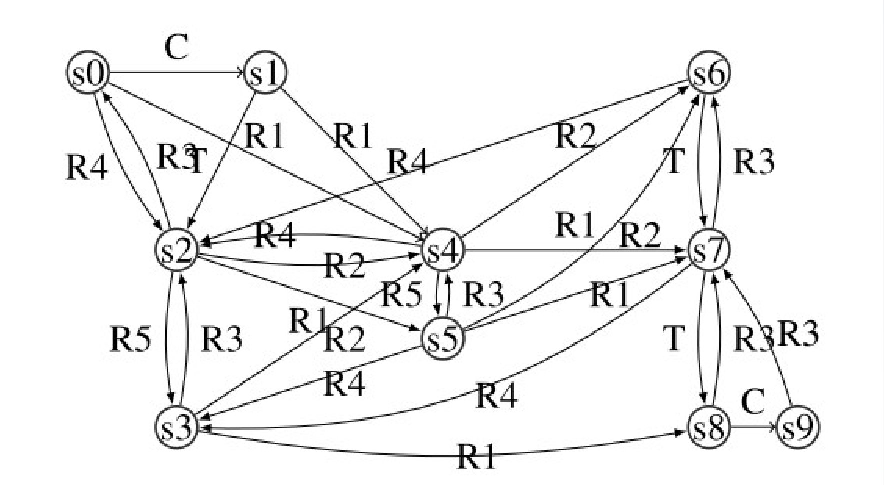


(d)Red: additional edges
Blue: missing edges

Graph of possible connections

- Math Expr are not linear structures: build the graph of all possible connections
  - Label the edges with directions





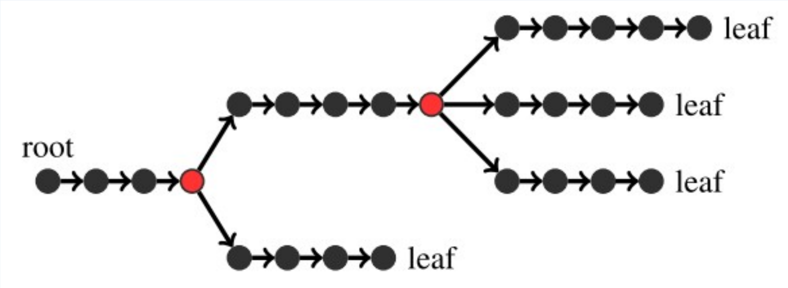
Tree based BLSTM

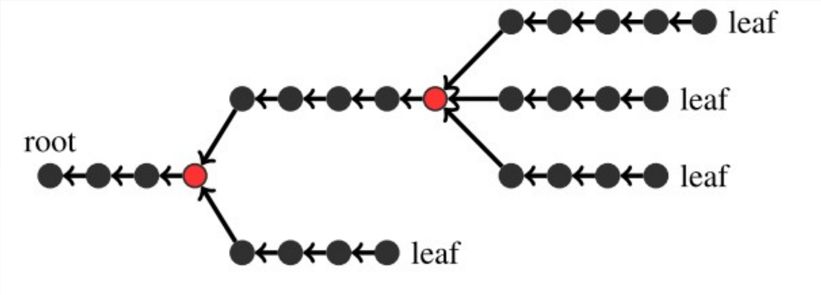
- Follow a tree structure
  - Both directions
  - Merge/split nodes with updated (back)propagation rules

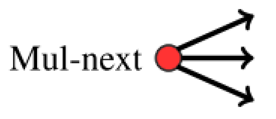
From root to leaves

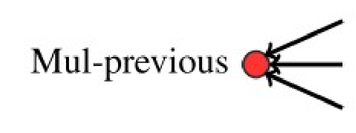
From leaves to root

Structure





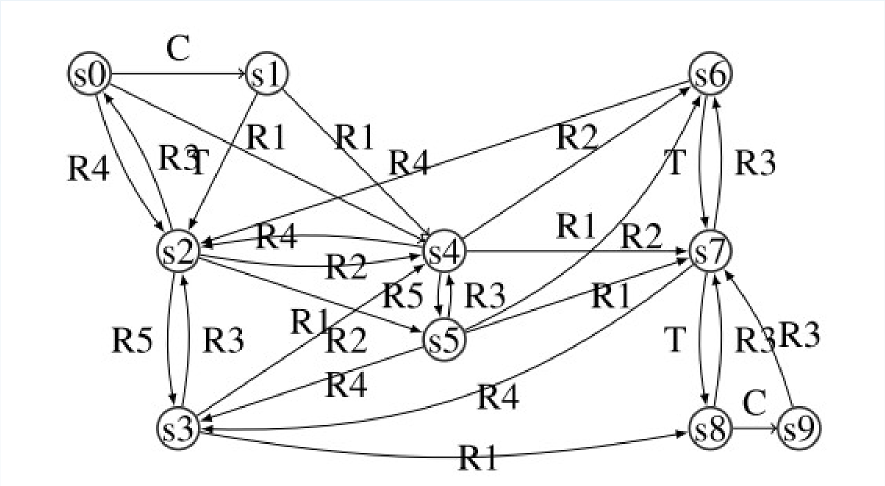


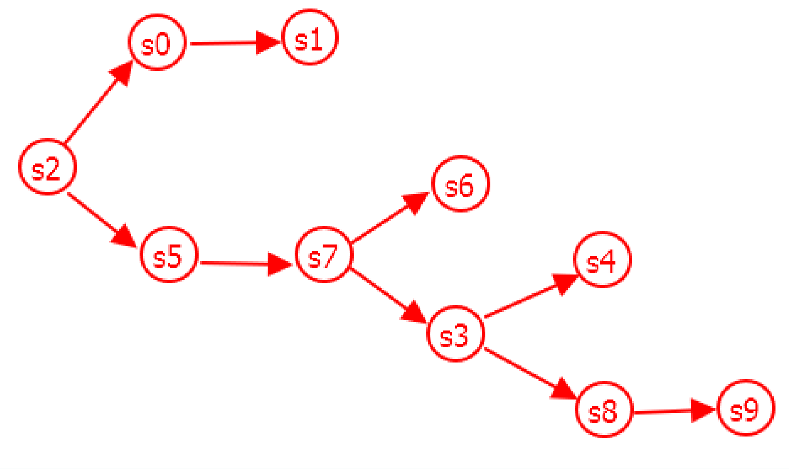


Forward propagation	duplicated to its next nodes	summed from the previous nodes
Error back propagation	summed from the next nodes	duplicated to its previous nodes

