(Re-)discovering the graphical structure of Chinese characters

Yves Lepage[1]

Abstract. The purpose of this paper is to show how it is possible to efficiently extract the structure of a set of objects by use of the notion of proportional analogy. As a proportional analogy involves four objects, the very naïve approach to the problem, has basically a complexity of $O(n^4)$ for a given set of n objects. We show, under some conditions on proportional analogy, how to reduce this complexity to $O(n^2)$ by considering an equivalent problem, that of enumerating analogical clusters that are informative and not redundant. We further show how some improvements make the task tractable. We illustrate our technique with a task related with natural language processing, that of clustering Sino-Japanese characters. In this way, we re-discover the graphical structure of these characters.

1 INTRODUCTION

1.1 Background

Analogy is defined in various ways by different recent authors [1, 7, 21]. Referring back to the most ancient definitions, one can reach an agreement on the following definition 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 analogy is noted $A : B :: C : D$.

In all generality, if the relation between two objects (noted by the colon :) is called a *ratio* and the relation between the two pairs of objects (noted by the two colons ::) is called a *conformity,* then proportional analogy is a conformity of ratios between two pairs of objects.

Proportional analogy can be seen between words on the level of form or on the level of meaning or on both at the same time (see [2] for abnormal cases).

Form but not meaning:

walk : *walked* :: *he* : *heed*

Meaning but not form:

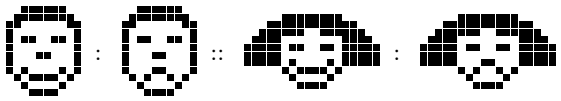
to walk : *walked* :: *to be* : *was*

Form and meaning:

to walk : *walked* :: *to work* : *worked*

Proportional analogies on the levels of form and meaning at the same time are called true analogies. Between chunks or short sentences, their number has been shown to be quite important [10, 12, 13]. Many studies, too many to cite here, address the efficiency of analogy for segmenting words or grouping them according to word families (as for Chinese, see for instance [19]). Forms which depart from declension or conjugation paradigms (groups of proportions in [14]) were called anomalies in Classical grammar [18]. Recently, analogies between word meanings (*water* : *riverbed* :: *traffic* : *road*) have been shown to be reproduceable on computers using large corpora and vector space models [17, 16, 15].

Proportional analogies are not only verbal. They may hold between any kind of objects provided, generally, that the objects be of the same kind, a point Aristotle, among other ancient and recent authors, insists on. The general principle, viewed as a cognitive process, is based on iconicity [3]. Taking the term to its restrained graphical sense, the following two examples illustrate proportional analogies between icons of black and white pixels.



嫁 : 妙 :: 稼 : 秒

The latter example is a graphical proportional analogy between four Sino-Japanese characters. By explicitly decomposing into constitutive elements, as in [20], to compute similarity between Sino-Japanese characters, it is understandable that the left and the right parts of the characters can be exchanged to give rise to the four different characters. The above analogy does not apply on the level of meaning, as the character meanings are unrelated: 'spouse', 'odd', 'to earn' and 'second (measure of time)'. It does not make an analogy on the level of pronunciation either.

The particular and practical problem which we tackle in a broader research concerned with ease of learning of Chinese characters, is to re-discover the graphical structure of Chinese characters in an automatic way by relying on the notion of proportional analogy. The general and theoretical problem that we thus tackle in the following sections is to rely on the properties of proportional analogies so as to automatically visualize the structure of a set of objects.

1.2 The problem

The naïve approach to the problem of the enumeration of all proportional analogies between a set of n objects consists in examining all possible quadruples of objects and checking for analogy. This naïve approach has a complexity of $O(n^4)$.

[1] IPS, Waseda University, Japan. Email: yves.lepage@waseda.jp

Without changing the complexity, the computation time may be reduced. For a given proportional analogy, there exists seven other equivalent forms (see Theorem 2.1 in [9]). This is implied by the basic properties of exchange of the means (exchanging objects B and C in the analogy $A : B :: C : D$, second line below) and symmetry of conformity (exchanging the terms on both sides of the :: sign, sixth line below). In this way, the following eight analogies are shown to be equivalent:

$A : B :: C : D$	
$A : C :: B : D$	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. ::

Because of these eight equivalent forms, the enumeration time can be divided by a factor of 8, but the complexity remains $O(n^4)$.

