Semantic Network Closure Structures in Dual Translation of Stochastic Languages

Lukáš Pichl

International Christian University
Osawa 3-10-2, Mitaka, Tokyo, 181-8585, Japan
lukas@icu.ac.jp
http://www.icu.ac.jp/

Abstract. Iterated semantic translation as reflected in dual space of two different languages provides - through a normalized reference set of dictionary expressed vocabulary that is inclusive of double sided link direction correspondence - a mechanism for assessment of word extent breadth and meaning delineation. This measure resolves according to the number of source language passages in round translating steps, until content exhaustion. Semantic clusters emerge in above procedural format through graph decomposition of the united word set above language pair to isolated segments of connected semantic vertices. Such a linkage is also present in single language morphological context of thesaurus format with enumerated word content proximity, typically farther subjected to additional support that arises from accompanying clauses, applied level of grammatical compatibility, and overall linguistic usage purity. Resorting to algorithmic reverberation and numerical case studies on stochastic data samples, it is argued that resemblance of contained and mirrored semantic vertex structures follows cultural interaction, although shadowed at various degrees of virtualization, in evolving linguistic relation that ultimately leads to a knowledge cohesion level frame of scientific depth. Since it has been known that complexity information depth as content purity grade paraphrases itself in Kolmogorov efficiency sense, either to stability measure of fixed structures or manifests as a catalytic growth factor in nurturing substances, this work provides a particular instantiation in topological context of semantic graphs.

1 Introduction

Content information measured in terms of the shortest available program to encode a given dataset is known to stochastically equate the system entropy value as expressed by means of the compositional probability function in constructive proof of such statement [1]. Application context of computing machinery, library science and language standardization provides interesting filter frames in practical instantiation so as to encompass incompleteness and organizational restrains for optimal content reconstruction.

To assess a vocabulary content in its entirety, a complete composition scheme in knowledge space is required from alphabet specification via meaning dimensions

S. Kikuchi, S. Sachdeva, and S. Bhalla (Eds.): DNIS 2010, LNCS 5999, pp. 210–224, 2010.

and notion typing stencils, both physical and abstracted, towards emergence of knowledge accumulation formalism. Such a process may start from replication of a single element (existence) to enumerated finite set of alphabet (replication and separation) and syllable formation, leaving aside a certain minimal subset with economic functions (determination, preposition), and continues further to morpheme formation, thereby usually subjected to notational combinatorial complexity, and consequently adjusted instead to fundamentals of actual semantic processing architecture, such as typical thinking and categorizing patterns native to a processing brain (action, emotion, subject), in a process gradually forcing outward and inward views to postulated coincidence patterns, which manifests as a reflection of the stabilizing processing organization onto semantic output fixation (function selector catalogues for instance).

In terms of self composition, nature selected and seed encoded complex structuring processes in plant growth mechanisms, while human civilization increasingly accumulates instances of such external knowledge for abstracted utilization elsewhere, a mechanism used in education that itself may alter or produce vocabulary items to above effect. This work concerns with a word level snapshot of language in two frequented semantic tool frames, first a thesaurus, then a dictionary, both reduced to word correspondence grains with logical, integer, and fractional weight scales adopted for numerical analysis. Such approach establishes a transform from construction parameters of probabilistic word data samples to the semantic cluster appearance. A reference numerical text corpus is then composed using derived cluster probability density of words, and subjected to metalevel analysis, namely text abstraction first by means of sentence extraction with local and global similarity coefficients, and then using absolute hierarchical ranking, in order to address representative features that arise from finite and restrained vocabularies, and apply to natural language datasets. Since the field of semantic processing is long established [2,3], this work focuses on purely numerical capture aspects that recently regained attention [4].

The article is organized as follows. After a section dedicated to graph formalism of numerical semantic circuits, construction of derived text documents is given along with the examples of related substance extraction methods. Replication of computational formalism is then recognized as a quantifiable cultural personalization mechanism, which is accompanied with actual natural language processing illustrations to conclude this text with a brief summary.