Tree in the graph

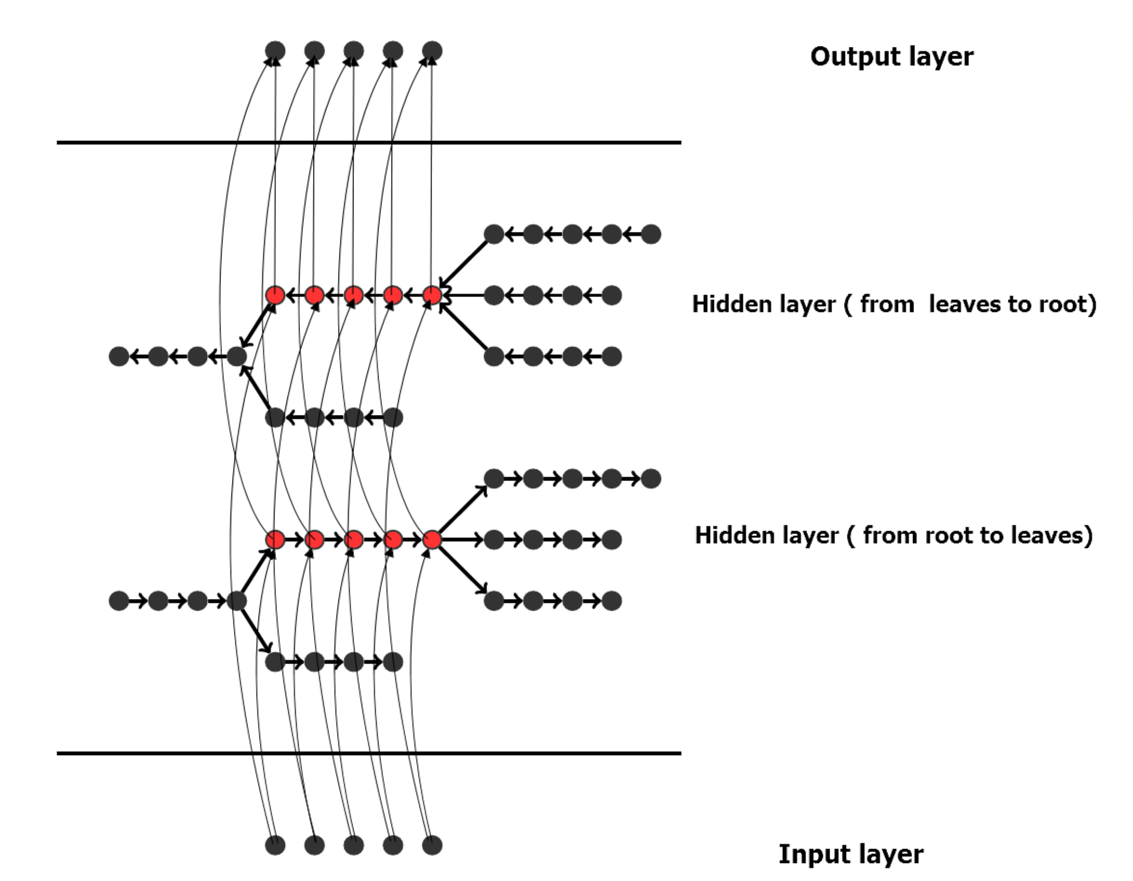
- Select a tree in the intermediate graph
  - Using direction criteria
    - Depth first with priority order (Crossing,R1, R3, R4, R2, R5, Time)





Tree labeling

- label one tree with tree-based BLSTM
  - Activation following the directions
  - Several staked BLSTM
  - Final decision locally
- Add a label "No Relation"


Output layer
Hidden layer (from leaves to root)
Hidden layer (from root to leaves)
Input layer

Graph labeling

- To label a graph
  - Generate several different trees
    - With different walking criteria
  - Label them all
  - Merge local decisions
    - Fusion rules (max, mean, weighted…)
  - Add some edges to complete the SLG
    - Segmentation edges
    - Relations in multi-stroke symbols

Results

- Network configuration
  - The hidden layers:
    - one level, Network (i)
    - two levels, Network (ii)
    - three levels, Network (iii)
    - four levels, Network (iv)
  - each level has the forward and backward layers
    - one layer contains 100 blocks
- Data set (CROHME 2014)
  - Train: 8834(10% for validation)
  - Test: 983

Results

- Performances increase with depth and merging more trees
- Recall increases but precision decreases...

Symbol level

Network, model	Symbol Segmentation Recall (%)	Symbol Classification Recall (%)	Relationship recognition Recall (%)
i, Merge 3	93.53	86.10	71.16
ii, Merge 3	95.01	88.38	76.20
iii, Merge 3	95.25	88.90	77.33
Merge 9	95.52	89.55	78.08

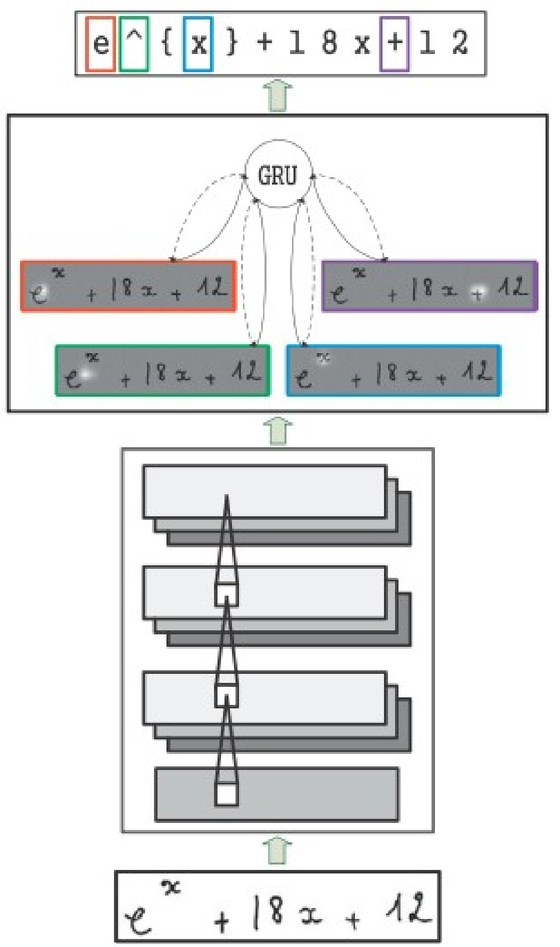
Expression level

Network, model	Correct (%)	<= 1 error (%)	<= 2 errors (%)
i, Merge 3	19.94	27.57	33.88
ii, Merge 3	25.94	36.72	42.32
iii, Merge 3	29.30	39.06	43.64
Merge 9	29.91	39.94	44.96