To make our point clear, consider the following naïve estimation. In a preliminary experiment, we estimated the average time needed for the verification of one analogy between four Sino-Japanese characters using 36 features (see Section 5.3 for a description of the features). An average time of 0.8 ms was measured. For almost fifteen thousand Sino-Japanese characters (see Section 5.2 for a description of the data), and knowing that there are approximately 3.2×10^7 seconds in a year, the time needed to compute all possible analogies would exceed a million years.[2]

In order to reduce the complexity of this problem, we propose to modify our goal. Rather than aiming at individual analogies, we compute all possible ratios between all possible objects at hand. This computation is basically $O(n^2)$. The result of this computation allows us to cluster pairs of objects according to their ratios. These clusters summarize all possible analogies between all objects in a non-redundant way that still provides the total amount of information (see Section 2). The sequel of the paper shows how to compute such clusters and presents some of the actual results of such a computation on a set of Sino-Japanese characters.

The paper is structured as followed: Section 2 shows how the problem can be transformed into a problem of quadratic complexity and introduces the notion of analogical clusters for this purpose. Section 3 gives our proposed method to output analogical clusters. Section 4 mentions some improvements that can reduce computational time. Section 5 describes the application of the proposed method to the problem of structuring Sino-Japanese characters, and gives the results obtained in our experiments.

2 ANALOGICAL CLUSTERS

2.1 Objects as feature vectors

In this work, we represent an object by a vector of features with numerical values. We also impose that the feature space be the same for all objects, so that it is trivially possible to define a ratio between two objects as the vector of their difference. In such a setting, conformity is trivially reduced to equality between vectors.

The following equation illustrates a possible case of proportional analogy between vectors in a four-dimensional space.

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

Vector difference as a ratio, and equality between vectors as conformity, consistently define analogies that meet the intuitive notions about proportional analogies. Among other properties, the eight forms of equivalence for the same proportional analogy (see above Section 1.2) always hold.

2.2 Transitivity of conformity: analogical clusters

It is not always the case that conformity verifies transitivity. For instance, [8] shows that the intuitive notions of proportional analogy between strings of characters imply that there is no transitivity for conformity.[3]

In our setting with conformity being an equality, i.e., an equivalence relation, transitivity naturally holds in addition to reflexivity and symmetry. For proportional analogies, transitivity of conformity implies that:

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

For our present task of enumerating all possible proportional analogies between all objects in a given set, a transitive conformity can lead to an enormous economy in representation. To illustrate this point, consider the following three proportional analogies.

$$\begin{array}{c} A : B :: C : D \\ C : D :: E : F \\ A : B :: E : F \end{array}$$

They can be represented in a more economical way by the following list of equal ratios:

$$\begin{array}{c} A : B \\ C : D \\ E : F \end{array}$$

All ratios being equal, any possible proportional analogy formed by taking any two ratios holds.

From the above example, it is clear that, provided conformity is transitive, a list of n pairs of objects with the same ratio stands for a list of $n \times (n-1) \ / \ 2$ non-trivial proportional analogies (see Section 2.5 for trivial analogies). Consequently, under the assumption of transitivity for conformity, the problem of enumerating all possible proportional analogies between all possible objects in a given set can be transformed into a problem of enumerating all possible pairs of objects with the same ratio. The former problem has a complexity of $O(n^4)$ while the latter one has a complexity of $O(n^2)$.

From now on, we shall call a list of ratios of objects with the same value, an *analogical cluster.*

[2] $14,655^4 \times 0.8$ ms $> 14^4 \times 10^{12} \times 0.8$ ms $> 48 \times 10^{12}$ s $> 48 \times 10^{12} \ / \ (3.2 \times 10^7)$ years $> 1.5 \times 10^6$ years

[3] This comes from the fact that some analogies between strings of characters admit multiple solutions. When this is the case, then, there is not transitivity for :: in the general case for the objects considered (see [8, p. 113]).

2.3 Equivalent forms of analogy: redundancy of clusters

Each analogical cluster stands for a different ratio, i.e., a vector that represents the difference between any two feature vectors each representing an object.

Because the order in analogical clusters is not relevant, an analogy extracted from an analogical cluster stands for two equivalent forms, obtained by symmetry of conformity (sixth line in the eight equivalent forms of proposition analogy in Section 1.2).

$$A : B :: C : D \;\Leftrightarrow\; C : D :: A : B$$

By inversion of ratios (third line in the eight equivalent forms of proposition analogy in Section 1.2), a proportional analogy involves two different ratios. And by exchange of the means, (second line in the eight equivalent forms of proposition analogy in Section 1.2), another two different ratios.

$$A : B :: C : D \;\Leftrightarrow\; B : A :: D : C \;\Leftrightarrow\; A : C :: B : D$$

Consequently, in total, the eight different forms of the same proportional analogy are to be found in four different analogical clusters (and only four clusters) among all the possible clusters that are output by a method yielding all the possible clusters standing for differences between all feature vectors representing all the objects in a given set.