2 Numerical Semantic Circuits

Since all computational text processing methods reduce content manipulation to number processing at certain depth of representation layer (program, software, compilation, middleware, instruction set) an enumerated semantic dataset is denoted as a graph collection of vertices,

$$V = \{v_i\}_{i=1}^{n_V}\,, \tag{1}$$

where domain meaning relations expressed in vertex annotation patterns are given in form of multiple connection edges,

$$E = \{\, e(v_i; \tau_i, v_j; \tau_j) | \, \forall v_i,\, v_j \in V \wedge \tau_i \in T \} \,, \tag{2}$$

where T is a set of tags (synonym domain, antonym domain, sets of cultural links, economic links, object resource links, processing links, or different languages). Edges are naturally represented as logical weights, integer weights, and rational weights,

$$\begin{aligned} e(;\,,;\,) &= \{0 \vee 1\} \\ e(;\,,;\,) &= \{\infty \vee p \vee p-1 \vee \ldots \vee 1\} \\ e(;\,,;\,) &= \{\infty \vee 1 \vee 1/2 \vee \ldots \vee 1/q\} \end{aligned} \tag{3}$$

so as to map elementary arithmetic thinking patterns. Numerical labels in Eq. (3) are sorted from edge absence value to the highest proximity. Connection circuits can be enabled with $\tau_i \neq \tau_j$ thus forming an overlay multigraph structure. In what follows edges are directed ($e(i;\,,j;\,)$ is not necessarily symmetric) and appear as a result of serial probabilistic generative process that can be conditioned with current graph snapshot topology.

Derived semantic evaluation criteria of practical interest include graph transitive closure with number and sizes of extracted strongly connected segments. Weighed graphs subresolve transitive closure to the shortest path version characterized with segment wide square pattern of minimal connection cost. Transitive closure is also a suitable aggregation level to derive and utilize complement graphs for semantic closure.

It is noticed here that optimized subgraph vertex skeleton layout, from which each word locates at most l-edges away, can provide a core vocabulary set for self-learner extension, whereas determination of all complete subgraphs (prior to transitive closure, and furthermore in comparison) shows aggregation depth in original dataset. Both approaches are nondeterministic polynomial complete problems even if tractable with low size of content nucleation typical in natural languages, and as such are not pursued here. Similarly, sparse nature of data content allows us to leave aside graph theoretic applications of vertex and edge set reconstruction paths as frame transforms of access compression and topological sort. Instead, multigraph structure of vertex colorization is applied using tags $\tau_i \in T$ for self-consistent frame-compatible content enhancement.

2.1 Thesaurus Format

Enumerated single layer word graph structure as given above (source and reverse coincide, $\tau_1 = \tau_2 = \tau$, is initialized by expanding list of source vertices one by one while assigning a n end vertex thus forming a thesaurus edge. Structure formation then continues according to three generative procedures,

1. uniform method selects a pair of vertices at no bias,
2. shaped method is binomially modulated (parameter p),
3. cluster method adopts current vertex degree statistics,

with vertex selection probability functions given as

$$\pi(v_k) = \begin{cases} \frac{1}{n_v} & \text{uniform} \\ {}_nC_k p^k (1-p)^{n-k} & \text{shaped} \\ \frac{\deg(v_k)}{\sum_{l=1}^{n_V} \deg(v_L)} & \text{cluster} \end{cases} \qquad k = 1, \ldots, n_V, \tag{4}$$

where indegree is selected for reverse end, outdegree is selected for source end, and denominator of the last fraction stands for the current number of graph edges. Weighted graphs, in addition, adopt in either case label value of $d+1$ or $1/(d+1)$, where d is the outdegree preceding current edge addition.

Symmetric version of the above algorithm automatically includes opposite directed edge to the graph at every generation step. In either case, data structure is represented in adjacency matrix $A[;\tau_i][;\tau_j]$ (to be resolved into τ pair instances in what follows).

The structure generation process grows above semantic circuits to edge concentration levels at fixed inverse proportion to graph size (constant average outdegree as common for natural language dataset) or higher, up to a numerical fabric with connection density $c \in (0,1)$ that mirrors parallel destruction of semantics on behalf of intense vertex feature generalization that turns to bypass specialization importance.

