Knowledge-Based Visualization of Textual Information Applied in Biomedical Text Mining

Joseph Leone
Dept. of Computer Science and Engineering
University of Connecticut
Storrs, CT 06269-3155 USA
Joseph.2.Leone@uconn.edu

Dong-Guk Shin
Dept. of Computer Science and Engineering
University of Connecticut
Storrs, CT 06269-3155 USA
shin@engr.uconn.edu

Abstract—**This paper describes a system, called VisuText, which creates visualized diagrams from textual descriptions. This work was motivated by the awareness that if additional contextual knowledge is appropriately utilized, one can develop a visualization system that systematically translates recognized objects and their relationships into a collection of one or more cohesively assembled pictures. VisuText first translates text into a computable representation, called SP form. SP forms are then converted into schematic diagrams by combining words and appropriate small images which themselves are stitched together to form a bigger meaningful picture. VisuText is especially suited for visualizing text that describes processes, particularly, those expressing similar facts and relationships in a large quantity. We find one excellent application area of VisuText is using it as a post-processing step after gene regulatory relationships are extracted through text mining of biomedical literature to pictorially represent discovered gene regulatory relationships for easier understanding by biomedical scientists. We illustrate how VisuText works by creating a pictorial representation of gene regulatory relationships from a set of statements extracted from the biomedical literature.**

Keywords-Text visualization; document visualization; natural language processing; text; semantic processing; dynamic ontology development; collaboration system; information retrieval; search; biomedical literature mining; gene regulatory relationships; cell signalling; picture rendering.

I. Introduction

The adage “a picture is worth a thousand words” is universally applicable when biomedical scientists summarize gene regulatory relationships from the literature. In the biological literature genomic structures, proteins, and other phenonena are generally described using natural language. The textual descriptions recount of elements that interact in very complex ways and the manner in which the elements interact to express gene regulatory and/or cell signaling relationships. Grasping these complex descriptions when they are presented in text is not an easy task. The problem becomes more difficult when these descriptions are not contained within a single document, but dispersed throughout various documents and need to be combined. The biomedical community, particularly, those working on discovering gene regulatory relationships face this problem more seriously, because each scientist may work on only a small set of genes and yet the community need to understand the big picture of how over 27,000 genes (in human case) work together.

In general, scientists currently read the biomedical literature and manually create schematic diagrams depicting the gene regulatory relationships summarized in the text. Examples include BioCarta [11], KEGG [12], and GenMapp [13]. They also extend existing diagrams when new information is garnered from the literature. The diagrams that they create, being a more adequate medium than language in conceptualizing complex interactions, help researchers quickly comprehend gene regulation relationships. Unfortunately, this process to convert textual information into a schematic diagram is done manually in most cases—an activity that is very laborious and prone to error. The question is whether one can design a system that can automate the process of generating pictorial representation of complex relationships, at least to a substantial degree, if doing that cannot be done entirely automatically.

This paper describes a system, called VisuText and its application in creating pictorial diagrams of gene regulatory relationships from textual descriptions. VisuText has been evolved from one of our earlier system, namely SPS [1, 2, 3], a system that performs phrase search of Web content and uses semantic processing to produce search results of very high quality and relevance.

In the rest of this paper, we describe VisuText in the following way. Section II discusses related works. Section III briefly describes SPS, the precursor of VisuText. Section IV describes VisuText’s architecture. Section V describes Picture Painter, a VisuText component that creates schematic representations from text. Section VI presents an example text and demonstrates how VisuText creates schematic diagrams. Finally, Section VII is the conclusion.

II. Related Work

The previous works in visualizing texts are generally categorized into two groups, analytic ones [4, 5, 6] and artistic ones [7, 8, 9]. The analytic approaches include phrase nets [4], word trees [5], and two-word tag clouds [6]. The artistic approaches include Literary Organism Maps [7], Document Contrast Diagrams [8], and Directed Sentence Diagrams [9]. The artistic ones, generally, have no tie-in between the text and its depiction, and we consider they are remotely related to our work. We omit further discussion of this genre of works.

Phrase nets [4] visualize relationships indicated by a pattern (e.g., as shown many times in Bible, "X begat Y"), between words or phrases. A phrase net displays a graph whose nodes are words (i.e., one node for X and one node for Y) and whose edges indicate that the two words are linked by the user-indicated relation (e.g., “begat”). A high frequency pattern is displayed using a larger font size.