Figure 1 shows such four analogical clusters for the proportional analogy $A : B :: C : D$. These four clusters are redundant because of the eight equivalent forms for the same proportional analogy, as we have just stated:

- In any cluster, the order of appearance of the pairs of objects being irrelevant, each cluster encapsulates two equivalent forms of the same proportional analogy. This is symmetry of conformity.
- Analogical clusters (1) and (2) together contain the same information as clusters (3) and (4) together. The relation between (1) and (2) (and between (3) and (4)) is the exchange of the means.
- Analogical clusters (1) and (3) are indeed the same up to an exchange of the objects on the left and the right of the : sign. This is actually inversion of ratios. The same is also true for clusters (2) and (4).

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

Figure 1. For a given proportional analogy $A : B :: C : D$, the set of analogical clusters output by a method that looks for all possible vector differences between all possible feature vectors representing objects in a given set should include four clusters.

It is trivially possible to eliminate the redundancy between clusters (1) and (2) and clusters (3) and (4). This can be done by avoiding the computation of the difference between two vectors and its opposite value (the same two vectors in the reverse order). For that, it suffices to sort all the vectors in some predefined increasing order, and to compute only the differences between two vectors ranked in that order. In this way, a particular proportional analogy will appear in two, and only two, different analogical clusters among the set of all clusters. As a result, globally, the set of all clusters will contain no redundant information.

2.4 Equality of feature vectors: separation of space

By definition of the ratio as a vector difference, the case where $A : B$ and $A : C$ belong to the same analogical cluster is only possible if the vectors representing B and C are the same. This can only happen if the feature vectors do not separate the space of objects into each individual object. Reciprocally, if the feature vectors are unique for each different objects in the given set, the two ratios $A : B$ and $A : C$ for different B and C will be different. For our proposed method, this implies to check for the *separation of the space of objects* before proceeding to clustering.

2.5 Trivial analogies: informativity of clusters

Finally, we must mention a particular case of no interest as it does not bring any information. This is the special case of the cluster for the null vector; i.e., null ratio. It has the following form.

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

It represents the set of all *trivial proportional analogies,* i.e., proportional analogies of the form: $A : A :: B : B$. As our interest is the enumeration of informative analogical clusters we simply avoid to produce this cluster.

By exchange of the means, trivial analogies are equivalent to analogies of the form $A : B :: A : B$. Enumeration of pairs of objects in a predefined sorting order trivially ensures that the difference between two objects is never computed twice. However, it does not prevent from outputting clusters that would contain only one pair of objects. This happens when two objects have a unique vector difference. This problem will be tackled in Section 4.1.

3 INFORMATIVE AND NON-REDUNDANT ENUMERATION OF ANALOGICAL CLUSTERS

3.1 Feature tree and quadratic exploration of the feature tree

Each object is represented by a vector of features. An order can be imposed on the features. In this way, each vector is considered as a list with a recursive structure of a head (the first feature value) and a tail (the remaining features). The lexicographic order, relying on the order on integers, can be applied to such a set of lists. In this way, it is possible to sort the feature vectors representing all objects.

A tree structure underlies such an ordered list. For the first feature, each different value can be encoded in one node. Each such node can be assigned the interval that represents the span over the sorted list of objects. This can be recursively applied to each interval with the tail

	[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				

	[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

Figure 3. On the left, computation of the value differences on the first level for the feature tree of Figure 2. The blank cells are not computed to avoid redundancy (opposite values) or trivial analogies (diagonal cells where intervals are reduced to one object). On the right, recursive computation of the value differences on the second level for a difference of 7 on the first level. The corresponding list of pairs of intervals is [1;2] x [4;4] + [3;3] x [5;5] (refer to table on the left). Two new lists of pairs of intervals are obtained: [1;1] x [4;4] + [3;3] x [5;5] for value 2 and [2;2] x [4;4] for value 3. The latter list of pairs of intervals is deleted as it contains only one pair of intervals, each interval reducing to one object (degenerated cluster).