2.2 Dictionary Format

Vertex side duplication in thesaurus procedure provides dictionary structure as a naturally derived extension, in which both vertex lists are link initialized (category format inclusion operator) and extended (structure builder) separately. Denoting total graph size as $2n_v$, there are following possibilities ($T = \{\tau_1, \tau_2\}$)

1. pure dictionary ($e(;\tau_1,;\tau_2) = 0 = e(;\tau_2,;\tau_1)$)
2. semicomplete base ($e(;\tau_i,;\tau_{3-i}) = 0$ for $i = 1 \vee i = 2$)
3. complete base (full dictionary and thesaurus pair).

To compare thesaurus and pure dictionary structures, the size of the latter doubles graph size of the former. Uniformly structured dictionary (on both sides, $\tau_i \rightarrow \tau_{3-i}$) combined with uniformly structured thesauri (source and reverse) scales a single thesaurus format from n_V to $2n_V$ as expected, to which mixed structures provide suitable means for semantic layout comparison.

2.3 Content Separation

Strongly connected graph segments correspond to semantic domain clustering as contained between original graph and its transitive closure version, and are conditioned by the path length of the shortest indirect connection (twice the number of translation rounds in case of pure dictionary closure). Whenever thesaurus and dictionary are combined, transitive closure of thesaurus should precede the phase that allows language translation, at least in classical linguistic way of frame categorization.

The algorithm for transitive closure in compact form reads [5].

```
procedure Floyd_Warshall(A[1:n][1:n])
  extern binop1, binop2
  C[][]:=A
  for k:=1 to n /*get links through 1 to k*/
    for j:=1 to n
      for l:=1 to n
        C[j][l]=binop1(C[j][l],binop2(C[j][k],C[k][l]))
  return C
end ! transitive closure and shortest path
```

Here the two binary operators are{ $\vee$, $\wedge$} on logical graphs and {min, +} on weighed graphs. To determine strongly connected components (the transform C[i][j]=((C[i][j]==∞)?0:1) is used for weighted graphs) it suffices (in cubic time complexity) to perform a straightforward postprocessing analysis below.

```
function Strong_Connected_Components(C[1:n][1:n])
  inlabel[1:n]=1, c=0
  for i=1 to n
    if (inlabel[i])
      print(line end, i), c=c+1
      for j=1 to n
        if (C[i][j] .and. C[j][i] .and. inlabel[j])
          print(space,j), inlabel[j]=0
  return c
end ! a posteriori analysis to Floyd_Warshall
```

If only strong connected graph components are needed (interlink statistics between full blocks is omitted), graph search is known to provide the required result in quadratic time complexity [6].

```
! uses stack of graph node labels
visited[1:n]=0

procedure search(i,first) ! recursive
  visited[i]=1;
  for j=1 to n
    if(C[i][j] .and. !visited[j])
      visited[j]=1, search(j,first)
      if(first==1) push(j)
end ! direction conditioned depth first search

procedure Kosaraju(c)
  for i=1 to n
   if(!visited[i]) search(i,1)
  for i,j=1 to n C[i][j]<->C[j][i] ! graph transpose
  visited[1:n]=0
  while(!stack_empty())
```

```
   i=pop(), print(line end,i), c=c+1 ! new component
   search(i,0) ! reversed traverse
   for j=1 to stack_size()
     if(visited[k=pop()]) print(space, k)
     else (push(k), break)
end ! strongly connected components
```

One sided (source or reverse) connected components rooted at a particular source word v_i are obtained with breadth first algorithm that is merely augmented with the tree depth labels along the node link extraction process.

```
! uses a first in first out queue of graph labels
depth[1:n]=-1

procedure translation_set(i,A[1:n][1:n])
! matrix A is (tau_1-> tau_2 and tau_2->tau_1) projected
  put(i), depth[i]=0
  while(queue_empty==0)
    j=get(), level=depth[j]
    for k=1 to n
      if (A[j][k]!=0 .and. A[j][k]<infty .and. depth[k]==-1)
        depth[k]=level+1, put(k)
end ! tree-depth-tracked breadth first search
```

When the above search procedure of translation set is depth conditioned in addition with each vertex inclusion to the strongly connected component of the root node v_i, the term

$$\sigma(i : \tau_i, j; \tau_i) = |\mathrm{depth}[i] - \mathrm{depth}[j]|/2 \tag{5}$$

gives the minimal number of translation passes via language τ_j to connect words i and j in language τ_i.