Word trees [5] visualize a user-supplied word or phrase and all the different phrases that follow it. The user-supplied phrases and the follow-up phrases are arranged in a tree-like branching structure.

Two-word tag clouds [6] show the most frequent two-word phrases in a body of text. Each two-word tag is displayed with font size varying by frequency of occurrence of the two-word phrase. Since two-word tag clouds provide more contexts by adding an additional word, this method aims to give a better sense of the text content than a single-word cloud.

The aforementioned approaches are mostly concerned with visualizing text words “in verbatim”, meaning they merely transform texts/phrases into either two- or multi-dimensional representation of expressed words in their exact forms. In contrast, our approach aims at visualizing phrases/sentences after extracting semantic meanings associated with them and use “that understanding” in formulating pictorial counterparts in which the diagrams may contain rephrased words and related words in strategic locations along with contextual images so that the whole picture can provide scientists with the intuition inferred in the adage “a picture is worth a thousand words”. Our approach first captures the text meaning (i.e., the context of the stated phrases/sentences), discourse structure, and discourse thread by using a computable knowledge representation. We then visualize, using pictograms that differ from the text they depict, the meaning of the text and not the text itself. For example, "cell wall" is depicted as an arc, "cell nucleus" as a circle, "interact" and "activate" as arrows, etc. Guided by the discourse structure and thread, the pictograms are combined into a schematic picture that reflects the totality of the stated text meaning.

III. SPS AS GENESIS OF VISUTEXT

Semantic Processing System (SPS) was initially developed to improve the relevancy of web search results. Web search can be divided into two phases: a “look” phase and a “find” phase. In the “look” phase a user presents keywords to a search engine and the search engine returns a set of pages the engine considers relevant to the user. In the “find” phase the user sifts through the search engine results to find the actual relevant/interesting information.

In SPS the "look" phase is performed by the *retrieval* subsystem, which receives a user’s phrase query, increases the quality of the keywords contained in the phrase query, and using a traditional search engine retrieves web pages containing those keywords. The "find" phase is carried out by the *relevance* subsystem, which automates the user cognitive task of sifting through search engine results (i.e., retrieved pages) to find the actual relevant/interesting information. A detailed SPS description can be found in [1, 2, 3].

VisuText is a spinoff of SPS in the sense that we conjectured use of three SPS components, SP Form, NL Parser, and Knowledge Lattice, could form a solid foundation for developing an automated visualization method that can pictorially depict relationships obtained from SPS driven discoveries. In particular, when the SPS discoveries find a large amount of similar, homogenous facts/relationships, we hypothesize that by utilizing additional contextual knowledge, one can develop a visualization system that systematically positions recognized objects and their respective relationships into one or more cohesive pictures.

IV. VISUTEXT ARCHITECTURE

The overview of VisuText architecture is presented in Figure 1. It consists of a GUI, Picture Painter, and three SPS components: SP Form, NL Parser (not shown), and Knowledge Lattice.

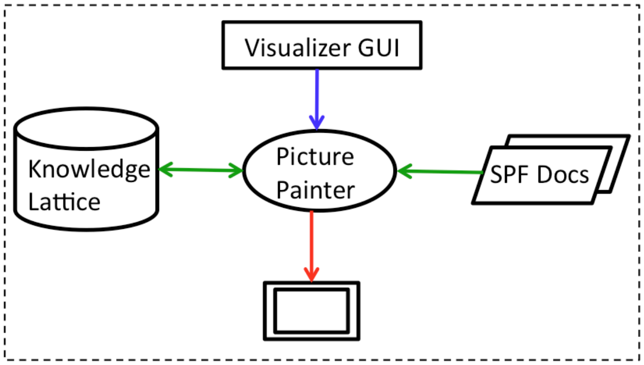


Figure 1. VisuText Architecture

A. *SP Form*

SP form [1] is the internal knowledge representation formalism used by both SPS and VisuText. A SP form expresses a sentence lexical structure in a computable format. A sentence consists of multiple phrases/clauses. Each phrase/clause is composed of syntactic and semantic elements. Syntactic elements, i.e., subject, verb, object, complement, adverb, etc. are *participants* in the meaning of a clause. Semantic elements, i.e., agent, instrument, affected, etc. are *roles* participants play. Each phrase/clause is encoded in SP form as a triple comprising a role and two participants.

