Re-discovering the graphical structure of Chinese characters

Y. Lepage

IPS, Waseda university

SAMAI 2012

The structure of Chinese characters

6 categories for the constitution of individual characters according to Chinese tradition (Xǔ Shèn, 說文解字 /shuōwén jiězì/, II AD):

- 象形 pictograms: 月 'moon'
- 指事 symbols: 上 'above'
- 形声 radical-pronunciation: 泣 'to cry'
- 会意 composition: 林 'forest'
- 転注 modification: 長 'long' or 'to grow' depending on tone
- 仮借 borrowing: 求 'leather clothing' being used for 'to seek', replaced by 裘 = same + 衣 'clothing'

The structure of Chinese characters

Usual claim: about 80% of the Chinese characters belong to the category 形声, i.e., semantic clue (called radical) + pronunciation part.

The structure of Chinese characters

Usual claim: about 80% of the Chinese characters belong to the category 形声, i.e., semantic clue (called radical) + pronunciation part.

泣	=	氵	+	立
'to cry'		'water'		/lì/
meaning	=	semantic clue	+	pronunciation

The structure of Chinese characters

Usual claim: about 80% of the Chinese characters belong to the category 形声, i.e., semantic clue (called radical) + pronunciation part.

泣	=	氵	+	立
'to cry'		'water'		/lì/
meaning	=	semantic clue	+	pronunciation

京:先	identical left part (semantic key)
凉:洗	氵 [water]
凉:洗	冫 [ice]
俍:佌	亻 [human]

The structure of Chinese characters

Usual claim: about 80% of the Chinese characters belong to the category 形声, i.e., semantic clue (called radical) + pronunciation part.

泣	=	氵	+	立
'to cry'		'water'		/lì/
meaning	=	semantic clue	+	pronunciation

京:先	identical left part (semantic key)
凉:洗	氵 [water]
凉:冼	冫 [ice]
侇:佌	亻 [human]

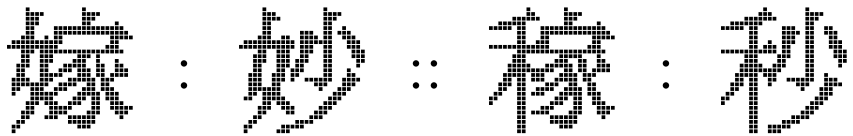
氵:亻	identical right part (pronunciation)	
洋:佯	羊	/yáng/
凉:侇	京	/liàng/
冼:佌	先	/ēn/

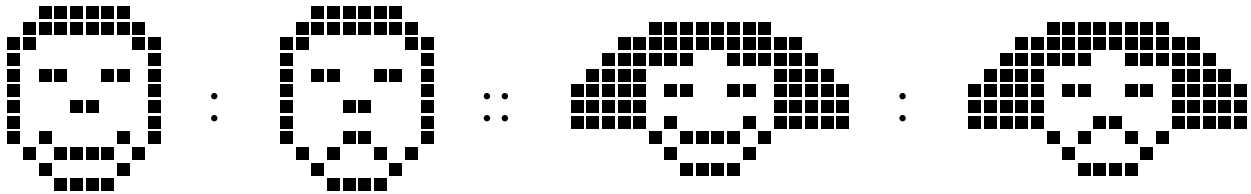
Can we substantiate this claim using analogy?

泣	:	立	::	泠	:	令	
/lì/	:	/lì/	::	/líng/	:	/líng/	
/lì/	:	/lì/	::	/lǐng/	:	/lǐng/	

Analogies between graphical objects

First necessary step: retrieve all analogies between the graphical forms of the characters themselves.





Background: explore the ways analogy structures language units following linguistic insights (Varro, Paul, Saussure, Bloomfield, Itkonen)

- Estimation of the number of true analogies (form and meaning) between short sentences in corpora
- Estimation of the number of true analogies between chunks in different languages (Japanese and English)
- Measure of proximity of languages by commonality in the structure of their vocabularies
- Use of analogy in machine translation and paraphrasing

A word about analogy

Proportional analogies

Most ancient definitions of *proportional analogy*:

> *Four objects A, B, C and D, are in analogical relation (proportional analogy) if the first object is to the second object in the same way as the third object is to the fourth object. Proportional analogies are noted* $A : B :: C : D$.

In all generality:

- *ratio* = relation between two objects (the colon :)
- *conformity* = relation between two ratios (the two colons ::)

Proportional analogy = conformity of ratios of two pairs of objects of the same kind.

An important remark

Conformity is not always an equivalence relation (*i.e.*, reflexive, symmetrical and transitive).

For instance, between strings of symbols, transitivity in the general case would imply:

$$A : A :: C : D \quad \text{possible with } C \neq D.$$

This would be similar to admitting *ex falso sequitur quodlibet* in logic.

General properties of proportional analogies

$A : B :: C : D$	
$A : C :: B : D$	exch. means
$B : A :: D : C$	exch. means + sym. :: + exch. means
$B : D :: A : C$	exch. means + sym. ::
$C : A :: D : B$	sym. :: + exch. means
$C : D :: A : B$	sym. ::
$D : B :: C : A$	sym. :: + exch. means + sym. ::
$D : C :: B : A$	exch. means + sym. :: + exch. means + sym. ::

Analogies between vectors

$$\begin{pmatrix} 3 \\ 6 \\ 10 \\ 7 \end{pmatrix} - \begin{pmatrix} 2 \\ 6 \\ 2 \\ 3 \end{pmatrix} = \begin{pmatrix} 10 \\ 8 \\ 9 \\ 5 \end{pmatrix} - \begin{pmatrix} 9 \\ 8 \\ 1 \\ 1 \end{pmatrix}$$

Object	=	vector of numerical values
Ratio	=	difference between vectors
Conformity	=	equality of vectors

In this setting, conformity is transitive.
Setting applicable to pixel images by converting an image into some vector.