The above components provide notational and procedural basis for numerical patterns analyzed below.

2.4 Stochastic Patterns

Figure 1 shows an inset view of two graphs, one composed with uniformly based binomially-augmented fixed vertex selection, and one augmented with a scale free growth procedure emphasizing the vertex indegree when determining endpoint of a new oriented edge. All graphs are initialized in the semantic frame proper sense, meaning each vertex indegree and outdegree is at least one. The extra average vertex valence is set to five in all computations with the exception of dense numerical image illustration of Fig. 1 ($N_m = 50$). Figure 2 gives an account of indegree and outdegree statistics resulting from the above outlined numerical circuit prototype composition, including the indegree power law behavior of the scale free component.

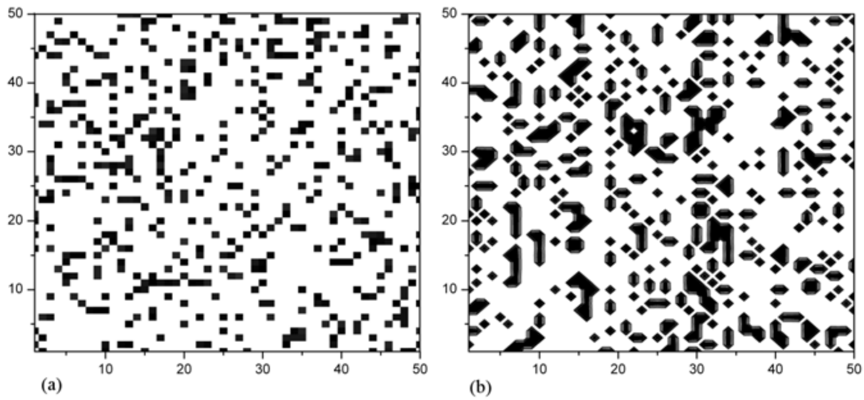

Fig. 1. Connection weight pattern in a semantic subgraph with (a) binomial and (b) clustering vertex selection methods

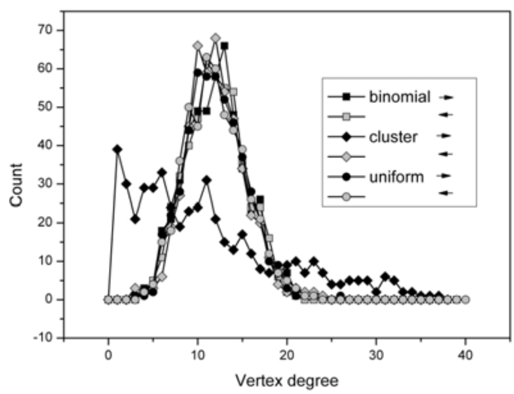

Fig. 2. Layout of vertex indegree and outdegree population for vertex selection methods explained in the text

Figure 3 gives vertex changing frequencies for the shortest paths among all vertex pairs obtained from the breadth first search generated traverse trees. Since in order to utilize the number of strongly connected components a substantial degree of frame incompleteness is needed, such a particular illustration is deferred and can instead be found in statistical work on natural language processing. Thus the current subgraph connection patterns, vertex linkage lengths, and histogram component shapes provide necessary elements for document processing and meaning depth inference.

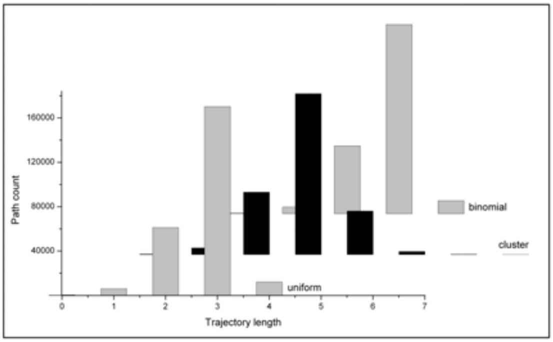

Fig. 3. Histogram of shortest path edge numbers for vertex selection methods explained in the text

3 Numerical Circuit Documents

Prototyped semantic layout of vocabulary circuitry generated above practically links to a document content as the induced meaning range, whether in full or additionally restrained, for instance up to a certain depth or an edge weight cut off.