3 trees: Tree-Time, Tree-Left-R1 and Tree-0-R1

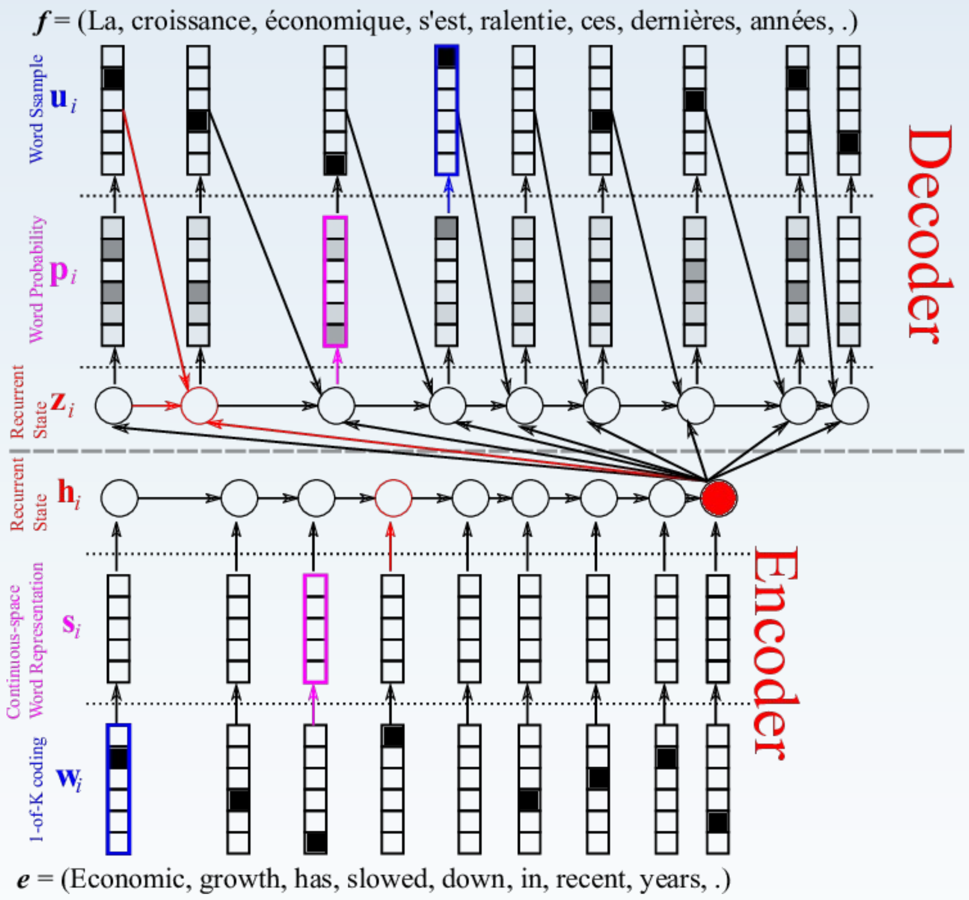
End-to-end approaches

- No intermediate structure
  - Feed the network with raw input
  - Estimate the cost directly using the output
- For the moment, no system are able to generate a graph
  - LaTeX string comparison



Watch, attend and parse: An end-to-end neural network based approach to handwritten mathematical expression recognition J. Zhang, PR 2017