Analogical clusters

Naïvely, the problem is basically $O(n^4)$.

naïve approach
$A : B :: C : D$ $C : D :: E : F$ $A : B :: E : F$
$O(n^4)$
enumeration of all analogies

Naïvely, the problem is basically $O(n^4)$.
In fact, equivalent to a problem of complexity $O(n^2)$.

$$A : B :: C : D \quad \text{and} \quad C : D :: E : F \quad \Rightarrow \quad A : B :: E : F$$

naïve approach	use of transitivity
$A : B :: C : D$ $C : D :: E : F$ $A : B :: E : F$	
$O(n^4)$	
enumeration of all analogies	

Naïvely, the problem is basically $O(n^4)$.
In fact, equivalent to a problem of complexity $O(n^2)$.

$$A : B :: C : D \quad \text{and} \quad C : D :: E : F \quad \Rightarrow \quad A : B :: E : F$$

naïve approach		use of transitivity
$A : B :: C : D$		$A : B$
$C : D :: E : F$	$\Leftrightarrow$	$C : D$
$A : B :: E : F$		$E : F$
$O(n^4)$		
enumeration of all analogies		

Naïvely, the problem is basically $O(n^4)$.
In fact, equivalent to a problem of complexity $O(n^2)$.

$$A : B :: C : D \quad \text{and} \quad C : D :: E : F \quad \Rightarrow \quad A : B :: E : F$$

naïve approach		use of transitivity
$A : B :: C : D$		$A : B$
$C : D :: E : F$	$\Leftrightarrow$	$C : D$
$A : B :: E : F$		$E : F$
$O(n^4)$		$O(n^2)$
enumeration of all analogies		

Naïvely, the problem is basically $O(n^4)$.
In fact, equivalent to a problem of complexity $O(n^2)$.

$$A : B :: C : D \quad \text{and} \quad C : D :: E : F \quad \Rightarrow \quad A : B :: E : F$$

naïve approach		use of transitivity
$A : B :: C : D$		$A : B$
$C : D :: E : F$	$\Leftrightarrow$	$C : D$
$A : B :: E : F$		$E : F$
$O(n^4)$		$O(n^2)$
enumeration of all analogies		enumeration of all ratios

Improvements for the enumeration of all ratios

- Properties of analogy
  - (i) Elimination of redundant clusters (symmetry of ratio)
  - (ii) Avoiding the trivial cluster (reflexivity of conformity)
- Quantity of information
  - (iii) Elimination of clusters reduced to one ratio
  - (iv) Conditional elimination of clusters reduced to one analogy

(i) Elimination of redundant clusters

Cluster number			
(1)	(2)	(3)	(4)
$A : B$	$\vdots$	$\vdots$	$\vdots$
$\vdots$	$B : D$	$B : A$	$C : A$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$C : D$	$A : C$	$\vdots$	$D : B$
$\vdots$	$\vdots$	$D : C$	$\vdots$

(i) Elimination of redundant clusters

Cluster number			
(1)	(2)	(3)	(4)
$A : B$	$\vdots$	$\vdots$	$\vdots$
$\vdots$	$B : D$	$B : A$	$C : A$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$C : D$	$A : C$	$\vdots$	$D : B$
$\vdots$	$\vdots$	$D : C$	$\vdots$

As difference between vectors: $(1) = -(3)$ and $(2) = -(4)$.

(i) Elimination of redundant clusters

Cluster number			
(1)	(2)	(3)	(4)
$A : B$	$\vdots$	$\vdots$	$\vdots$
$\vdots$	$B : D$	$B : A$	$C : A$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$C : D$	$A : C$	$\vdots$	$D : B$
$\vdots$	$\vdots$	$D : C$	$\vdots$

As difference between vectors: $(1) = -(3)$ and $(2) = -(4)$.
Impose an order on vectors to compute ratios between u and v only when $u \leq v$

(i) Elimination of redundant clusters

Cluster number		
(1)	(2)	
$A : B$	$\vdots$	
$\vdots$	$B : D$	
$\vdots$	$\vdots$	
$C : D$	$A : C$	
$\vdots$	$\vdots$	

As difference between vectors: $(1) = -(3)$ and $(2) = -(4)$.
Impose an order on vectors to compute ratios between u and v only when $u \leq v \quad \Rightarrow$ from n^2 to $n^2/2$.

(ii) Avoiding the trivial cluster

Null ratio = set of all trivial analogies: $A : A :: B : B$.

$$\begin{array}{c} A : A \\ B : B \\ C : C \\ \vdots \end{array}$$

(ii) Avoiding the trivial cluster

Null ratio = set of all trivial analogies: $A : A :: B : B$.

$$\begin{array}{c} A : A \\ B : B \\ C : C \\ \vdots \end{array}$$

$\Rightarrow$ Enumerate ratios strictly above the diagonal, i.e., compute ratios between u and v only when $u < v$.