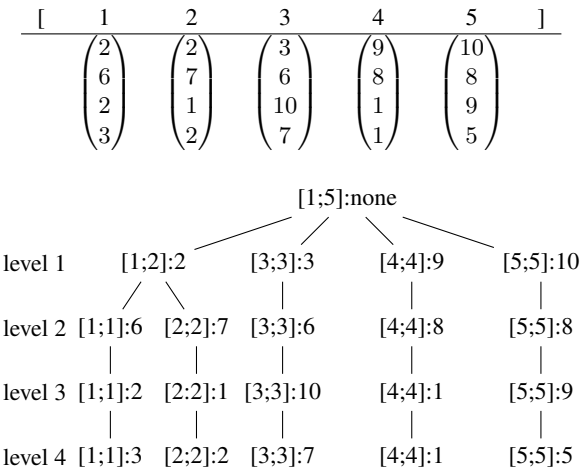


Figure 2. The feature tree (below) corresponding to a set of five objects represented as feature vectors (above). The intervals, noted with brackets, are followed by the value of the feature on that level.

of the feature vectors considered as lists (thus for the second feature and so on) to build a tree structure where the levels stand for each different feature and where each node holds the interval of the sorted objects with the same value for that feature, given all values above are equal. On the last level, each interval should be reduced to one object if the space is well separated. Such a tree structure can be traversed in breadth-first order. Figure 2 illustrates such a data structure for a set of 5 feature vectors.

This data structure[4] is quite different from the one used in [5] to search a space of strings of characters for analogies. Firstly, the geometry is different. In [5], the nodes on the same level may correspond to different characters (i.e. features). This is not the case here. Each level must correspond to exactly the same feature. Consequently, on the contrary to the structure in [5] no intermediate node can stand for an object. All the objects are to be found on the leaves. Secondly, the labels borne by the nodes are different. In our tree, the nodes bear the spanning interval in the sorted list of objects and the value of the feature; the name of the feature, being useless, is forgotten. This is of primary importance for the parallel traversal in sorted order of objects with the first interval never overtaking the second interval so as to avoid redundancy (see Section 2.3 and see below). Thirdly, our use is different as we aim at a complete enumeration of all possible ratios, which compelled the design of this data structure.

The computation of all ratios between all feature vectors simply consists in traversing the same tree in parallel in breadth-first order (a kind of a Cartesian self-product), and computing the differences between the values on each pair of nodes. For the same difference at a given local level, all the pairs of intervals are memorized in a list. This procedure is recursively applied down to the last level for each different value at a local level. Figure 3 illustrates this process for the feature tree given in Figure 2.

Sections 4.1 and 4.2 show that it is possible to terminate the exploration by checking for some structural conditions on the list of pairs of intervals memorized.

In the parallel traversal, we impose that for two lists of pairs of intervals to be processed, the first list be strictly before the second list. This is tantamount to explore only the upper corner of a matrix excluding its diagonal. This avoids redundancy and non-informativity. when computing all possible analogies: the ratio of two vectors is computed once, its opposite is not (Section 2.3); intervals that are reduced to one object on the diagonal are checked to avoid trivial clusters (Section 2.5).

3.2 Sketch of the method

The following gives a sketch of the proposed method.

- Convert each object into a feature vector;
- check for separation of space;
- define an order on the feature vectors (we use least correlations of values among features);
- sort the feature vectors according to lexicographic order in the defined order of features;
- build a feature tree for the sorted feature vectors;
- traverse the feature tree in parallel in breadth-first order to compute the differences between the feature vectors by blocks;
- output the list of pairs of intervals (on the last level, each interval should be reduced to one object if the space is well separated) that corresponds to each vector difference.

By construction and by definition, each list of pairs of objects, that share the same feature vector difference, is an analogical cluster.

[4] The tree structure described here is the same as the one used in two of our previous works: for the complete enumeration of all analogies between sentences contained in corpora of 100,000 short sentences in Chinese, Japanese and English [10] (with sequences of bits as features and various ratios for various features and automatic sorting of the features for early detection of useless zones in the cluster space so as to speed up the overall process); and for the enumeration of clusters reflecting linguistic oppositions among 40,000 short sentences in English and Japanese [11]. In these two works, respecting the equality of edit distance for analogies of commutation between strings of characters implied extra processing.

4 IMPROVEMENTS

4.1 Elimination of clusters reduced to one ratio

We call degenerated clusters those clusters which contain only one ratio, $A : B$ i.e., one pair of objects. Obviously, such clusters do not give rise to any analogy other than the trivial analogy $A : B :: A : B$ and are thus not worth to output. An early detection of such cases leads to an important reduction in processing time.

The implementation of the early detection of such degenerated clusters relies on the data structure of feature tree. After the computation of all possible differences between all possible vectors down to a certain level in the tree, it is easy to scan all the differences and look at the intervals they represent. If a set of pairs of intervals contains only one pair of intervals, each of which being reduced to one object, this is a case of a degenerated cluster. Such a cluster may be immediately deleted so as to stop any further computation on the lower levels.

