

# Multimodal Knowledge Capture from Text and Diagrams

**Kate Lockwood**

QRG, Northwestern University  
Evanston, IL USA  
kate@u.northwestern.edu

**Ken Forbus**

QRG, Northwestern University  
Evanston, IL USA  
forbus@northwestern.edu

## ABSTRACT

Many information sources use multiple modalities, such as textbooks, which contain both text and diagrams. Each captures information that is hard to express in the other, and evidence suggests that multimodal information leads to better retention and transfer in human learners. This paper describes a system that captures textbook knowledge, using simplified English text and sketched versions of diagrams. We present experimental results showing it can use captured knowledge to answer questions from the textbook's curriculum.

## Categories and Subject Descriptors

I.2.4 Knowledge Representation Formalisms and Methods

## INTRODUCTION

Researchers from psychology and learning sciences have examined the question of whether, and under what conditions, people learn better from multimodal presentations of information than from single-modality information (e.g. [13][16]). Many of these studies have shown that subjects are able to perform better on tests of retention and transfer when they were presented with multimodal information sources, such as animations with narration or text with diagrams. Indeed, many traditional sources of instructional material contain multiple modalities. For example, textbooks contain both text and diagrams. To exploit such materials, knowledge capture systems should be able to integrate information across modalities into coherent chunks of knowledge.

There are multiple theories as to how and why multimodal sources of information lead to better recall and transfer performance. The *multimedia learning theory* [16] posits that instead of passively absorbing information, learners cognitively engage with it in an active attempt to understand. Under this theory, multimedia presentations of information lead to better understanding because learners actively engage in sense making activities as they attempt to integrate

information from the two modalities, and it is this active engagement with the material that leads to deeper learning.

Our knowledge capture system is based on this theory, which asserts that such learning is a five step process:

- 1) Selecting relevant words for processing in verbal working memory
- 2) Selecting relevant images for processing in visual working memory
- 3) Organizing selected words into verbal mental model
- 4) Organizing selected images into visual mental model
- 5) Integrating verbal and visual representations along with existing knowledge.

In our model steps 1 and 2 (selection) are done manually by dividing the text and diagrams into discrete chunks. Text chunks are determined by paragraph structure in the text and by diagram references in the text. Each diagram is its own diagram chunk. Step 3, developing a representation from the text, is done using the EA NLU natural language understanding system described below. Extraction of information from the diagrams (step 4) is accomplished via the CogSketch sketch understanding system. The final step, Integration, uses the Structure Mapping Engine model of analogy and similarity to perform the cognitive task of comparing and integrating the two representations.

After describing each of these steps in more detail, we present an experiment in which we use our system to learn a chapter of content from a physics textbook: *Basic Machines* [3]. The system is evaluated on its ability to answer the homework questions provided by the publisher.

## SYSTEM ARCHITECTURE

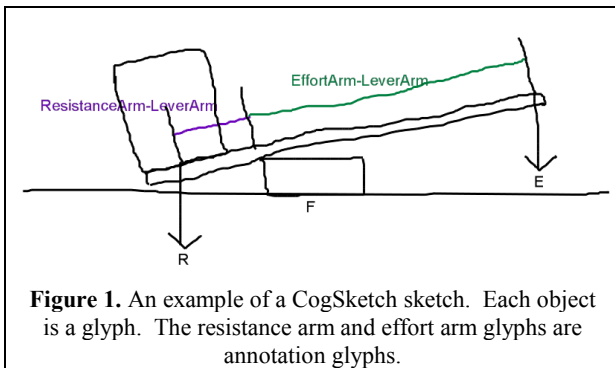
The parsing of the text and the initial interpretation of the diagrams were done using two existing systems: The CogSketch sketch understanding system for the diagrams and the EA NLU system for the text. This section briefly summarizes them and describes the methodology used to convert experiment materials from the original source format to the format they require using an example from our evaluation materials.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

## CogSketch

CogSketch<sup>1</sup> [11] is the first open-domain sketch understanding system. Each object in a CogSketch sketch is a *glyph*. Glyphs have *ink* and *content*. The ink consists of polylines, i.e., lists of points representing what the user drew. The content is a symbolic token used to represent what the glyph denotes. In CogSketch, users indicate the type of the content of the glyph in terms of concepts from an underlying knowledge base. This is one form of *conceptual labeling*. Our KB contents are drawn from ResearchCyc, which contains over 30,000 concepts, plus our own extensions. In addition to conceptual labels, the content of glyphs can also be given *names*, natural language strings that can be used to refer to the content of the glyph.