(iii) Elimination of non-informative clusters

Non-informative clusters = clusters that contain only one ratio, i.e., one pair of objects: $A : B$ and thus represent only one trivial analogy: $A : B :: A : B$

(iii) Elimination of non-informative clusters

Non-informative clusters = clusters that contain only one ratio, i.e., one pair of objects: $A : B$ and thus represent only one trivial analogy: $A : B :: A : B$

Early detection of such cases $\Rightarrow$ Reduction in processing time and memory.

(iii) Elimination of non-informative clusters

number of chars processed	runtimes in seconds without	with	time reduction in percentage
1,000	9	14	+55 %
2,000	39	36	-7 %
3,000	92	82	-10 %
4,000	173	142	-17 %
5,000	277	219	-20 %
6,000	426	313	-26 %
7,000	605	438	-27 %
8,000	739	557	-24 %
9,000	944	702	-25 %
10,000	1204	836	-30 %
11,000	1517	1123	-25 %
12,000	1864	1302	-30 %
13,000	2265	1342	-40 %
14,000	2646	1791	-32 %
14,655	2873	1889	-34 %

(iv) Conditional elimination of clusters reduced to one analogy

$$\begin{array}{ccc} A & : & B \\ C & : & D \end{array} \quad \Bigg| \quad \begin{array}{ccc} \vdots & & \\ A & : & C \\ \vdots & & \\ B & : & D \\ \vdots & & \end{array}$$

(iv) Conditional elimination of clusters reduced to one analogy

$$\begin{array}{ccc} A & : & B \\ C & : & D \end{array} \quad \Bigg| \quad \begin{array}{ccc} \vdots & & \\ A & : & C \\ \vdots & & \\ B & : & D \\ \vdots & & \end{array}$$

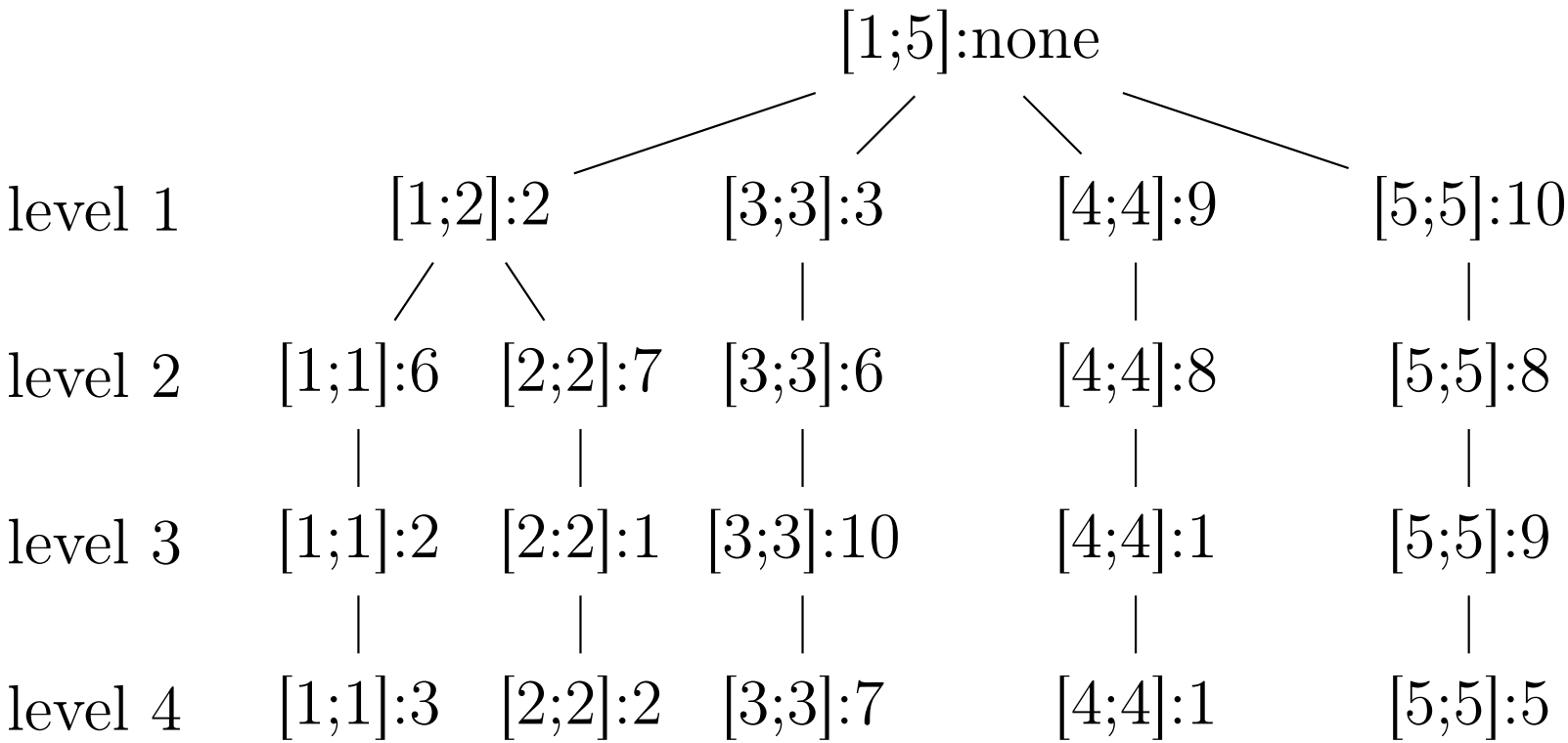
Elimination of clusters reduced to one analogy $\Rightarrow$ Reduction in memory used.