A comparison of two runs of the programs with or without early detection of degenerated clusters is given in Table 1. It shows that, for our special case of structuring Sino-Japanese characters, a reduction of one third of the computational time can be achieved. There exists some overhead as is shown by the fact that an increase of 55% in computational time is observed for 1,000 characters.

Table 1. Comparison of runtimes with or without detection of degenerated clusters (clusters reduced to one ratio).

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 %

4.2 Conditional elimination of clusters reduced to one proportional analogy

In Section 2.5, it was shown that an analogy appears in only two analogical clusters. For economy of description, we would like to eliminate redundant information as most as possible. When an analogy belongs to two clusters that contain a large number of pairs of objects, it is a priori impossible, without loss of information, to remove those lines that correspond to this analogy from one of the cluster. This is not the case when one of the analogy is reduced to a cluster that contains only one analogy, i.e., exactly those two lines corresponding to the analogy at hand. This situation is illustrated below:

$$\begin{array}{l|l} A : B & \vdots \\ C : D & A : C \\ & \vdots \\ & B : D \\ & \vdots \end{array}$$

In this case, it is possible to delete the cluster reduced to one analogy. This can be performed during the enumeration of analogical clusters, level by level, using the feature tree. In this case, clusters reduced to one analogy should be memorized on each level. At the end of the exploration of each level of the feature tree, such clusters can be removed from the list of clusters to explore further. This should lead to a reduction in the total computational time. Our current implementation does make use of this possibility and performs the deletion of clusters reduced to one analogy after complete enumeration of analogical clusters in a post-processing phase.

5 EXPERIMENTS

In the frame of a larger study concerned with measuring the ease with which learners can remember Chinese characters along with their pronunciation, we are interested in studying the regularities and the correspondences between the Chinese graphical forms of characters and their pronunciation.

It is known that Chinese characters exhibit some structure and are made of graphical elements which reflect either some iconic meaning or some pronunciation. As a first step in this study, we extracted all the possible analogies between Sino-Japanese characters using a fixed-sized font. We report hereafter some of the results obtained.

5.1 The structure of Chinese characters

A large number of Chinese characters exhibit some structure concealed in their components. The most known structure consists of two elements, one being a pronunciation clue and the second one being a meaning clue, usually called semantic key. An illustration is given in Figure 4.

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

氵:亻	identical right part (pronunciation)	
泮:伴	半	PÀN
凉:倞	京	LIÀNG
冼:侁	先	ĒN

Figure 4. On the left, on each line, the characters share the same semantic key. On the right, the characters share a same pronunciation indicated by the right part.

This structure, although being quite common, is not valid for all characters. It is also believed that, because of phonetic changes, many characters that exhibited such kind of structure in ancient times cannot be interpreted in this way anymore.

In this paper, we are not interested with the relationship between graphical form and pronunciation. Our goal is limited to the extraction of the graphical structure of Chinese characters by automatic means.

5.2 Characters in monospace fonts

Monospace (or fixed-width or fixed-size) fonts are lists of characters described as black and white icons of fixed height and width. The font we use in our experiments is knj10B.bdf[5]. We use the 14,655 Sino-Japanese characters available in this font in the range between the Unicode codepoints 13,312 (一) and 40,891 (龥). Figure 5 shows a sample of these characters. As shown in this figure, the characters in this font have a fixed height of 18 lines and a fixed width of 24 pixels. The actual width used is 18 pixels, so that this font is a 18×18 pixel font, the value of 24 comes from the encoding of each line of the icons on 3 bytes that we use without change. Figure 5 visualizes the representation of three randomly selected characters from this font.

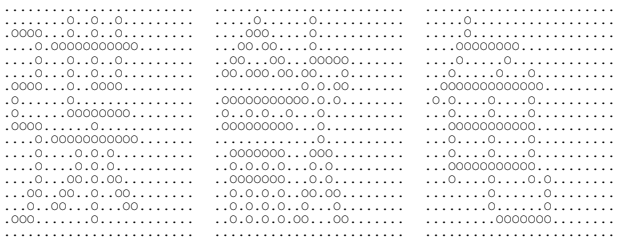


Figure 5. Visualization of some characters from the font knj10B as icons. White pixels are visualized as a dot, black pixels as a small circle.

5.3 Features used

The features we used are simply the number of black pixels on each line and each column. We use 18 lines and 18 columns; making up a sum of 36 features per characters. As an illustration, the first features for the leftmost character in Figure 5 are:

0, 3, 7, 12, 4, 4, 9, 2, 9, 5, etc.