Naturally languages for human communication are complicated due to personalization of prototypes (incompatible layouts) or an incomplete personal acquisition depth pattern of the same prototype standard (varying education depth). A communication problem in the latter situation is how to assess author depth of meaning intent, while the former circumstance may also include multigraph editing preprocessing to match conventional semantic circuit etalons. Exclusion of author intended meaning depth assessment through emphasis on recipients effect or a personal semantic utility naturally leads to acknowledgement maximization of received document text patterns.

Elementary document composition rules are outlined next and processed for a numerical provision of illustrative examples. The number of clauses is fixed as m_d; each is long from 1 to n_c according to a binomial shape of probability density with $p = 0.5$.

3.1 Markov Composition

Defining a current state as the word at the end of a clause fragment, the probability of the next word can be taken as

$$m_{pq} = \frac{M_{pq}}{\sum_q M_{pq}}. \tag{6}$$

Without introducing further composition parameters, M_{pq} may stand either for graph matrix or its complement (computed for weighted graphs as a reversal $w_i \leftrightarrow w_{|w|-i+1}$ in the set of predefined instantiated values). Since either case is a path traverse on matrix M that is rather sparse in the original semantic domain, complement matrix corresponds to a syntax-adjusted equally randomized text population, whereas the original graph matrix would employ semantic shifts that may cycle (or freeze if isolated vertices were allowed). Therefore the matrix type is altered with probability ratio q set as 0.5 unless mentioned otherwise. Sentence opening words in each clause are taken at random.

Although transitive closure matrices would provide for higher delineated transitions among strongly connected components, out of which instances could then be taken at random or in above manner for more realistic semantic content nucleation, the procedure in Eq. (6) suffices to provide an unbiased probabilistic test document corpus that serves as a suitable comparative reference to the following method.

3.2 Scale Free Assembly

Dynamic collaborative systems such as communication providers or certain types of metabolic networks are known to exhibit hub structures and overal organization that is called scale free when the vertex degree density follows a power law with negative exponent [7]. Such a complexity pattern may appear in text documents and is simulated as follows.

An empty dynamic list to contain separate document included words and their frequencies is initialized. Another static list is made for vocabulary and initialized with vertex outdegrees. New clause members are selected from either list with probability ratio q and $1 - q$. All word inclusion operations are accompanied with document word list update and frequency increment.

3.3 Substance Extraction

Document content compression to core information is usually performed by means of abstract generation. Provided the actual vertex choice pattern in a clause largely includes grammar effects (whether after accurate morpheme annotation with grammatical T tags in a clause, or directly adopting the sinogram system that is complete as a sequence of graphical morpheme characters), the abstract can be generated by means of clause extraction (and a posteriori indirect grammar pattern restoration, if any is required). In such a representation, substance extraction reduces to a clause selection algorithm. To this aim, first the similarity clustering method of [10] is reviewed (as a document relative tool) and then compared to clause ranking based on scores derived from the semantic prototype (as its own knowledge-base-rooted absolute tool that can emphasize any depth of semantic extent structure).

Content Similarity Clustering. Document substance extraction based on similarity of clauses is an effective clustering method that selects new cluster parents as well as candidates for seed merger by using the percentage of shared component vertices. The terminology is as follows [10].

Global similarity coefficient of two clauses C_i and C_j reads

$$\mathrm{GSC}(C_i, C_j) = \frac{2 \times n(\text{common words of } C_i \text{ and } C_j)}{n(C_i) + n(C_j)} \quad \forall i \neq j \leq l \tag{7}$$

Local similarity coefficient of a clause C_i and a cluster C reads in the same way

$$\mathrm{LSC}(C_i, C) = \frac{2 \times n(\text{common words of } C_i \text{ and } C)}{n(C_i) + n(C)} \quad \forall i \leq l. \tag{8}$$

The method starts with a set of word lists and computes their pairwise global similarity coefficient at first.