Program = Data structure + Algorithm

Data structure used: feature tree

$$\begin{array}{ccccc} [& 1 & 2 & 3 & 4 & 5 &] \\ \hline & \begin{pmatrix} 2 \\ 6 \\ 2 \\ 3 \end{pmatrix} & \begin{pmatrix} 2 \\ 7 \\ 1 \\ 2 \end{pmatrix} & \begin{pmatrix} 3 \\ 6 \\ 10 \\ 7 \end{pmatrix} & \begin{pmatrix} 9 \\ 8 \\ 1 \\ 1 \end{pmatrix} & \begin{pmatrix} 10 \\ 8 \\ 9 \\ 5 \end{pmatrix} & \end{array}$$

Data structure used: feature tree



Exploration of space 1/2

	[1;2]:2	[3;3]:3	[4;4]:9	[5;5]:10
[1;2]:2	[1;2]x[1;2]:0	[1;2]x[3;3]:1	[1;2]x[4;4]:7	[1;2]x[5;5]:8
[3;3]:3			[3;3]x[4;4]:6	[3;3]x[5;5]:7
[4;4]:9				[4;4]x[5;5]:1
[5;5]:10				

Exploration of space 2/2

	[4;4]:9	[5;5]:8
[1;1]:6	[1;1]x[4;4]:3	
[2;2]:7	[2;2]x[4;4]:2	
[3;3]:6		[3;3]x[5;5]:2

Summary: overall sketch of the method

- Convert each object into a feature vector
- Check for separation of space = List equivalent objects
- Define order on feature vectors (we use least correlations of values among features)
- Sort feature vectors according to lexicographic order in the defined order of features
- Build feature tree for the sorted feature vectors
- Compute the differences between the feature vectors by blocks = Traverse feature tree in parallel in breadth-first order, above first diagonal only, applying the 4 possible improvements
- Output list of pairs of intervals corresponding to each different possible ratio

Application to Chinese characters

Data

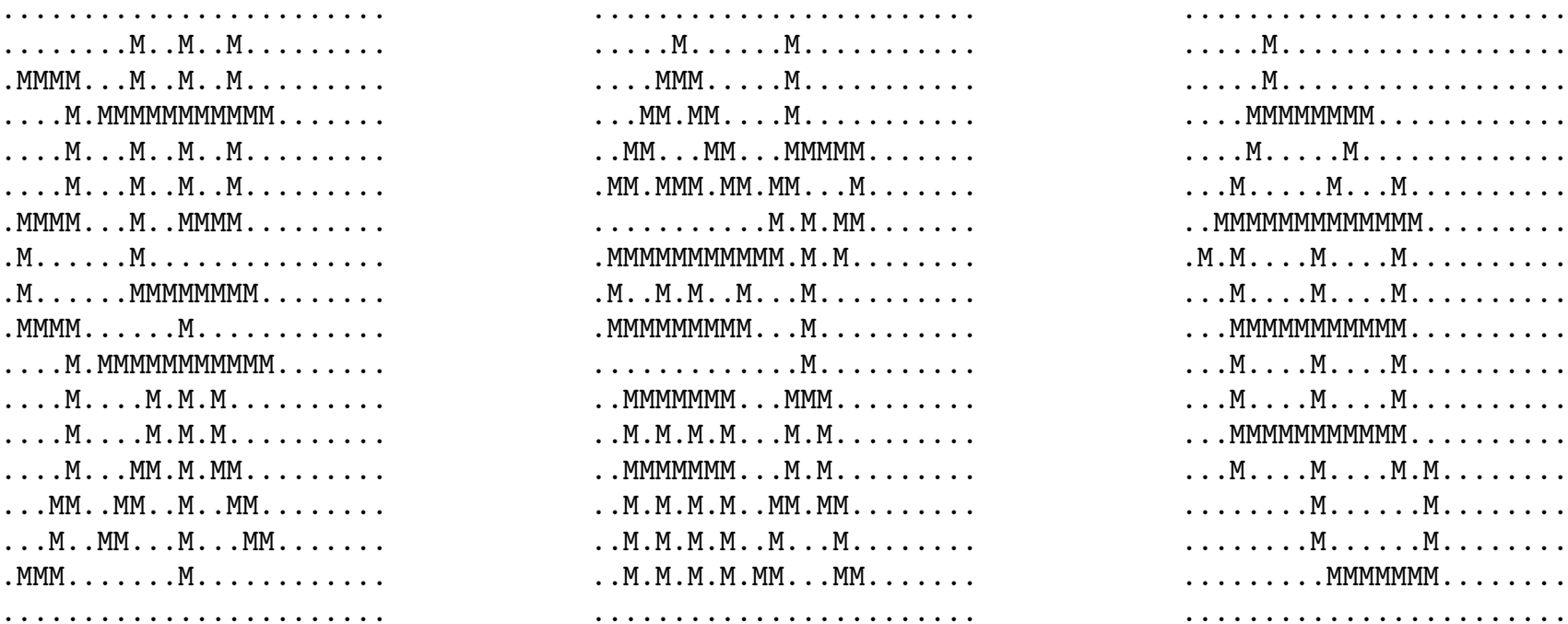
Monospace (or fixed-width or fixed-size) fonts are lists of characters described as black and white icons of fixed height and width.