We checked that the space of objects is completely separated by these features, *i.e.*, each vector of feature values represent only one object in the set of objects. The above 36 features discriminate each character among the 14,655 characters used in our experiments.

5.4 Analogical clusters obtained

We applied out method to extract the graphical structure of the 14,655 characters in our selected fixed-point font. The program, written in Python, needed less than 30 minutes to terminate on a machine with 4 Gb memory and equipped with an Intel Core i5 processor with a 1.7 GHz frequency.

Figure 6 shows a sample of 15 clusters output by the method. Visual inspection reveals the typical structure of characters decomposed into a left and a right part. This is in no way surprising given the features that we used and the frequency of this structure.

Figure 7 shows two examples of less usual patterns that consist in the addition of some elements in the middle of characters or a longer stroke in the central part of the characters.

Experiments performed with a lesser number of features show examples of clusters where the differentiation between the characters

[5] Font designed by Nagao Sadakazu (snagao@tkb.att.ne.jp). We use version 1.1 of 1999.

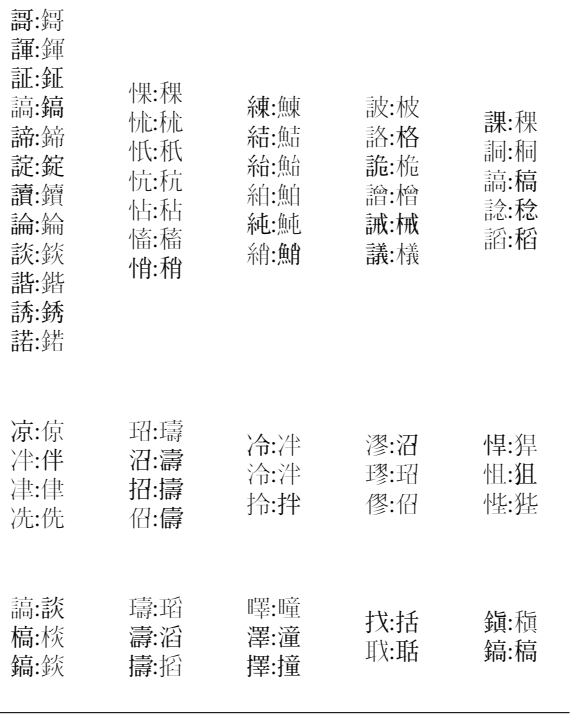


Figure 6. A sample of 15 analogical clusters output by our clustering method on 14,655 characters from the font knj10B. For the first six ones, the tenth one and the last one, both characters on the same line share the same right part (radical). The left parts (keys) are different but common to all lines. Conversely, in the other clusters the right part of the characters is the same in each column of the cluster. These clusters show commutations of various keys (four, three or two only) with only two different radicals.

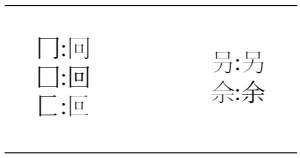


Figure 7. Two examples of less typical cases of analogical clusters output by our clustering method. The first one on the left shows the insertion of a square in the middle of the character. The second one captures a longer stroke in the center of the characters on the right.

Figure 8. A cluster output by our clustering method using only the number of horizontal black pixels as features. Taking into account the columns eliminates this analogy.

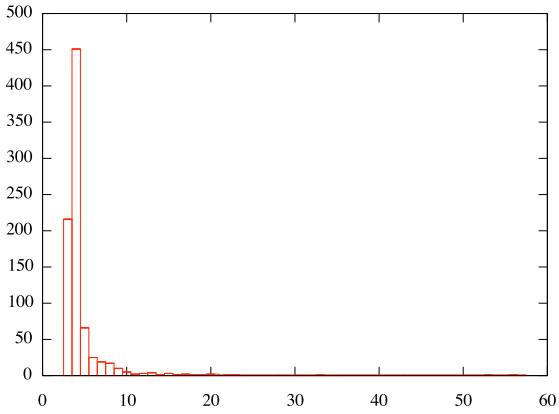


Figure 9. Distribution of clusters by number of pairs. In abscissae, number of pairs in the clusters. In ordinates, number of clusters with the same number of pairs. The largest cluster contains 55 pairs. There are only 16 clusters between 13 and 55 on the horizontal axis.