CogSketch automatically computes a number of qualitative spatial relations and attributes for glyphs in a sketch. The relations computed include the RCC-8 qualitative relations [7] that describe all possible topological relations between two-dimensional shapes (e.g. disconnected, edge-connected, partially-overlapping). RCC-8 relations are also used to guide the computation of additional spatial relationships, such as positional relations like right/left.



**Figure 1.** An example of a CogSketch sketch. Each object is a glyph. The resistance arm and effort arm glyphs are annotation glyphs.

In addition to standard glyphs, CogSketch also allows users to create *annotation glyphs*. An annotation glyph is a glyph that indicates an important property in the sketch, e.g. physical quantities like length or height. Like other glyphs, annotation glyphs consist of both ink and the entity that is represented by the glyph. Unlike other glyphs, annotation glyphs also refer to other glyphs in the sketch depicting the entity that they are providing information about. For example, if an annotation glyph represents the length of a lever, it has a reference to the lever glyph whose length it defines. Figure 1 shows an example of a CogSketch sketch. Each object is a glyph, and the resistance arm and effort arm glyphs are `LengthIndicator` annotation glyphs which are tied to the glyph representing the lever since they provide information about its properties. The resistance arm and effort arm labels are names used to identify the objects. The other labels in the sketch are also name labels (e.g. “R”). Spatial relationships computed include things like the

fulcrum being RCC8-EC (edge connected) to the lever glyph and the lever glyph being above the fulcrum glyph.

For our multimodal knowledge capture system, CogSketch is used to create the sketched equivalent of a textbook diagram. Then CogSketch creates an output case consisting of facts that describe the objects in the sketch and the qualitative spatial relationships between them.

## EA NLU

The EA NLU (Explanation Agent Natural Language Understanding) system [14][18] uses Allen’s parser [1], the COMLEX lexicon, and ResearchCyc KB contents to parse and interpret English text. Input to EA NLU is in QRG-CE (QRG Controlled English), a simplified language that we developed to simplify parsing and to allow us to focus on semantics. Translation to simplified English is far more intuitive and time-efficient than translating directly to predicate calculus. QRG-CE allows new vocabulary terms to be introduced in texts, unlike other controlled languages, which is essential for knowledge capture. The input to EA NLU is a chunk of text from the source material that has been translated into QRG-CE. The output is a case consisting of facts representing the semantic content of the text.

## SME

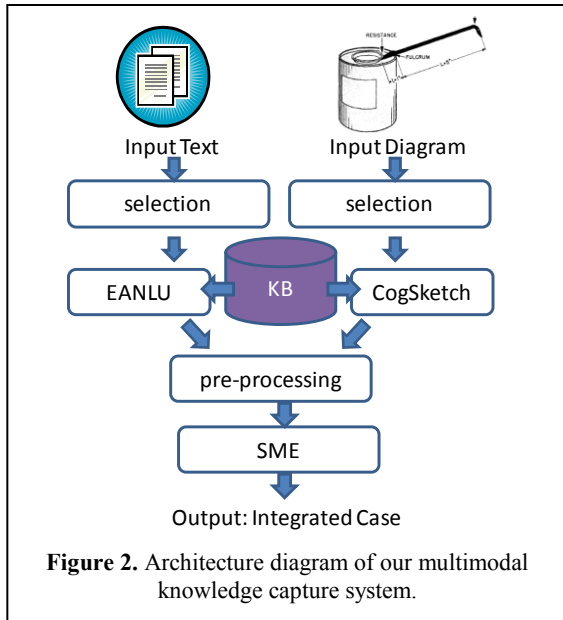
The Structure Mapping Engine (SME) [8] is a cognitive model of analogy and similarity based on Structure Mapping theory [12]. SME takes as input two cases, a *base case* and a *target case*. Each input case consists of a set of entities, attributes and relations. Given these two cases, SME produces one to three mappings between them by aligning their common structure. Each mapping contains: (1) a set of correspondences between elements (entities, attributes and relationships) in the base and elements in the target, (2) the *structural evaluation score* which is a numerical measure of similarity and (3) *candidate inferences* which are inferences that are carried over from the base to the target based upon their common structure. Structural evaluation prefers mappings which align higher-order relations. In addition to allowing SME to choose the best alignment of elements, it can be provided with *required correspondences*. Required correspondences force a correspondence between a given element in the base and one in the target, so that mappings must be consistent with this correspondence.