- Font used: knj10B.bdf[1], a 18 × 18 pixel font.
- Each character has a fixed height of 18 lines and a fixed width of 24 pixels (each line is encoded on 3 bytes, but actual width is 18 pixels).
- We use the 14,655 Sino-Japanese characters available in this font in the range between the Unicode codepoints 13,312 (㐀) and 40,891 (鿋).

[1] Designed by Nagao Sadakazu (snagao@tkb.att.ne.jp), version 1.1 of 1999.

Characters as black and white icons

Visualization of some characters of the font knj10B as icons



Conversion of pixel images into vectors of numerical features

```
.........................
........M..M..M.........
.MMMM...M..M..M.........
....M.MMMMMMMMMMM.......
....M...M..M..M.........
....M...M..M..M.........
.MMMM...M..MMMM.........
.M......M...............
.M......MMMMMMMM........
.MMMM......M............
....M.MMMMMMMMMMM.......
....M....M.M.M..........
....M....M.M.M..........
....M...MM.M.MM.........
...MM..MM..M..MM........
...M..MM...M...MM.......
.MMM.......M............
.........................
```

Conversion of pixel images into vectors of numerical features

```
.......................  0
........M..M..M.........
.MMMM...M..M..M.........
....M.MMMMMMMMMMM.......
....M...M..M..M.........
....M...M..M..M.........
.MMMM...M..MMMM.........
.M......M...............
.M......MMMMMMMM........
.MMMM......M............
....M.MMMMMMMMMMM.......
....M....M.M.M..........
....M....M.M.M..........
....M...MM.M.MM.........
...MM..MM..M..MM........
...M..MM...M...MM.......
.MMM.......M............
.......................
```

Conversion of pixel images into vectors of numerical features

```
.......................  0
........M..M..M.........  3
.MMMM...M..M..M.........
....M.MMMMMMMMMMM.......
....M...M..M..M.........
....M...M..M..M.........
.MMMM...M..MMMM.........
.M......M...............
.M......MMMMMMMM........
.MMMM......M............
....M.MMMMMMMMMMM.......
....M....M.M.M..........
....M....M.M.M..........
....M...MM.M.MM.........
...MM..MM..M..MM........
...M..MM...M...MM.......
.MMM.......M............
.......................
```

Conversion of pixel images into vectors of numerical features

```
.......................  0
........M..M..M.........  3
.MMMM...M..M..M.........  7
....M.MMMMMMMMMMM.......
....M...M..M..M.........
....M...M..M..M.........
.MMMM...M..MMMM.........
.M......M...............
.M......MMMMMMMM........
.MMMM......M............
....M.MMMMMMMMMMM.......
....M....M.M.M..........
....M....M.M.M..........
....M...MM.M.MM.........
...MM..MM..M..MM........
...M..MM...M...MM.......
.MMM.......M............
.......................
```

Conversion of pixel images into vectors of numerical features

```
.........................  0
........M..M..M.........  3
.MMMM...M..M..M.........  7
....M.MMMMMMMMMMM.......  12
....M...M..M..M.........  4
....M...M..M..M.........  4
.MMMM...M..MMMM.........  9
.M......M...............  2
.M......MMMMMMMM........  9
.MMMM......M............  5
....M.MMMMMMMMMMM.......  12
....M....M.M.M..........  4
....M....M.M.M..........  4
....M...MM.M.MM.........  6
...MM..MM..M..MM........  7
...M..MM...M...MM.......  6
.MMM.......M............  4
.........................  0
```

Conversion of pixel images into vectors of numerical features

```
.......................  0
........M..M..M.........  3
.MMMM...M..M..M.........  7
....M.MMMMMMMMMMM.......  12
....M...M..M..M.........  4
....M...M..M..M.........  4
.MMMM...M..MMMM.........  9
.M......M...............  2
.M......MMMMMMMM........  9
.MMMM......M............  5
....M.MMMMMMMMMMM.......  12
....M....M.M.M..........  4
....M....M.M.M..........  4
....M...MM.M.MM.........  6
...MM..MM..M..MM........  7
...M..MM...M...MM.......  6
.MMM.......M............  4
.......................  0
```

0

Conversion of pixel images into vectors of numerical features

```
.........................  0
........M..M..M.........  3
.MMMM...M..M..M.........  7
....M.MMMMMMMMMMM.......  12
....M...M..M..M.........  4
....M...M..M..M.........  4
.MMMM...M..MMMM.........  9
.M......M...............  2
.M......MMMMMMMM........  9
.MMMM......M............  5
....M.MMMMMMMMMMM.......  12
....M....M.M.M..........  4
....M....M.M.M..........  4
....M...MM.M.MM.........  6
...MM..MM..M..MM........  7
...M..MM...M...MM.......  6
.MMM.......M............  4
.........................  0
```