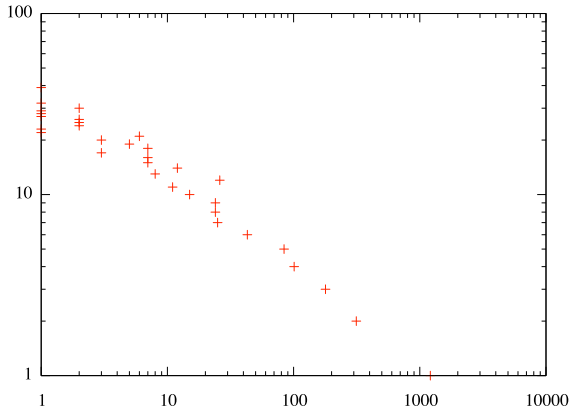


Figure 10. Distribution of characters by number of clusters they appear in. In abscissae, number of characters that appear in the same number of clusters. In ordinates, number of clusters. Logarithmic scales.

is not sufficient. An example of this is given in Figure 8. In this example, only the number of black pixels on the lines have been used (18 features). In this case, the vertical directions of the strokes in the left characters on the first and second lines have not been distinguished so that the method concluded to a proportional analogy that may be questionable (or not for reasons of equivalence between various forms of writing).

The distribution of clusters by number of pairs of objects (36 features) is plotted on Figure 9. This distribution exhibits a long tail. Few clusters are very large while short clusters are more numerous The fact that the number of clusters with 3 pairs of characters (451) is greater than the number of clusters with only two pairs of objects (216) is explained by the elimination of redundant clusters reduced to one analogy. The largest cluster contains 55 pairs of characters.

5.5 Number of clusters per character

The total number of characters that appear in all analogical clusters was 5,982. This represents 41% of the total number of characters used (14,655). We *a priori* expected a higher number.

We also measured the number of (non-redundant) clusters each character appears in. Figure 10 plots of the distribution of characters per number of clusters they appear in. The use of logarithmic scales suggests a Zipfian distribution that needs more enquiry. This measure gives an estimation of the complexity of a character by the number of oppositions it has with all other characters. This reflects its degree of freedom in the overall graphical system. A character which does not appear in any cluster is somehow free relatively to the overall system. In vision of our future experiments, we hypothetize that characters that appear in more clusters may be easier to remember if the learner has access to the global view given by the clusters. In addition, of course, the number of strokes should be taken into consideration.

6 RELATED WORKS AND CONCLUSION

This paper presented a method to automatically extract all possible proportional analogies from a given set of objects represented as feature vectors. In previous works, we showed how to do this for short sentences [10, 11] but extra computation was required to check for the edit distance constraint necessary in proportional analogies of commutation between strings of symbols[6].

Relying on specific properties of our formalization of objects as feature vectors, we defined the ratio between objects as a difference between vectors, and conformity as equality between vectors. This particular setting allowed us to reformulate the problem, which has a complexity of $O(n^4)$, in an equivalent problem with a quadratic complexity, that of enumerating analogical clusters, i.e., lists of pairs of objects with the same ratio.

We proposed an adequate data structure this problem and, by further exploiting the properties of proportional analogies, we showed how to avoid redundancy in the enumeration and non-informative clusters. With all this, we showed that the problem becomes tractable as as to solve our problem at hand: extracting all analogies between Sino-Japanese characters in their graphical form.

Although the iconicity of proportional analogies has already been stressed in a broader sense of the word [3, 4], this is the first attempt at solving analogies between icons of black and white pixels using their graphical form directly. This problem had already been mentioned in [8] but without a solution. Our formalization and application to Sino-Japanese characters shows that an explicit description of characters in terms of their constituents (keys or radicals) as is proposed in [20] can be avoided.

Our proposed method does not exhaust the subject of proportional analogies between icons made of black and white pixels. There remains a number of problems. One problem is the necessary fixed size of the icons to compare, hence our use of monospace fixed-size fonts. Firstly, any shift of a character by one or several lines (or columns) would disrupt the analogical relations that are made possible to compute with our feature vectors because almost all characters are well lined up with the first line and column. Secondly, analogical relations between characters of different sizes cannot obviously be captured with the method proposed here.

The work presented here is a preprocessing step in a larger study

[6] [6] is the first mention of the edit distance constraint in terms of similarities; [7] gives the equivalent expression with edit distances; [9] is the published form of the proceedings in which [7] appeared, with few years delay. The edit distance constraint is necessary between strings of symbols to avoid too many spurious analogies that would be formed without it. Experiments with several hundreds of thousands of short sentences in Chinese collected from the Web confirm this point.

of proportional analogies of graphical form and pronunciation among Chinese or Sino-Japanese characters. We perform the same kind of analogical clustering on the level of pronunciation and compute the intersection between analogical clusters on the graphical and on the pronunciation levels. We hypothesize that knowing analogical correspondences between the graphical and pronunciation levels of Chinese or Sino-Japanese characters would ease their memorization by learners. We intend to test this hypothesis with subjects.