## System Architecture and Integration

Our system is composed of five components, including those described in the preceding sections. Figure 2 shows how these systems fit together to form our multimodal knowledge capture system (MMKCap). First, the original text is manually divided into chunks during the selection step. A chunk is defined as either (1) a distinct paragraph from the source material or (2) a portion of a paragraph that all pertains to a given diagram (as specified via a figure tag in the text, i.e., “as shown in Figure 1-1”). Since our cur-

<sup>1</sup> CogSketch is available on the web at:  
[http://spatiallearning.org/projects/cogsketch\\_index.html](http://spatiallearning.org/projects/cogsketch_index.html)

rent model is concerned with multimodal integration as opposed to the intricacies of NLP or image processing, there are several steps taken to simplify the input for processing. Text is translated to controlled English and diagrams are sketched using CogSketch.



**Figure 2.** Architecture diagram of our multimodal knowledge capture system.

During the translation to QRG-CE, changes are limited to:

- 1) Sentences that do not contribute topical information can be deleted: e.g., “[Machines] have taken much of the backache and drudgery out of the sailor’s lift.” or “Machines are your friends.” Future versions should distinguish between useful and non-content sentences automatically, but for now this is done by hand.
- 2) Long sentences or sentences containing conjunctions are broken into shorter, easier to parse sentences. (compare Figures 3 and 4 for an example of original text and its simplified counterpart)
- 3) The mathematical steps in worked examples in the text (involving equations and numerical substitutions) are hand-represented in predicate calculus to make the steps in the problem solving process clear, and available for later use. This is another step that we hope to be able to automate in the near future.
- 4) Summary information at the end of the chapter is excluded for now. This information is redundant (by definition). Later we plan to develop a method to use summary information as a first pass check of our knowledge capture

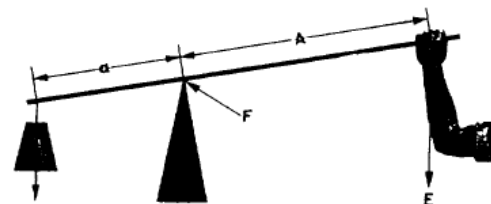
After the text has been translated into QRG-CE, each chunk is run through the EA NLU system. The output from EA NLU is a case containing facts representing the semantic content of the text. If the original text contained a reference to a diagram, a fact is manually added that provides a link

from the discourse to the sketch (see the sketchForDiscourse fact in figure 5).

Each diagram from the source material is sketched using CogSketch. Figure 3 shows an example of a text-diagram chunk from the original text and Figure 4 shows the same chunk after the text has been translated into QRG-CE and the diagram has been sketched using CogSketch. A strict set of guidelines dictates how each diagram is translated into a sketch:

- Objects in the sketch are drawn to preserve existing spatial relationships
- Objects in the sketch are only labeled in the sketch if they are labeled in the source diagram.
- Objects labeled with conceptual information in the source diagram (i.e. Fulcrum) are given a conceptual label in CogSketch
- Objects that only have identifying labels in the original source (i.e. “A”) are labeled using the glyph name in CogSketch. Note that names in CogSketch are case insensitive, so labels in the text that rely on case sensitivity must be changed to be distinguishable in CogSketch
- If an object is labeled with a numerical value in the source material then an annotation glyph is used in CogSketch with that numerical value.