06

Conversion of pixel images into vectors of numerical features

```
.........................  0
........M..M..M.........  3
.MMMM...M..M..M.........  7
....M.MMMMMMMMMMM.......  12
....M...M..M..M.........  4
....M...M..M..M.........  4
.MMMM...M..MMMM.........  9
.M......M...............  2
.M......MMMMMMMM........  9
.MMMM......M............  5
....M.MMMMMMMMMMM.......  12
....M....M.M.M..........  4
....M....M.M.M..........  4
....M...MM.M.MM.........  6
...MM..MM..M..MM........  7
...M..MM...M...MM.......  6
.MMM.......M............  4
.........................  0

....1...1..1..1.........
06461034163547053000000
```

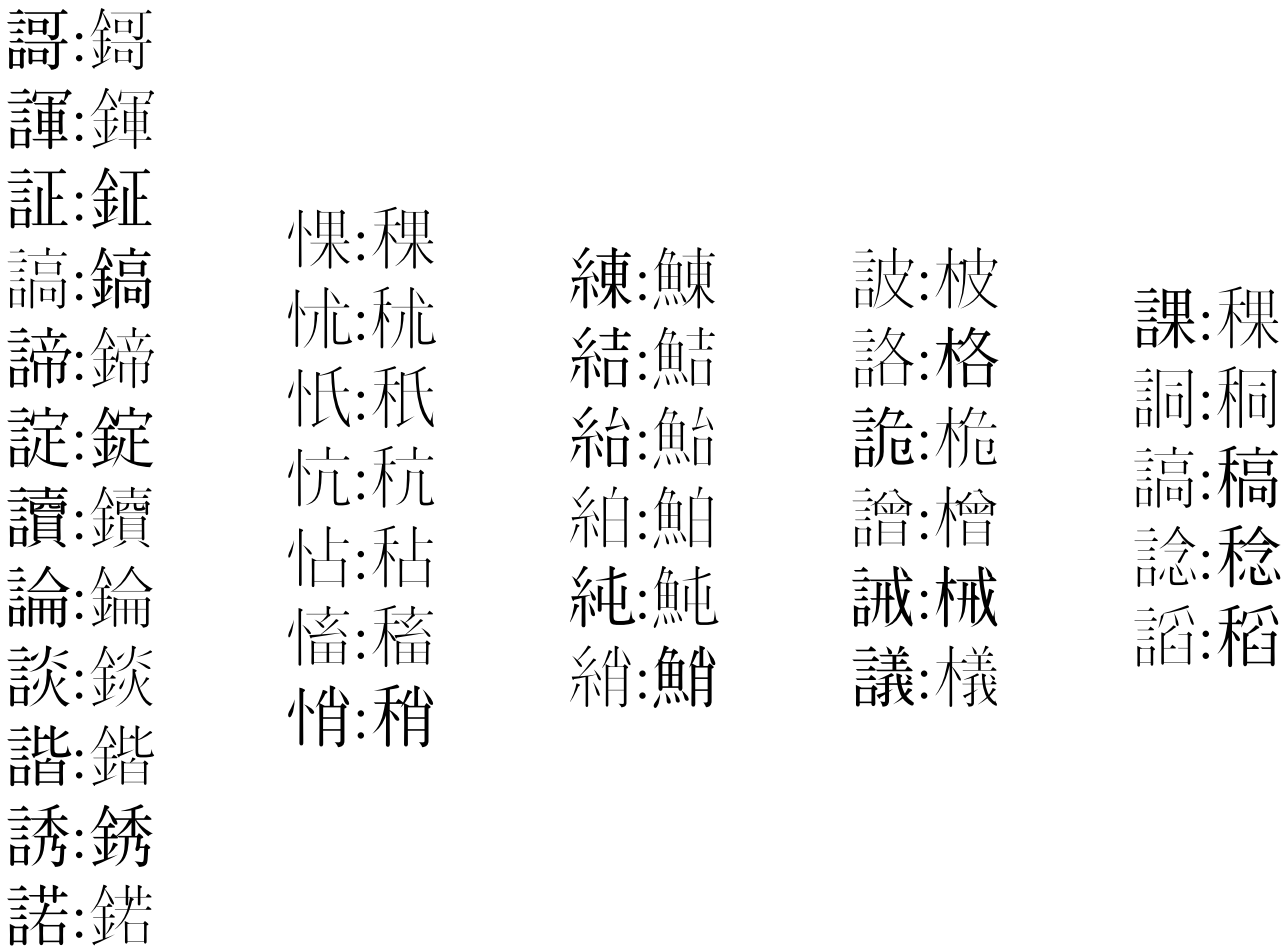
Conversion of pixel images into vectors of numerical features

```
.........................  0
........M..M..M.........  3
.MMMM...M..M..M.........  7
....M.MMMMMMMMMMM.......  12
....M...M..M..M.........  4
....M...M..M..M.........  4
.MMMM...M..MMMM.........  9
.M......M...............  2
.M......MMMMMMMM........  9
.MMMM......M............  5
....M.MMMMMMMMMMM.......  12
....M....M.M.M..........  4
....M....M.M.M..........  4
....M...MM.M.MM.........  6
...MM..MM..M..MM........  7
...M..MM...M...MM.......  6
.MMM.......M............  4
.........................  0

....1...1..1..1.........
064610341635470530000000
```