REFERENCES

[1] Dedre Gentner, 'Structure mapping: A theoretical model for analogy', *Cognitive Science*, **7**(2), 155–170, (1983).

[2] Robert R. Hoffman, 'Monster analogies', *AI Magazine*, **11**, 11–35, (1995).

[3] Esa Itkonen, 'Iconicity, analogy, and universal grammar', *Journal of Pragmatics*, **22**(1), 37–53, (1994).

[4] Esa Itkonen, 'Analogy as structure and process: Approaches in linguistics, cognitive psychology and philosophy of science', in *Human cogntive processing*, eds., Marcelo Dascal, Raymond W. Gibbs, and Jan Nuyts, volume 14, 250 p., John Benjamins Publishing Company, Amsterdam / Philadelphia, (2005).

[5] Philippe Langlais and François Yvon, 'Scaling up analogical learning', in *Coling 2008: Companion volume: Posters*, pp. 51–54, Manchester, UK, (August 2008). Coling 2008 Organizing Committee.

[6] Yves Lepage, 'Languages of analogical strings', in *Proceedings of COLING-2000*, volume 1, pp. 488–494, Saarbrücken, (July-August 2000).

[7] Yves Lepage, 'Analogy and formal languages', in *Proceedings of FG/MOL 2001*, pp. 1–12, Helsinki, (August 2001).

[8] Yves Lepage, *Of that kind of analogies capturing linguistic commutations (in French)*, Habilitation thesis, Joseph Fourier Grenoble University, May 2003.

[9] Yves Lepage, 'Analogy and formal languages', *Electronic notes in theoretical computer science*, **53**, 180–191, (April 2004).

[10] Yves Lepage, 'Lower and higher estimates of the number of "true analogies" between sentences contained in a large multilingual corpus', in *Proceedings of COLING-2004*, volume 1, pp. 736–742, Genève, (August 2004).

[11] Yves Lepage and Chooi-ling Goh, 'Towards automatic acquisition of linguistic features', in *Proceedings of the 17th Nordic Conference on Computational Linguistics (NODALIDA 2009)*, eds., Kristiina Jokinen and Eckard Bick, pp. 118–125, Odense, (May 2009).

[12] Yves Lepage, Julien Migeot, and Erwan Guillerm, 'A corpus study on the number of true proportional analogies between chunks in two typologically different languages', in *Proceedings of the seventh international Symposium on Natural Language Processing (SNLP 2007)*, pp. 117–122, Pattaya, Thailand, (December 2007). Kasetsart University, ISBN 978-974-623-062-9.

[13] Yves Lepage, Julien Migeot, and Erwan Guillerm, 'A measure of the number of true analogies between chunks in Japanese', *Lecture Notes in Artificial Intelligence*, **5603**, 154–164, (2009).

[14] Hermann Paul, *Prinzipien der Sprachgeschichte*, Niemayer, Tübingen, 1920.

[15] Peter Turney, 'A uniform approach to analogies, synonyms, antonyms, and associations', in *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pp. 905–912, Manchester, UK, (August 2008). Coling 2008 Organizing Committee.

[16] Peter D. Turney, 'Similarity of semantic relations', *Computational Linguistics*, **32**(2), 379–416, (2006).

[17] Peter D. Turney and Michael L. Littman, 'Corpus-based learning of analogies and semantic relations', *Machine Learning*, **60**(1–3), 251–278, (2005).

[18] Marcus Terentius Varro, *De lingua latina*, Coll. Belles-lettres, Paris, 1954. Trad. J. Collart.

[19] Tony Veale and Shanshan Chen, 'Learning to extract semantic content from the orthographic structure of Chinese words', *proceedings of the 17th Irish conference on Artificial Intelligence and Cognitive Science (AICS2006)*, (sept 2006).

[20] Lars Yencken and Timothy Baldwin, 'Measuring and predicting orthographic associations: Modelling the similarity of Japanese kanji', in *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pp. 1041–1048, Manchester, UK, (August 2008). Coling 2008 Organizing Committee.

[21] François Yvon, Nicolas Stroppa, Laurent Miclet, and Arnaud Delhay, 'Solving analogical equations on words', Rapport technique ENST2004D005, ENST, (July 2004).