(<role> (<direction1> <participant1>) (<direction2> <participant2>))

The collection of such phrases (i.e., SP forms) constitutes a sentence.

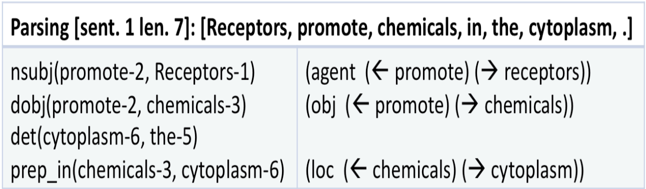
For example,

(agent (← activate) (→ chemicals))
"agent *of* activate *is* chemicals"

The direction symbol → that points away from the role is read as "is", and the direction symbol ← that points to the role is read as "of".

B. *NL Parser: Stanford typed dependencies*

NL Parser implements the Stanford typed dependency (SD) [10] parser. The SD parser represents sentence grammatical relationships as typed dependency relations, i.e., triples of a relation between pairs of words, such as "the subject of *promote* is *receptors*" in the sentence "*Receptors promote chemicals in the cytoplasm*". Each sentence word (except head of sentence) is the dependent of one other word. The dependencies are represented as *relation_name* (<*governor*>, <*dependent*>). All are binary relations: grammatical relation holds between a governor and a dependent.



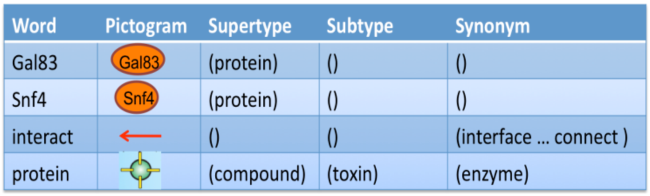
Parsing [sent. 1 len. 7]: [Receptors, promote, chemicals, in, the, cytoplasm, .]	
nsubj(promote-2, Receptors-1)	(agent (← promote) (→ receptors))
dobj(promote-2, chemicals-3)	(obj (← promote) (→ chemicals))
det(cytoplasm-6, the-5)	
prep_in(chemicals-3, cytoplasm-6)	(loc (← chemicals) (→ cytoplasm))

Figure 2. Parser dependency output and SP Form

The representation, as triples of a relation between pairs of words, is well suited for mapping SD parser output to SP forms. Figure 2 shows an SD parse of the sentence "*Receptors promote chemicals in the cytoplasm*". The parse output, i.e., the syntax tree (not shown) and the SD dependencies, is mapped to SP forms.

C. *Knowledge Lattice / Image Element Depictions*

The *Knowledge Lattice* (KL) is a data structure for storing words, their subtype / supertype relationships, their synonyms, and their pictograms. Pictograms are used to compose pictures from text. The subtype / super-type relations comprise the word's hypernyms and hyponyms. Note that the KL stores no word definitions. Included with the data structure is a set of operations for reasoning about the relations between words. The Knowledge Lattice is updated and extended by the *Interactive Learning Component*. A detailed description of the Knowledge Lattice and Interactive Learning Component can be found in [3].



Word	Pictogram	Supertype	Subtype	Synonym
Gal83	Gal83	(protein)	()	()
Snf4	Snf4	(protein)	()	()
interact	←	()	()	(interface … connect)
protein	⌖	(compound)	(toxin)	(enzyme)

Figure 3. Knowledge Lattice Fragment

Figure 3 shows the computational representation of a Knowledge Lattice fragment. In Figure 3, the word "interact" has an arrow pictogram, no supertype or subtype, but many different synonyms. When composing a picture involving the word "interact" or any of its synonyms, the red arrow is used in the picture's composition. The arrow orientation is determined by the phrase in which the word "interact" occurs.

1) *Knowledge Lattice Operations*

Knowledge Lattice Operations, described in [3], compute word synonyms, hypernyms, and hyponyms. When a word pictogram is missing, the pictogram of the word's synonym or the word's supertype could be used in picture composition.

V. Picture Painter

Picture Painter creates and renders the actual text visualization. Picture Painter interprets *SPS logical form* as a *Picture Description Language* (PDL), creates images from phrases, combines the various images into a whole (i.e., a picture), and finally places the whole into a frame for viewing.