Using RNN for sequence encoding

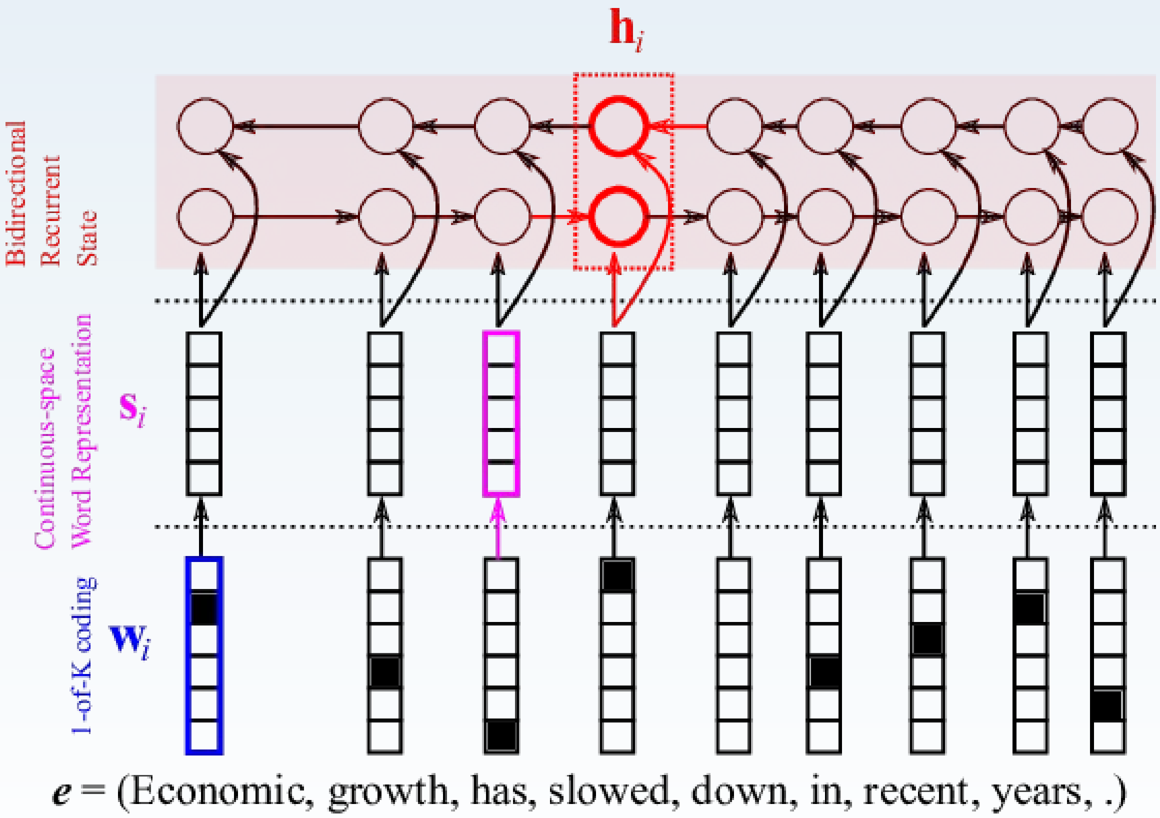

f = (La, croissance, économique, s'est, ralentie, ces, dernières, années, .)
Word Ssample
$\mathbf{u}_i$
Word Probability
$\mathbf{p}_i$
Recurrent State
$\mathbf{z}_i$
Decoder
Recurrent State
$\mathbf{h}_i$
Continuous-space Word Representation
$\mathbf{s}_i$
1-of-K coding
$\mathbf{w}_i$
Encoder
e = (Economic, growth, has, slowed, down, in, recent, years, .)

- One RNN encode
  - All info is merged in one final state
- One RNN decode
  - Generate a seq of output
  - Reuse the previous output for the next

[Loïc Barrault, Cours ATAL 2017]

Using RNN for sequence encoding

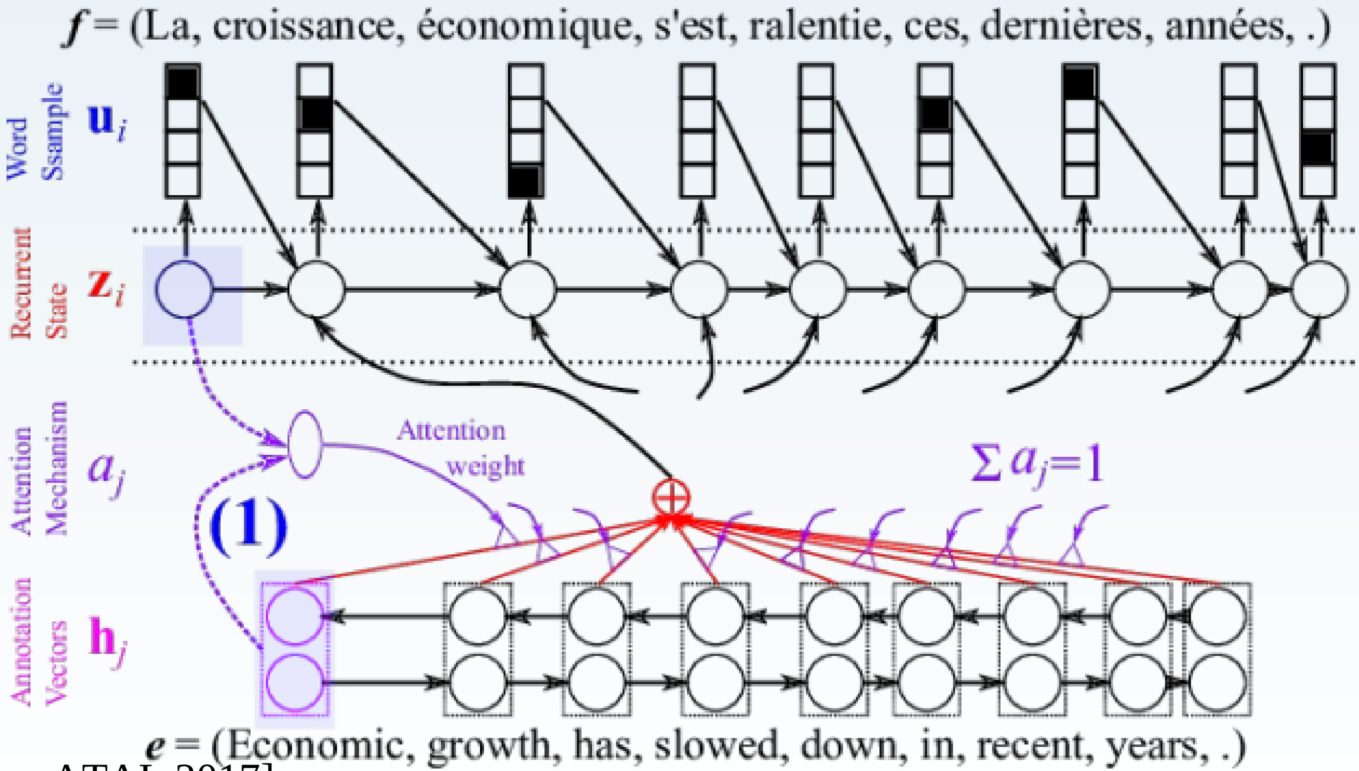
- Using Bi-RNN
  - All information available in each state


h_i
Bidirectional Recurrent State
Continuous-space Word Representation
s_i
1-of-K coding
w_i
e = (Economic, growth, has, slowed, down, in, recent, years, .)

[Loïc Barrault, Cours ATAL 2017]