You will find that all levers have three basic parts: the fulcrum (F), a force or effort (E), and a resistance (R). Look at the lever in figure 1-1. You see the pivotal point (fulcrum) (F); the effort (E), which is applied at a distance (A) from the fulcrum; and a resistance (R), which acts at a distance (a) from the fulcrum. Distances A and a are the arms of the lever.



**Figure 1-1.-A simple lever.**

**Figure 3.** An example of a text and diagram chunk taken from our source text: *Basic Machines*. In this particular example, you can see references in the text to labels in the diagram (e.g. “fulcrum (F)”) refers to the part of the diagram annotated with the same label “F”.

These guidelines are meant to assure that, as much as possible, the sketched diagram adheres to the spirit of the source material. In particular, we are careful to make sure that we do not encode any inferences that might be drawn about the diagram at this stage of the process. Once the diagram is drawn it is stored as a sketch case containing facts about the

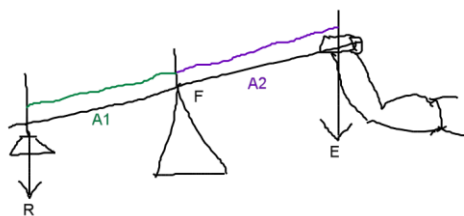
objects in the sketch and the spatial relationships between them. Both EA NLU and CogSketch make use of the ResearchCyc KB contents when constructing the output cases.

**Chunk 1:**

A lever has three basic parts.  
 A fulcrum is a basic part of a lever.  
 A force is a basic part of a lever.  
 A weight is a basic part of a lever.

**Chunk 2:**

F is the fulcrum  
 R is the weight  
 E is the force  
 A2 is the distance between the weight and the fulcrum.  
 A1 is the distance between the force and the fulcrum.  
 A1 is an arm of the lever.  
 A2 is an arm of the lever.



**Figure 4.** The text and diagram from Figure 3 after the text has been translated to QRG-CE and the diagram has been sketched in CogSketch. The text has been manually divided into two chunks: chunk 1 contains the text that does not refer to the diagram and chunk 2 is the text that does.

```
(isa lever6354 Lever)
(isa a2 LeverArm)
(possessiveRelation lever6354 a2)
(isa f Fulcrum)
(isa e ForceVector)
(isa r Weight)
(isa a2 Distance)
(between r f a2)
(isa a1 Distance)
(between e f a1)
(isa a1 LeverArm)
(isa lever6231 Lever)
(possessiveRelation lever6231 a1)
(possessiveRelation lever6231 a2)
(sketchForDiscourse
  "kb-resource://Figure1-1.sk"
  (DrsCaseFn DRS-3446218074-8197))
```

**Figure 5.** The output generated from running the simplified text through the EA NLU system. The `sketchForDiscourse` fact ties the sketch to the text and is added manually.

The case resulting from processing the text with EA NLU contains 14 facts and is shown in Figure 5. The case resulting from sketching and processing the diagram contains 203 facts. In addition to the labels for objects, it also contains qualitative spatial relationships between objects in the

sketch, such as those in Figure 6. The `visualQuantityQuantitativeMeasurement` fact lets us know that the glyph A1 is actually an annotation glyph which indicates that it has the potential to have a numerical quantity associated with it.

```
(enclosesHorizontally
  (GlyphFn Object-4 Layer-2)
  (GlyphFn Object-147 Layer-2))
(rcc8-EC (GlyphFn Object-4 Layer-2)
  (GlyphFn Object-141 Layer-2))
(visualQuantityQuantitativeMeasurement
  ((ConceptKnownAsFn "A1")
  (GlyphFn Object-4 Layer-2)) A1)
(above Object-145 Object-4)
```

**Figure 6.** An example of some of the facts generated by CogSketch about the qualitative spatial relationships in the sketched diagram.