A. *Picture Description Language*

Words are the *SPS logical form* primitives. Words are combined to create an SP form expression, which consists of a role and two participants (see Section IV.A). A collection of SP forms constitutes a sentence. Picture Painter re-interprets SPS logical form as a *Picture Description Language* (PDL).

PDL is treated as a pictorial analogue of SPS logical form. In PDL, words are still primitives; but the words are interpreted as pictograms -- words (and their synonyms) are bound to pictograms in the Knowledge Lattice (see Figure 3). Words (i.e., pictograms) are combined to form images, which are the pictorial analogue of SP forms. A collection of images is combined into a picture.

B. *Picture Composition*

A picture is composed bottom-up by first creating an image (i.e., the pictorial analogue of SP form), and then combining the images.

1) *Image Creation*

Image creation is specified by the following rules.

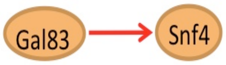
1. Roles determine pictogram orientation.
2. Participants denote pictograms.
3. Links signify the connected participants.
   A link is the common participant that connects two or more roles in multiple SP forms.
4. A connector is the link pictogram. A connector generally has two ends for stitching the participants.

An example illustrates application of these rules.

(agent (← interact) (→ (protein Gal83))
(obj (← interact) (→ (protein Snf4))

The roles are "agent" and "obj". The participants (i.e., pictograms) are "interact", "Gal83", and "Snf4" (see Figure

3). The link is “interact”. The connector is the “interact” pictogram. *Image-creation* aligns the “agent” participant to the connector base and the “obj” participant to the connector top, thus producing the following image:



Note that if “Gal83” or “Snf4” were not bound to pictograms, the participant’s supertype (i.e., “protein”) pictogram would be used.

2) *Image Combination*

Image combination is specified by the following rules.

1. Links signify the connected images. A link is a common participant that connects two or more images, or an image and a pictogram.
2. Role-of-link determines the alignment/orientation of images, or image and pictogram.

Application of these rules is illustrated by the example: "*Growth factors attach to receptors in the cell membrane*." This sentence’s PDL, separated into the images it produces, is shown below. Rendered images are shown in Figure 4 dashed-rectangle 1.

Image 1:
(type (← factors) (→ growth))
(agent (← attach) (→ factors))
(dest (← attach) (→ receptors))

Image 2:
(type (← membrane) (→ cell))
(loc (← receptors) (→ membrane))

Image 1 has three distinct pictograms: “attach”, “growth factors”, and “receptors” (“type” role dictates that “growth” and “factors” be treated as a single pictogram). Image 2 has only the “membrane” pictogram; “receptors” pictogram is available from image 1. The link that connects the two images is “receptors”, and the role-of-link is “loc” which connects image 1 to the pictogram “membrane”. The role-of-link is “loc” instead of “dest” because in the “loc” phrase the link participant is an “of” participant. The role “type” causes a labeling, which is handled by picture rendering.

Image-creation stitches “growth factors” to “receptors” to create image 1 (i.e., grouping). *Image-combination* stitches “receptors” on “membrane”. The result of image-combination is shown in Figure 4 dashed-rectangle 1.

C. *Picture Rendering*

When images are created and stitched together, the picture that is formed is placed within a parallelogram-shaped frame for viewing. Picture orientation and alignment of its elements is determined by rendering primitives, type of pictogram, and amount of zoom.

1) *Rendering Primitives*

During image-combination, as images are created and stitched together, a *rendering expression* is formed. A rendering expression is built from the following primitives.

```
<expression> ::=  <id>
                  | (beside <id> <expression>)
                  | (below  <id> <expression>)
                  | (diag1 <id> expression)
                  | (diag2 <id> expression)
                  | (on <id> <expression>)
        <id> ::=  <pictogram> | <image>
```

The discourse thread guides the expression formation. Pictograms within images (and picture) are linearly arranged/aligned in the relative direction of the discourse thread. The completed rendered expression is used by the rendering system to place/locate the images within the frame. Placement can be *vertical*, *horizontal*, *diagonal*, or *scattered*. A scattered placement results when no discourse thread exists, but the text nonetheless has common elements (e.g., sentences, with common participants, collected from different documents).