The pair with the highest GSC value, provided it is above a threshold level τ, is replaced with a new global cluster clause that enlists only the common words. Then the local similarity coefficient is computed for all other lists, and if the value is above τ, such a list is deleted at the end of round while the cluster list updates to commonly shared words. The procedure then recurs.

Each cluster is annotated with a label to their single original clause placed high at source document. As the cluster clauses diminish through common word mergers in the length of representation words, similarity coefficients eventually decrease below the parameter τ. Clusters are ranked with the clause count in which order their labels enter the abstract text. The label with a higher position in the original document is retained in symmetric mergers.

Vocabulary Template Scoring. Provided a given semantic prototype, here expressed in its multigraph structure, the absolute vertex meaning relevance can be represented as word outdegree (or negative cumulative weight of outgoing edges in weighted version), and the relevance of entire clause then assessed as the total of such values, rescaled to the average per contained vertex, so as to obtain a scoring method independent of the numerical clause length,

$$S(C) = \frac{1}{l} \sum_{i=1}^{l} w(C_i) \quad C = \{C_i\}_{i=1}^{l}, \quad w = w(V, E, v_i(c_i)). \tag{9}$$

Natural variations motivated above include scoring with sizes of connected trees or strongly connected components, and, importantly, application of complement graphs.

A word scoring method based on original graph emphasizes commonality of meaning whereas the one derived from complement version accentuates the content specialization degree.

3.4 Stochastic Patterns

Above enumerated semantic composition structure in its procedural closure transforms a particular set of probability density patterns (stochastic operators) forward from the level of prototype assembly to the level of expanded document set construction, and backward at the level of substance extraction for abstract composition. Figure 4 provides numerical coordinates for an illustrative run of one hundred documents that have word vertices either selected in random path attachment with interlaced straight connection matrices of the original and complement graphs (a), or in a hub scaling growth process (b), for each of the three sets of previously outlined semantic prototypes. The number of clauses in a document, abstract, and the maximal word count parameters are selected as 20, 5, and 50, respectively. The abscissa values represent the ratio of clauses nucleated as clusters into the resulting abstract substance, and consequently are rather sparse, which allows to apply a small horizontal shift among all three datasets to avoid their overlap. The ordinate values use vertex outdegree to weight the relevance domain of each word, at the levels of the abstract and the entire document. It is confirmed that the scale free procedure of document content composition makes the similarity ranking based method of abstract extraction more effective. Since furthermore bottom semantic circuit structures imprint onto machine extracted abstracts differently, numerical spaces of extraction method conditioned efficiencies could be used in complexity assessment, genre categorization, and other linguistic applications.

It may be worthwhile noting that representation of Fig. 4, possibly with more superposed evaluation dimensions, naturally brings material science issues of point concentration, geometric separation, and focus location into a dynamic picture of document evolution and message elaboration, for instance as it is driven and represented in the genetic programming formalism for novel applications.

4 Replication Layer Saturation

Since above graph notation is optimally generic, it is worthwhile to review its application domain in linguistics. Thesaurus is a language tool that can be termed self explanatory. As it certainly appears in various editions of what is widely considered the same standard, this fact implies equivalent numerical representation, excepting perhaps context emphasizing depth and content volume effect. Such equivalence relation can be studied for practical purpose using strongly connected component analysis as a tool for instance. Online encyclopedias provide another illustration although at a higher level of topical documents.

Partial instantiation schemes of the same graph semantic structure may correspond to personal learning staging. If errors are admitted, interesting educational tasks emerge on how as to infer them indirectly from documents and what fixation mechanisms are effective for what purpose even in the absence or scarcity of error signaling documents supplied for evaluation.