Using RNN for sequence encoding

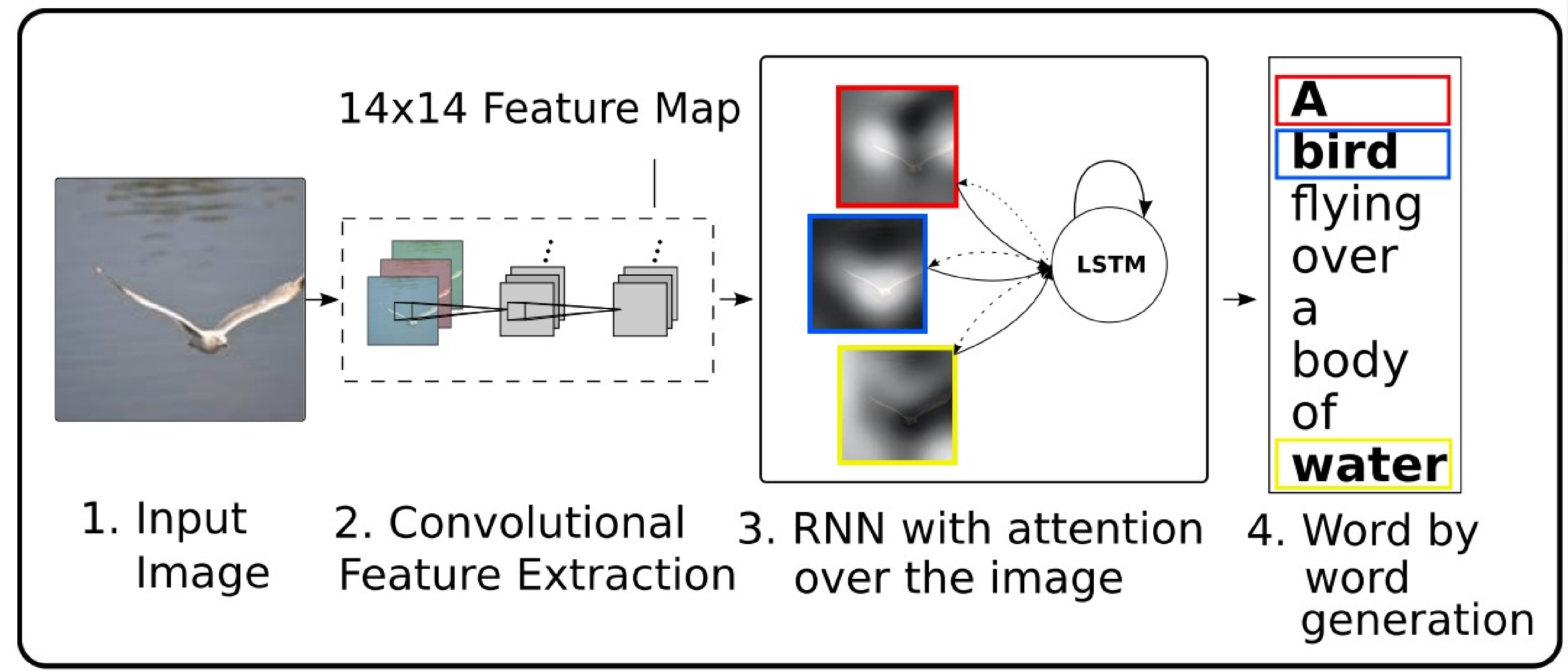
- Using Bi-RNN + Attention
  - Attention is focused on a sub part of information by weighted connection


f = (La, croissance, économique, s'est, ralentie, ces, dernières, années, .)
Word Ssample
u_i
Recurrent State
z_i
Attention Mechanism
a_j
Attention weight
Σ a_j=1
(1)
Annotation Vectors
h_j
e = (Economic, growth, has, slowed, down, in, recent, years, .)

[Loïc Barrault, Cours ATAL 2017]

Attention, images, text

- RNN LSTM to generate sentences
  - Show, Attend and Tell: Neural Image Caption Generation with visual Attention [K Xu 2015]


14x14 Feature Map
LSTM
A
bird
flying
over
a
body
of
water
1. Input Image
2. Convolutional Feature Extraction
3. RNN with attention over the image
4. Word by word generation

Show, Attend and Tell...

- Input : image
- Output : sequence of words
  - C words among K
  - One hot vector (K = 10 000)

$$y = \{\mathbf{y}_1, \ldots, \mathbf{y}_C\},\ \mathbf{y}_i \in \mathbb{R}^K$$

- Visual Encoder
  - Convolutional, keeping last conv layer
    - 14 x 14 map with 512 features
    - **L** vector of **D** features
  - VGG pretrained
    - (or any other)

$$a = \{\mathbf{a}_1, \ldots, \mathbf{a}_L\},\ \mathbf{a}_i \in \mathbb{R}^D$$

Show, Attend and Tell...

- Decoder : LSTM based
  - i,f,o,g :input, forget, memory, output and hidden state
  - $\hat{z} \in R^D$: context vector (features)
  - $E \in R^{m \times K}$: embedding matrix
  - T : all LSTM weights

Previous word embedded (dim m)

$$\begin{pmatrix} \mathbf{i}_t \\ \mathbf{f}_t \\ \mathbf{o}_t \\ \mathbf{g}_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} T_{D+m+n,n} \begin{pmatrix} \mathbf{E}\mathbf{y}_{t-1} \\ \mathbf{h}_{t-1} \\ \hat{\mathbf{z}}_t \end{pmatrix}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t).$$

Hidden state (dim n)
Context vector (dim D)

- Output
  - Deep output layer merging embedding, hidden and context

$$p(\mathbf{y}_t | \mathbf{a}, \mathbf{y}_1^{t-1}) \propto \exp(\mathbf{L}_o(\mathbf{E}\mathbf{y}_{t-1} + \mathbf{L}_h \mathbf{h}_t + \mathbf{L}_z \hat{\mathbf{z}}_t))$$

Show, Attend and Tell...

- Attention mechanism
  - α_{ti} = how much each location should be participate to the context vector $\hat{z}_t$

    $$\sum_i \alpha_{ti} = 1$$

  - Hard / Soft
    - Soft

$$\mathbb{E}_{p(s_t|a)}[\hat{\mathbf{z}}_t] = \sum_{i=1}^{L} \alpha_{t,i}\mathbf{a}_i$$

$$e_{ti} = f_{\text{att}}(\mathbf{a}_i, \mathbf{h}_{t-1})$$

$$\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{k=1}^{L} \exp(e_{tk})}.$$

$$\hat{\mathbf{z}}_t = \phi\left(\{\mathbf{a}_i\}, \{\alpha_i\}\right)$$

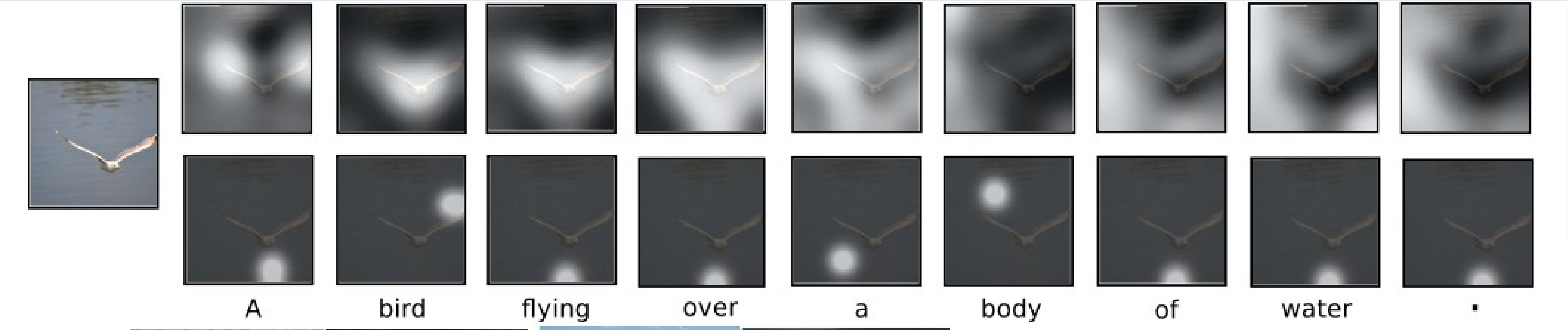
Show, Attend and Tell...