$18 + 24 = 42$ features. . . or less

Clusters obtained (sample): typical clusters

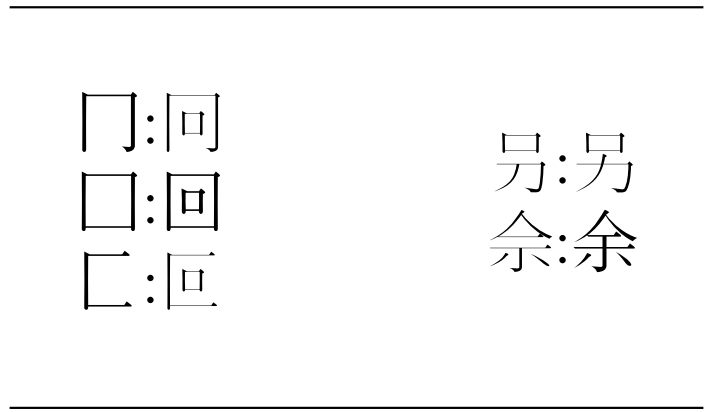


Clusters obtained (sample): typical clusters

凉:倞	玿:璹	冷:泮	潺:沼	悍:猂
泮:伴	沼:濤	泠:泮	璆:玿	怚:狙
津:佯	招:擣	拎:拌	僇:侶	恄:猘
冼:侁	侶:儔			

謒:談	璹:玿	曋:曈	找:括	鎮:稹
槁:棪	濤:滔	澤:潼	戕:聒	鎬:稿
鎬:錏	擣:掐	擇:撞		

Non-typical clusters



Non-typical cluster obtained with less features

Features: # of 'on' pixels on each line and col.

Number of pairs in cluster	Number of clusters
2	216
3	451
4	66
5	25
6	19
7	17
8	10
9	5
10	2
11	3
12	4
13	1

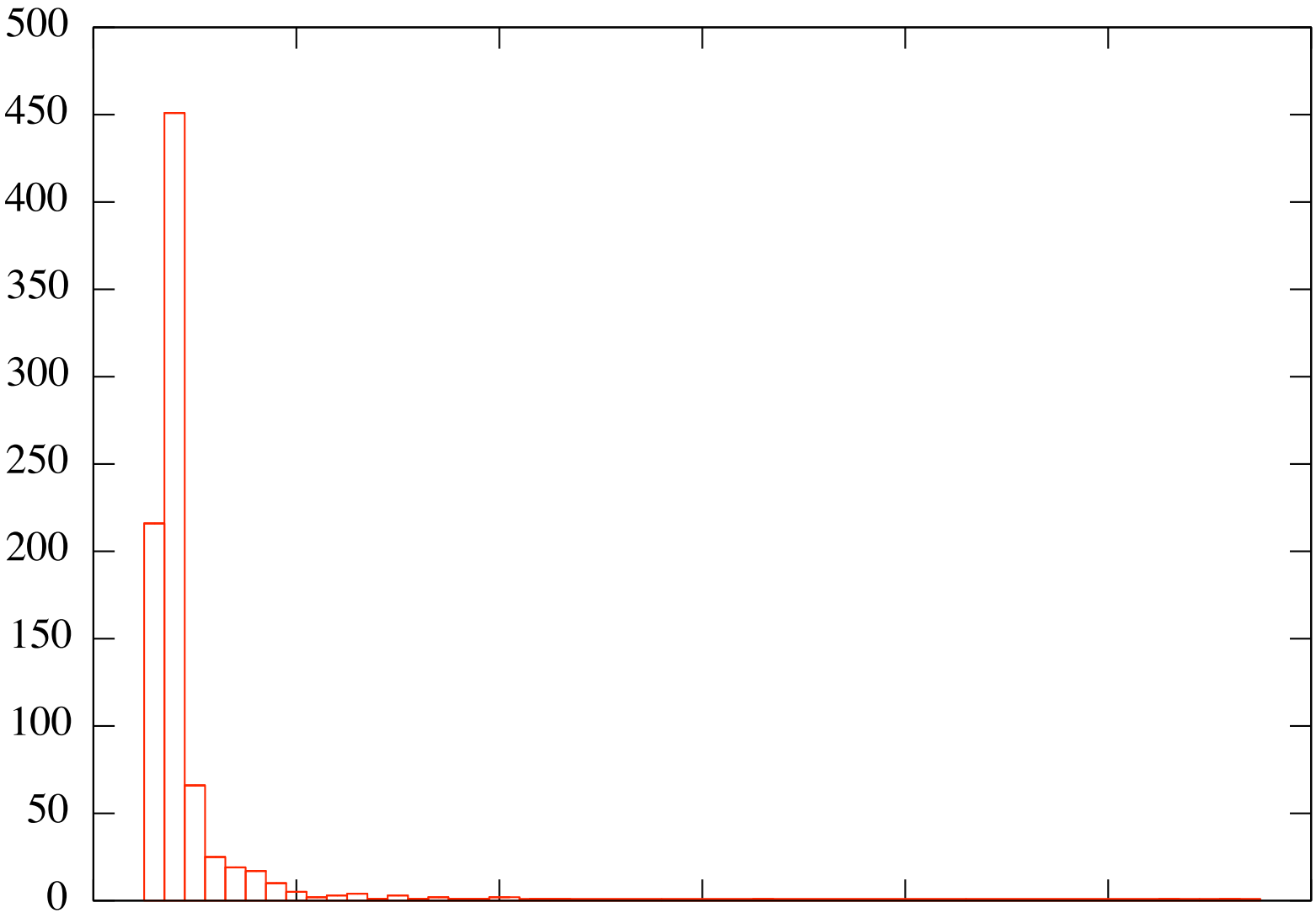
Number of pairs in cluster	Number of clusters
14	3
15	1
16	2
17	1
18	1
19	2
21	1
22	1
32	1
52	1
55	1