It is understandable that education and personal experience in a given environmental setting solidify with certain priors, that are learnt to flex in part when a different language experience is met. A saturation problem then arises

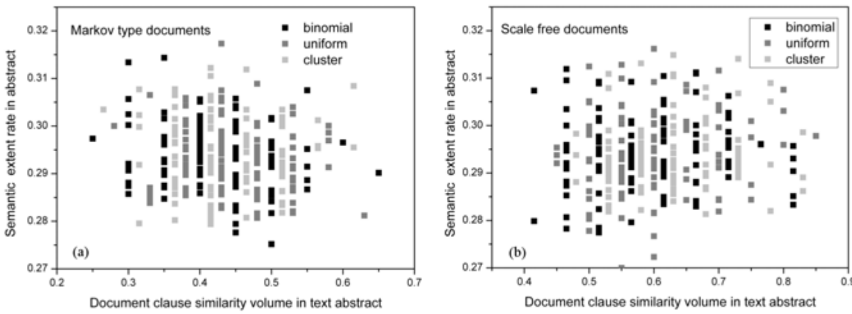

Fig. 4. Numerical documents as positioned by summary efficiency ratio pairs for (a) fixed transition and (b) current state preference text growth methods

for each personal semantic fixation skeleton on as to what other semantic graph structures have to be met before all acquired semantic path obstacles can be overcome. Similarly language separation measures in a knowledge space including innovative potential of new members could be hypothesized using linked vertex structures, which applications are though not in scope of the present static frame based article.

The field of numerical semantics extends to a number of other computer science applications of practical interest [8,9].

5 Natural Language Excurse

This part provides application software examples for above notions in Asian languages. In particular, Korean-Japanese-Chinese language family is known to instantiate semantic patterns in a scientific manner from the processing operator emphasized form to the knowledge system empowered formalism.

On practical footing, unicode's CJK encoding long ago provided the practical standardization platform to numerical text processing that was instantiated in two Java virtual machine programming endeavors to natural language processing that are referred hereafter.

5.1 Korean Syllabi Cycles

The applet environment shown in Fig. 5 implemented a Korean vocabulary set with its Japanese dictionary counterpart, and illustrates translation routes among both languages as a part of a broader study beyond scope of this article [11].

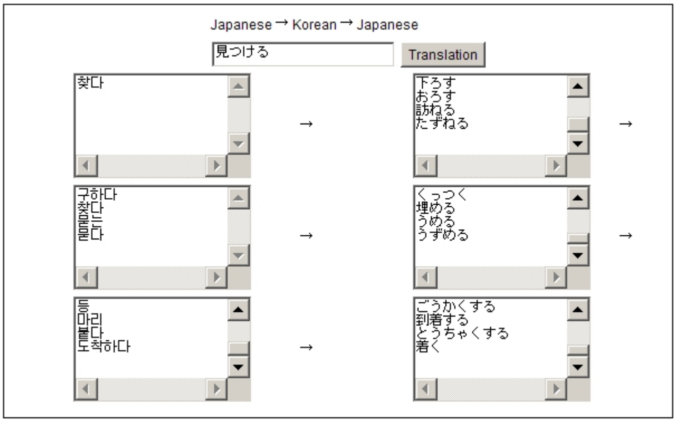

Fig. 5. Java applet for dictionary closure loops between Japanese and Korean languages based on a graph search