- Training

$$L_d = -\log(P(\mathbf{y}|\mathbf{x})) + \lambda \sum_{i}^{L} (1 - \sum_{t}^{C} \alpha_{ti})^2$$

  - Loss :
    - Minimizing the log likelihood
    - Regularization of alpha to 1 (try to use all locations)
  - CNN : VGG pretrained
  - stochastic gradient descent using adaptive learning rate (Adam)
  - mini-batch of size 64 of homogenous length of caption
  - early stopping on BLEU score
  - Microsoft COCO dataset (83.000)
  - Flickr8k/Flickr30k

Show, Attend and Tell...



A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.

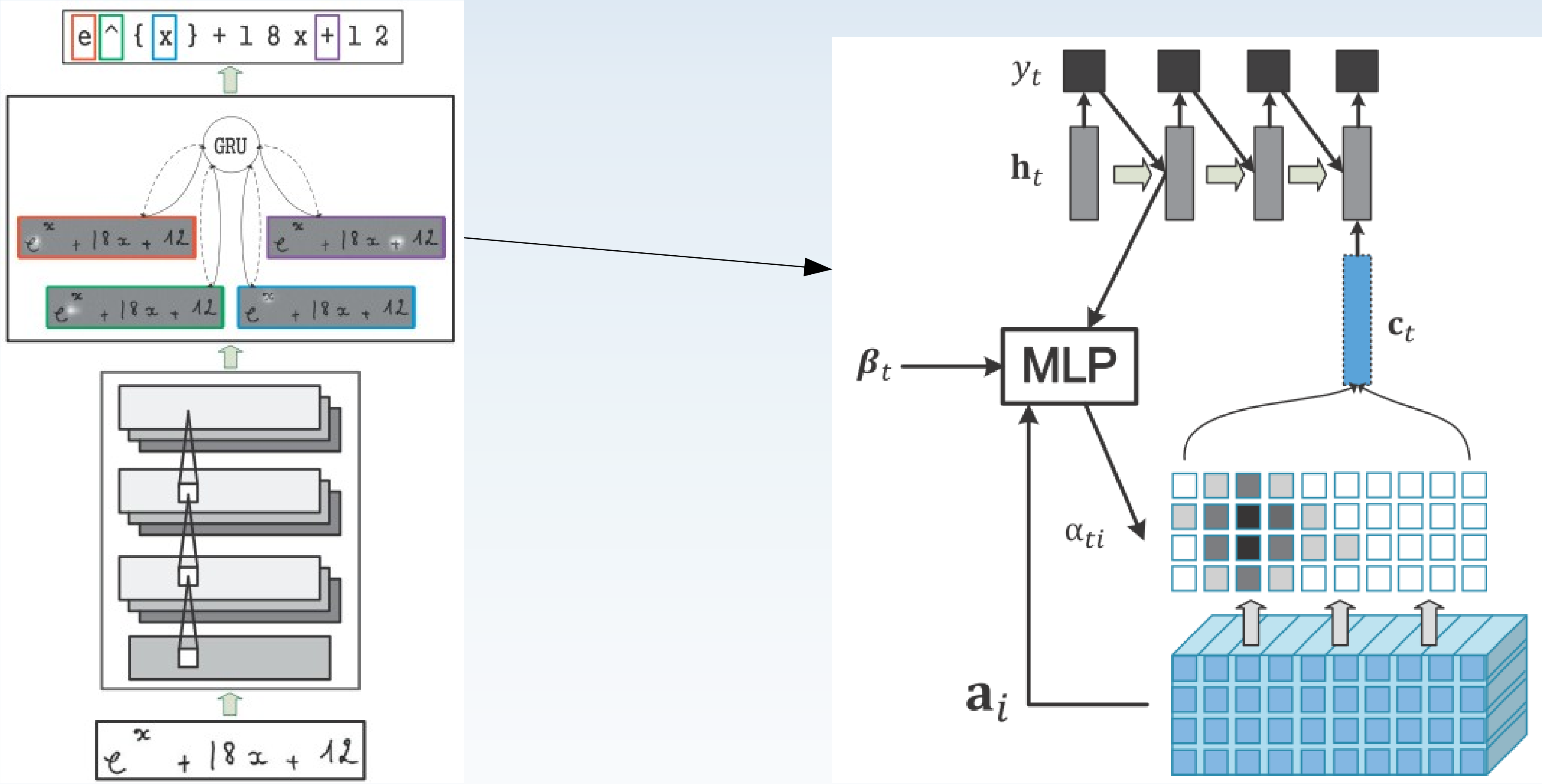
A man wearing a hat and a hat on a skateboard.

A group of people sitting on a boat in the water.

A giraffe standing in a forest with trees in the background.

A man is talking on his cell phone while another man watches.