2) *Pictogram Types*

A picture is composed of images, which are in turn composed of pictograms. Pictograms are of two types: mutable and immutable. Immutable pictograms cannot be scaled. All pictograms stored in the KL are immutable and their relative size is constant.

Mutable pictograms instead can be scaled and stretched. These pictograms do not actually exist in the KL, but instead are drawn by the rendering system. Examples of such pictograms are arcs, circles, ovals, lines, rectangles, hexagons, pentagons, diamonds, etc.

Mutable pictograms depict entities that are containers for other entities. For example, an arc could represent a cell wall and an oval could represent the cell itself. Pictograms of entities such as cells must be mutable, because as more elements are placed inside the cell, the extent of the cell (i.e., oval) and the size of the cell wall (i.e., arc) need to increase.

Also, if an entity contains another identical entity, both entities to be distinguished must be of different size. For example, if a cell and cell nucleus are both represented as a circle, the two circles must both be of different size, with the outer circle bigger than the inner circle.

3) *Zoom*

The picture elements visible within a frame depend on whether a picture is rendered from a long-shot or a close-up (i.e., zoom). For example, a zoom-in of a cell might show only a portion of the cell membrane near the frame edge and a very large cell nucleus, whereas a zoom-out of a cell would show the entire cell membrane within the frame and a tiny cell nucleus.

VI. Example: Picture Composition and Rendering

This section illustrates, via an example, the workings of VisuText as it converts a natural language text into a picture.

The natural language text:

"Growth factors attach to receptors in the cell membrane[1]. The receptors promote chemicals in the cytoplasm[2]. The cytoplasm chemicals activate kinases[3]. Kinases activate chemicals that can pass through the wall of the cell nucleus to turn-on transcription factors[4]. Transcription factors turn-on the genes that make the cell divide[5]."

A. Natural Language Parsing

NL Parser translates the text into SP form (numbers correspond to sentence identifiers given in paragraph).

1: a. (type (← factors) (→ growth))
b. (agent (← attach) (→factors))
c. (dest (← attach) (→ receptors))
d. (type (← membrane) (→ cell))
e. (loc (← receptors) (→ membrane))

2: a. (agent (← promote) (→ receptors))
b. (obj (← promote) (→ chemicals))
c. (loc (← chemicals) (→ cytoplasm))

3: a. (type (← chemicals) (→ cytoplasm))
b. (agent (← activate) (→ chemicals))
c. (obj (← activate) (→ kinases))

4: a. (agent (← activate) (→ kinases))
b. (obj (← activate) (→ chemicals))
c. (agent (← pass) (→ chemicals))
d. (affirm (← pass) (→ can))
e. (obj (← pass) (→ wall))
f. (type (← nucleus) (→ cell))
g. (kind (← wall) (→ nucleus))
h. (agent (← turn-on) (→ chemicals))
i. (obj (← turn-on) (→ factors))
j. (type (← factors) (→ transcription))

5: a. (agent (← turn-on) (→ factors))
b. (obj (← turn-on) (→ genes))
c. (type (← factors) (→ transcription))
d. (agent (← make) (→ genes))
e. (agent (← divide) (→ cell))
f. (result (← make) (→ divide))

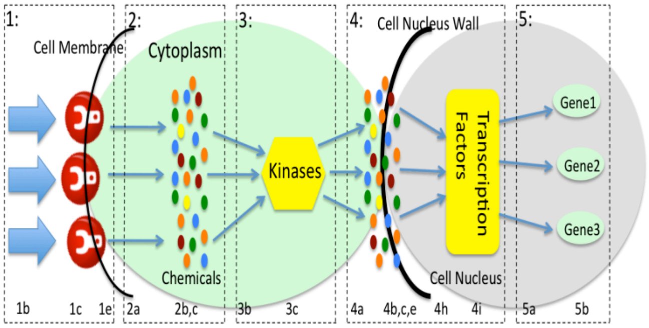
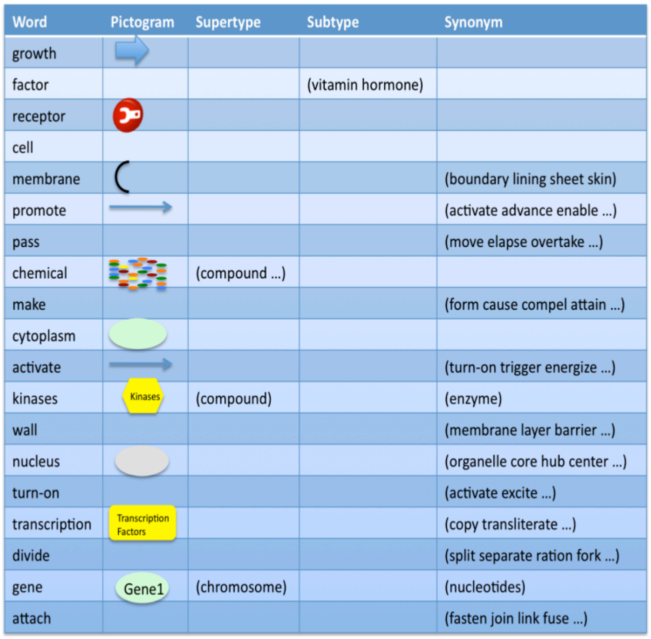