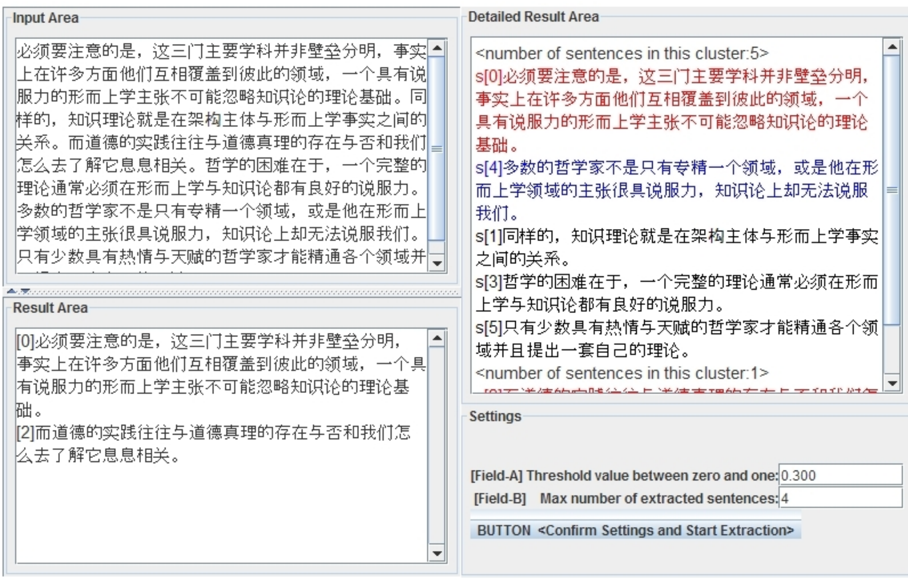

Fig. 6. Java applet for abstract summarization of short sinogram text based on the similarity clustering method

5.2 Sinograph Characters

The applet environment shown in Fig. 6 implemented a Chinese character set with the single language prototype left to readers, and illustrates similarity coefficient based abstract extraction as a part of a broader study beyond scope of this article [12].

6 Concluding Remarks

Semantic translation closure was formulated using graph theory formalism and instantiated in a metaphoric frame that numerically impersonalizes points of view and thinking patterns as those culturally represented in closed thesaurus and open dictionary knowledge structures. Apart from reinstating enumeration nature of language processing in the machine framework that provides for fundamental stochastic transforms on knowledge probability spaces, significance of direct semantic prototype linkage to document summarization and substance extraction algorithms was found, namely one that can be reinforced as terminological processing coincidence at notation and organization levels. Numerical semantic circuit frame assembled in the present article has been accompanied with recited actual natural language applications to illustrate structural interface in either frame. Completion effort in sense of pure efficiency guides computerized categorization efforts to incorporation of nature structures as imprinted onto material organization in mosaic decomposition of knowledge. Confluent structures of dictionary mediated knowledge culture interface outlined here may numerically illustrate an old proverb the more languages you know, the more times a person you are, perhaps most significantly in view of irreducible knowledge components.

References

1. Shiryaev, A. (ed.): Selected Works of Andrey Kolmogorov: Information Theory and Theory of Algorithms. Springer, Heidelberg (1993)
2. Minsky, M. (ed.): Semantic Information Processing. MIT Press, Cambridge (1968)
3. Sowa, J.F. (ed.): Principles of Semantic Networks: Explorations in the Representation of Knowledge. Morgan Kaufmann Publishers, San Francisco (1991)
4. Theoharis, Y., Tzitzikas, Y., Kotzinos, D., Christophides, V.: On Graph Features of Semantic Web Schemas. IEEE Transactions on Knowledge and Data Engineering 20(5), 692 (2008)
5. Floyd, R.: Algorithm 97: Shortest Path. Communications of the ACM 5(6), 345 (1962); Warshall, S.: A theorem on Boolean matrices. Journal of the ACM 9(1), 11 (1962)
6. Aho, A., Hopcroft, J., Ullman, J.: Data Structures and Algorithms. Addison-Wesley, Reading (1983)
7. Barabasi, A.-L., Albert, R.: Emergence of scaling in random networks. Science 286, 509 (1999)

8. Raghavan, S., Rohana, R., Leon, D., Podgurski, A., Augustine, V.: Dex: A semantic-graph differencing tool for studying changes in large code bases. In: Proceedings of the 20th IEEE International Conference on Software Maintenance, pp. 188–197. IEEE Computer Society, Los Alamitos (2004)
9. Ramamritham, K., Chrysanthis, P.: Semantics-based Concurrency Control. In: Encyclopedia of Database Systems 2009, p. 2591 (2009)
10. Kumar, A., Kumar, P., Rao, S., Reddy, K.: An Improved Approach to Extract Document Summaries Based on Popularity. In: Bhalla, S. (ed.) DNIS 2005. LNCS, vol. 3433, pp. 310–318. Springer, Heidelberg (2005)
11. Takagi, M.: Translation Routes and Implicit Language Structure of Japanese and Korean, thesis work, International Christian University Library (2008)
12. Mikami, N.: Popularity-based Approach to Automatic Extraction of Abstracts from Chinese Text Documents, thesis work, International Christian University Library (2006)