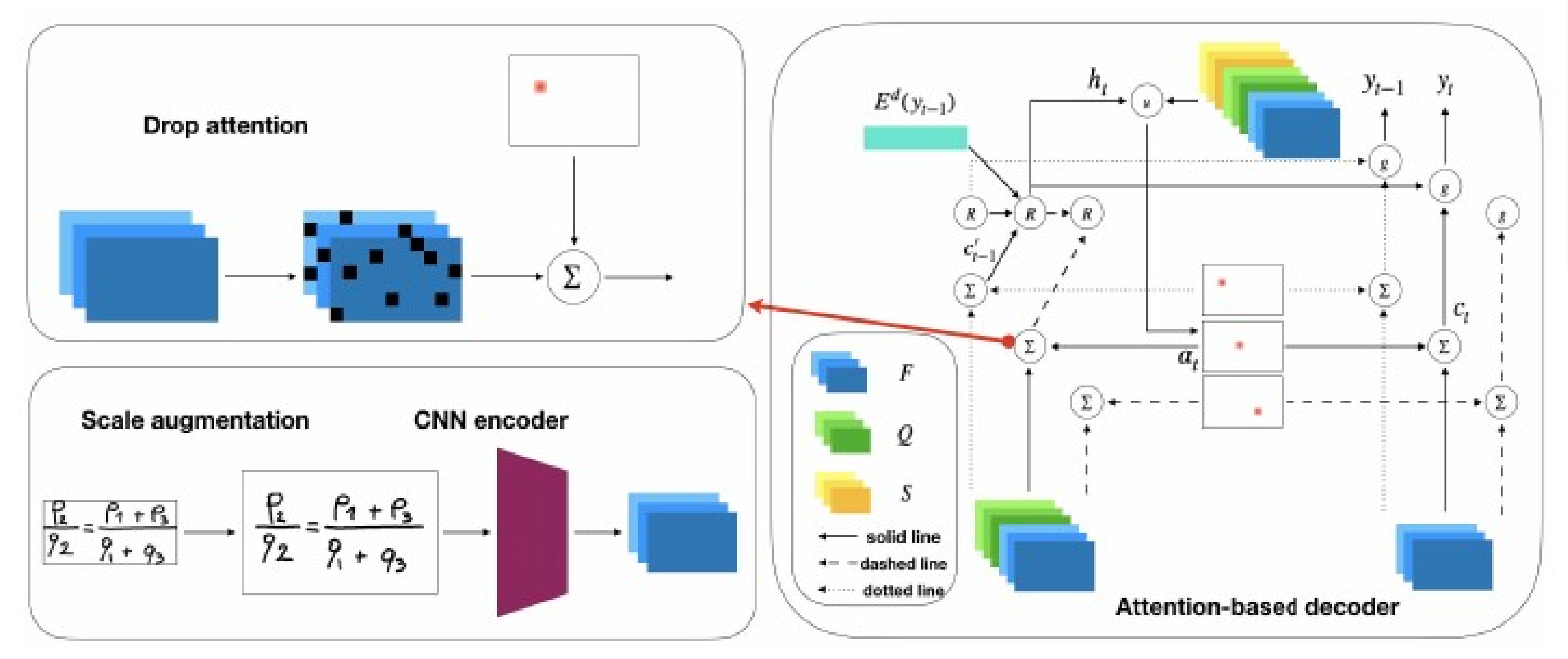
•Watch, attend and parse



Watch, attend and parse: An end-to-end neural network based approach to handwritten mathematical expression recognition J. Zhang, PR 2017

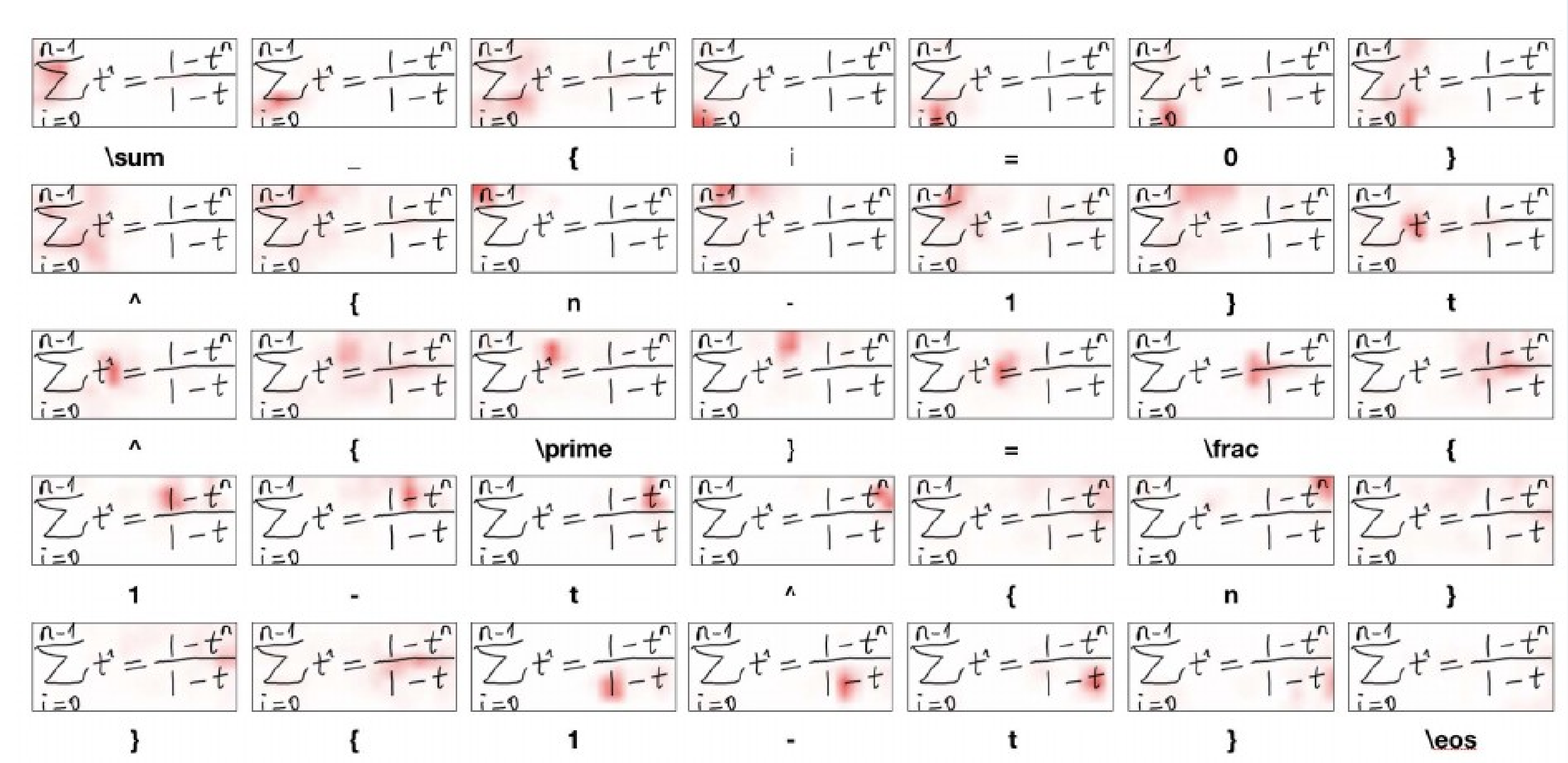
Attention + LSTM in Math Reco

- Improving Attention-Based Handwritten Mathematical Expression Recognition with Scale Augmentation and Drop Attention
  - ICFHR 2020