Figure 4. Picture Rendering -- Horizontal

B. Image Creation, Combination, and Rendering

Picture Painter interprets the SP forms as PDL. From each sentence, an image is created and combined with images from other sentences. Figure 4 shows the images created. Each dotted box encloses an image that corresponds (via the number) to the PDL clauses (i.e., sentence) from which the image is created. Numbers beneath the pictograms refer to the individual clause from which the pictogram is derived. Figure 5 shows a KL fragment containing pictogram depictions of various words.



Word	Pictogram	Supertype	Subtype	Synonym
growth	➡			
factor			(vitamin hormone)	
receptor	●			
cell				
membrane	(			(boundary lining sheet skin)
promote	→			(activate advance enable ...)
pass				(move elapse overtake ...)
chemical	●	(compound ...)		
make				(form cause compel attain ...)
cytoplasm	⬭			
activate	→			(turn-on trigger energize ...)
kinases	Kinases	(compound)		(enzyme)
wall				(membrane layer barrier ...)
nucleus	⬭			(organelle core hub center ...)
turn-on				(activate excite ...)
transcription	Transcription Factors			(copy transliterate ...)
divide				(split separate ration fork ...)
gene	Gene1	(chromosome)		(nucleotides)
attach				(fasten join link fuse ...)

Figure 5. KL Participant Pictogram Depiction

1) Image Creation

a) Box 1:

Box 1 says that the "agent" of "attach" is "factors" which are of type "growth", the "dest" of "attach" is "receptors", and the "loc" of "receptors" is "membrane" which is of type "cell". Figure 5 shows the KL pictogram depiction of these participants: "growth factors" (1b) as a thick solid blue arrow, "receptors" (1c) as wrench symbol, and "membrane" (1e) as a black arc. These participants are united according to the rules in Section V.B.1 to create the box 1 image.

b) Box 2:

Box 2 says that the "agent" of "promote" is "receptors", the "obj" of "promote" is "chemicals", and that the "loc" of "chemicals" is the "cytoplasm". Pictogram depictions: "promote" (2a) as a thin blue arrow, "chemicals" (2b) as a collection of multi-color ovals, and "cytoplasm" (2c) as a large green oval. These participants are united, according to Section V.B.1 rules, to form the image shown in box 2.

c) Box 3:

Pictogram depictions: "activate" (3b) as a single solid blue arrow with a thin body and "kinases" (3c) as a labeled yellow hexagon. Note that if "activate" did not have a

ISBN: 978-1-61208-202-8

pictogram, then the pictogram of its synonym "promote" could be used.

d) Box 4:

Box 4 contains many participants that have already been seen from boxes 1, 2, and 3. The new participants are "pass", "nucleus", "wall", "turn-on", and "transcription". Of these participants only "nucleus" (4i), which is depicted as a grey oval and "transcription" (4ji), which is depicted as a rounded rectangle, have a pictogram. "Pass", which in the context of sentence 4 does not denote a thing but instead describes an occurrence, does not have an associated pictogram. "Wall" and "turn-on" also do not have a pictogram; consequently, the synonyms of their pictograms are used: a black arc for "wall" (4e), and a thin blue arrow for "turn-on" (4h).

e) Box 5:

The new box 5 participant is "gene" (5b) which is depicted as a labeled light green oval.