Features: 'on/off' pixels themselves

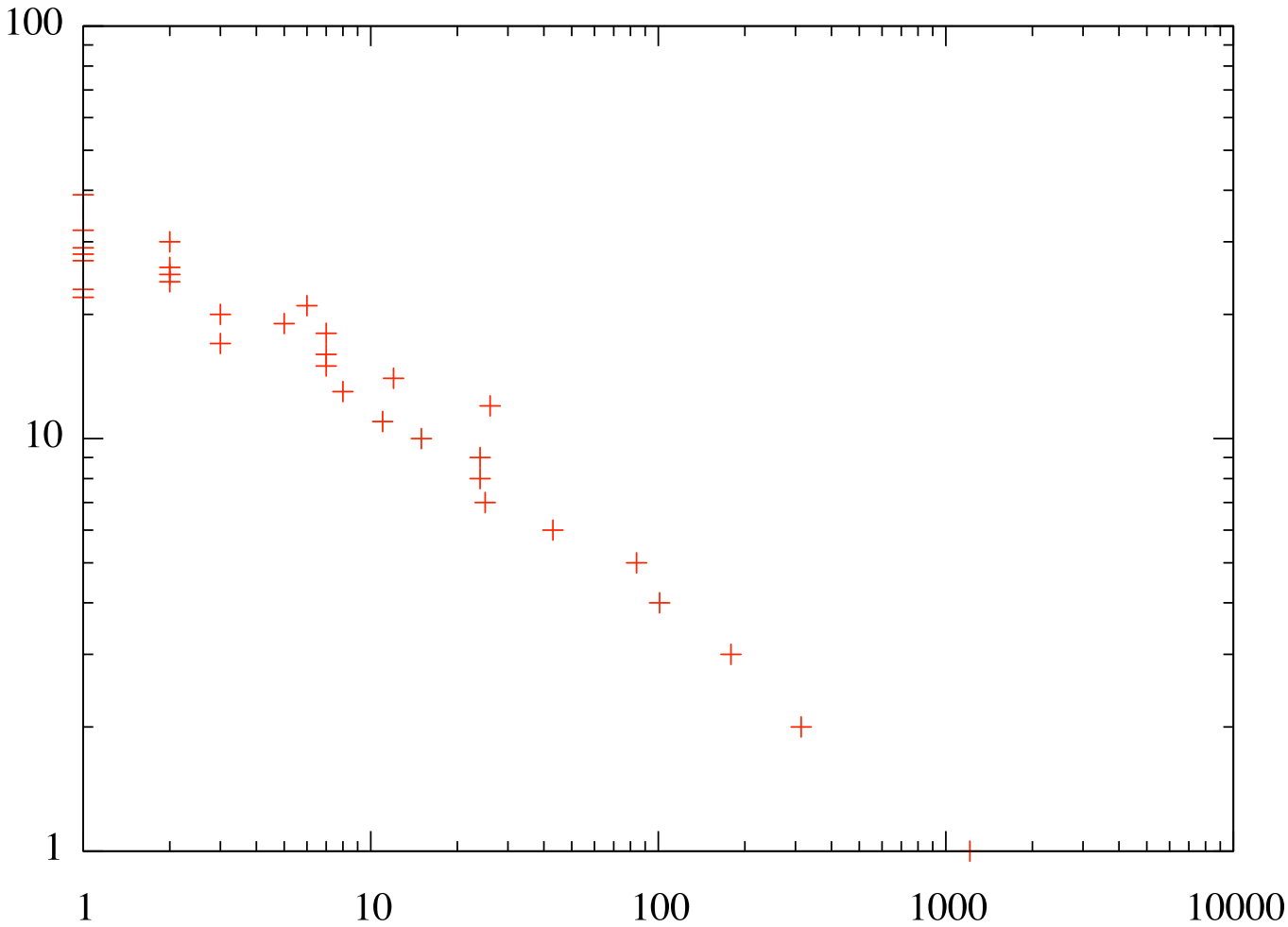
Number of pairs in cluster	Number of clusters
2	213
3	427
4	66
5	23
6	19
7	17
8	10
9	5
10	3
11	3
12	4
13	2

Number of pairs in cluster	Number of clusters
14	1
15	1
16	2
17	1
18	1
19	2
21	2
32	1
50	1
53	1

of clusters (ordin.) with same # of pairs (absc.)



of clusters (ordinates) per char (abscissae)



Total number of characters appearing in clusters: 5,982 over 14,655 used = 41%.

Conclusion

- A first attempt at applying analogy to pixel images of the same size by converting them into feature vectors
- A general method for enumerating all analogies between all objects in a set of objects, represented as numerical feature vectors, by enumerating all possible ratios (analogical clusters)
- Few preliminary results on automatically rediscovering the structure of Chinese characters

Open problem

- Selection of features (horizontal, vertical, diagonal lines, circles?)
- Shift in images not handled
- Scaling of images not handled

Do not all these problems reduce to the same problem, that of simulating human visual perception?

Thanks for listening 谢谢