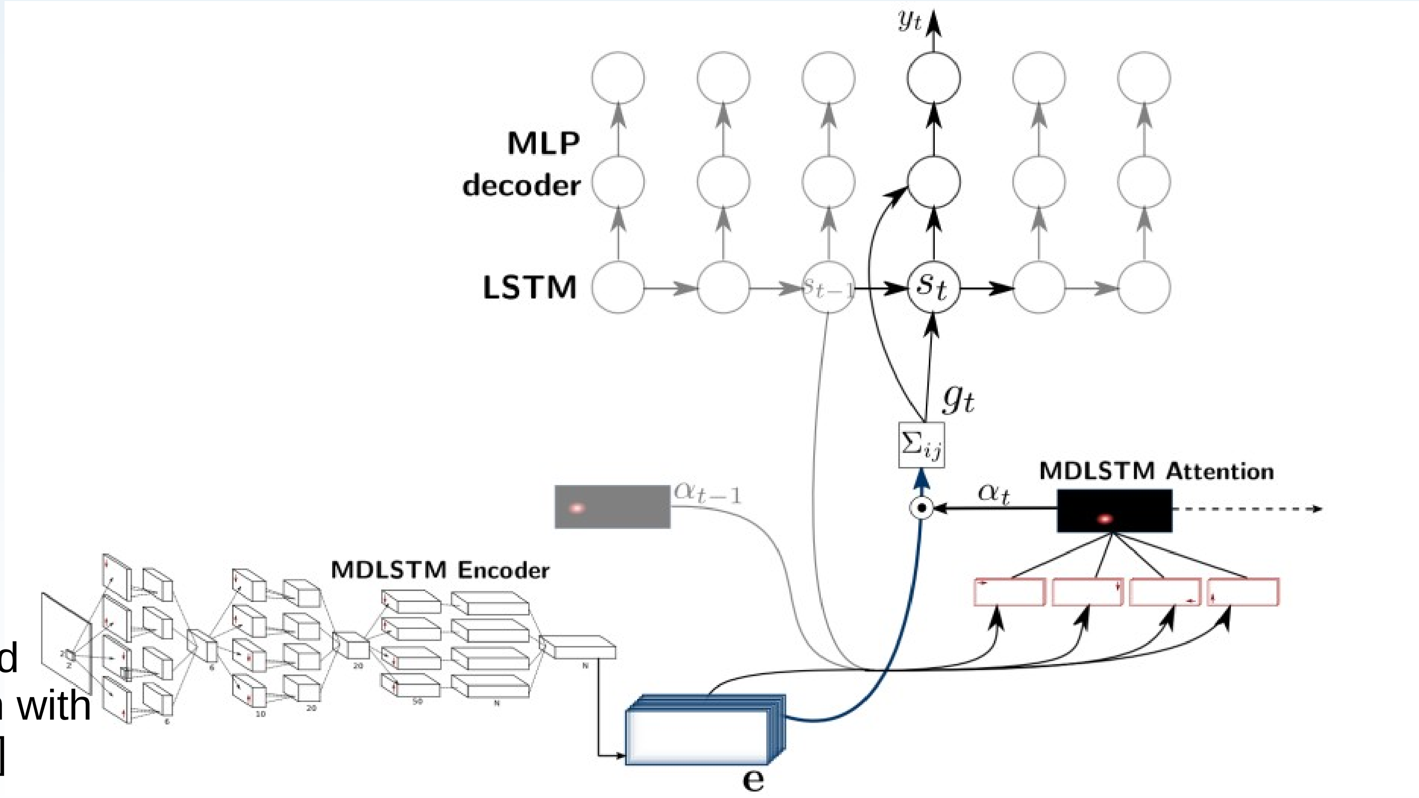
Attention + LSTM in Math Reco

- Attention on symbol and over the relationships…



MDLSTM + Attention + LSTM

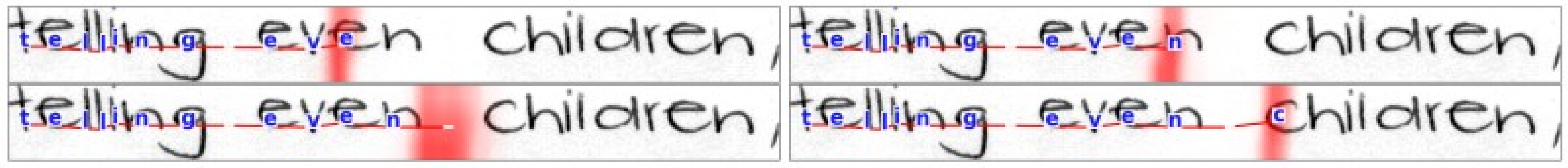
- MDLSTM & CNN layers extract shape features
- Attention module extract local features
- LSTM decode characters one by one



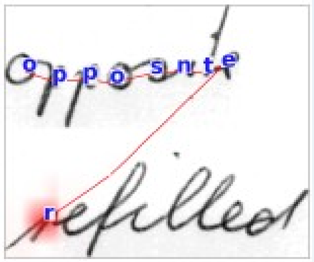
[Scan, Attend and Read: End-to-End Handwritten Paragraph Recognition with MDLSTM Attention, *T. Bluche 2016*]

MDLSTM + Attention + LSTM

- Line level

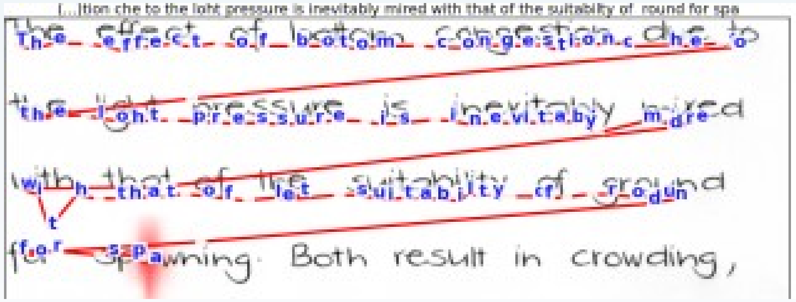


- Line breaks



- Paragraph level

  demo


[...]ition che to the loht pressure is inevitably mired with that of the suitability of round for spa

[Scan, Attend and Read: End-to-End Handwritten Paragraph Recognition with MDLSTM Attention, *T. Bluche 2016*]