2) *Image Combination*

Participants (i.e., pictograms) are united according to the rules in Section V.B.1 to create images (see Figure 4 boxes). Images are stitched together to create a picture.

The combining of images is guided by the discourse thread, which is encoded as a rendering expression (section V.C.1). In this example the discourse thread is "*growth factors – receptors – chemicals – kinases – chemicals – cell nucleus – chemicals – transcription factors – cell divide*".

The rendering expression is *(beside (beside (beside (beside 1b (on 1d 1e)) (beside 2a (on 2b 2c))) (beside 3b 3c)) (beside (beside 4a (on 4b 4e)) (beside 4h 4i)) (beside 5a 5b))*.

3) *Picture Rendering*

The rendering system uses the rendering expression to place/locate the stitched images within a frame. Placement can be vertical, horizontal, diagonal, or scattered. In Figure 4 placement is horizontal.

VII. Conclusion

We have presented a framework that is designed to carry out a post processing following a text mining step in order to covert the recognized relationships obtained from a text mining into a set of pictorial diagrams. We demonstrated that our automated methodology is well suited for better representing text mining outcomes of gene regulatory relationships buried in the biomedical literature. Using a series of examples we have illustrated that our proposed method can indeed capture textual meanings of stated words using knowledge lattice and can create visual depiction of the key elements of the involved objects at the appropriate conceptual level automatically. In a nutshell, we point out that incorporating this extra layer of visual knowledge into the picture creation is what makes the user's understanding of diagrams far more intuitive than simple narration of multiple related sentences. VisuText is especially suited for visualizing text that describe processes, such as gene regulatory relationships, which consist of various elements that interact with each other or trigger interactions. Currently we are refining the methodology and are experimenting, using large scale text mining of biomedical literature, with a prototype in order to gauge its performance.

References

[1] J. Leone, "A Semantic Processing System (SPS) for Web Search", Ph.D thesis, University of Connecticut, 2011 (under preparation).

[2] J. Leone and D. G. Shin, "SPS: A Web Content Search System Utilizing Semantic Processing," Content 2011: The Third International Conference on Creative Content Technologies, Rome, Italy, September 25-30, 2011.

[3] J. Leone and D. G. Shin, "A Semantic Processing System (SPS) for Biomedical Literature Web Search," Advances in Data Mining 11th Industrial Conference, ICDM 2011, New York, USA, August/September 2011.

[4] F. van Ham, M. Wattenberg, and F. B. Viégas, "Mapping Text with Phrase Nets", Proc. IEEE InfoVis, 2009.
http://www.research.ibm.com/visual/papers/phrase-net-rev5.pdf [retrieved: April, 2012]

[5] M. Wattenberg and F. Viégas, "The Word Tree: An Interactive Visual Concordance," Proc. IEEE InfoVis, 2008.
http://researchweb.watson.ibm.com/visual/papers/wordtree_final2.pdf [retrieved: April, 2012]

[6] F. Viégas and M. Wattenberg, "Tag Clouds and the Case for Vernacular Visualization", ACM Interactions, XV.4 July/August, 2008.
http://www.research.ibm.com/visual/papers/vernacular_visualization.pdf [retrieved: April, 2012]

[7] http://www.itsbeenreal.co.uk/index.php?/wwwords/literary-organism [retrieved: April, 2012]

[8] http://www.neoformix.com/2008/DocumentContrastDiagrams.html [retrieved: April, 2012]

[9] http://www.neoformix.com/2008/DirectedSentenceDiagrams.html [retrieved: April, 2012]

[10] http://www-nlp.stanford.edu/software/stanford-dependencies.shtml [retrieved: April, 2012]

[11] http://www.biocarta.com/Default.aspx [retrieved: April, 2012]

[12] M. Kanehisa, S. Goto, Y. Sato, M. Furumichi, and M. Tanabe, "KEGG for integration and interpretation of large-scale molecular data sets", Nucleic Acids Res. 2012 Jan;40(Database issue):D109-14. Epub 2011 Nov.10.

[13] N. Salomonis, K. Hanspers, A. Zambon, K. Vranizan, S. Lawlor, K. Dahlquist, S. Doniger, J. Stuart, B. Conklin, and A. Pico. GenMAPP 2: new features and resources for pathway analysis. *BMC Bioinformatics, Jun 2007; 8: 217*