After both the text case and the sketch case have been constructed, they are run through a pre-processing routine that prepares them for integration. The purpose of the pre-processing is to look for potential required correspondences to constrain the mapping between the two cases. Required correspondences are created under two circumstances: (1) If there is a label on the diagram that is specifically referred to in the text (for example, in Figure : “you see the pivotal point (fulcrum) (F)” where “F” refers directly to a label in the diagram. In this case, a correspondence is created between the entity F in the text case and the glyph object that is labeled with the namestring “F” in the sketch case). (2) If there is a spatial preposition in the text case (as determined by a relation that is an instance of the Cyc concept `RelativeLocationalPredicate`) a correspondence is created between the entities in the text case that appear in the spatial preposition fact and the objects in the sketch that are of the same type (note that if they were labeled they would have already been addressed in step 1). A list of chapter labels is maintained that corresponds to the namestring labels found in Step one. We have found that the same labels will often be used throughout a chapter (e.g. F to identify the fulcrum) but will only be explicitly explained in the text once. Text and diagram pairs are integrated one at a time, preserving the order in which they appear in the text.

After the required correspondences have been created, SME is run with the text case as the base case and the sketch case as the target case. The best mapping, i.e. the mapping with the highest structural evaluation score, is used to create the result of the integration step. Since the diagram case was used as the target case, the facts from the target case are transferred to a new case and form the basis for the integrated case. Then, the candidate inferences from the mapping are also added to the new integrated case – this is how the information from the text gets integrated with the information from the diagram. Text chunks that did not have any

associated diagrams are not subject to the integration process, and instead the text case is treated as the final outcome of the knowledge capture process.

## EXPERIMENT

In order to evaluate our multimodal knowledge capture system, we ran an experiment to examine its ability to capture the knowledge in one chapter of a physics textbook on basic machines.

### Materials

The source material for this experiment was Chapter 1: Levers from [3], a US Navy training manual. The original chapter contained 8 pages of text and 15 diagrams. The book also includes homework assignments, with questions tied to each chapter, providing an external benchmark to evaluate how well the system was able to capture the knowledge in the chapter.

There were 29 questions about levers, but 14 of them required solving equations, a capability which is still being incorporated in our system. Consequently, we focused on the 15 questions that did not require equations.

### Evaluation

For each of the 15 questions, the information and the query it contained were hand-translated into predicate calculus, to avoid errors in question-parsing. For questions containing diagrams, each diagram was sketched following the same procedure used for diagrams in the text. The same multimodal integration process was used to combine diagram and translated text information. Each question was assigned to a category based on the type of query it involved, since learning problem-solving methods is not being tackled in this project. The breakdown in terms of number of questions of each type is:

- 1) True/False questions {2}
- 2) Text queries: information that can be directly queried for {6}
- 3) Diagram concept queries: queries that involved identifying which diagram matches a given concept (i.e. which diagram is an example of a first class lever) {3}
- 4) Diagram measurement queries: a diagram is given and a measurement is asked for (i.e. what is the length of the resistance arm in the given diagram?) {4}

### Example Questions

True/False questions (see Figure 8) were answered by creating a query that directly asked for the information in the

question. The same technique was used for text query questions (see Figure 7). Figure 9 shows an example of the type of query that was developed to ask the system for the requested information.

Which of the following simple machines works on the same principle as the inclined plane?

1. Screw
2. Gear
3. Wheel and axle
4. Block and tackle

Figure 7: Example of a text query

When a chain hoist is used to multiply the force being exerted on a load, the chain is pulled at a faster rate than the load travels.

1. True
2. False

Figure 8: Example of a true/false question

```
(termToSolveFor
  (querySentenceOfQuery
    BasicMachines1-5
    (and (refersToTypeOf
          ?tool InclinedPlane)
         (isa ?tool ?collection)))
  BasicMachines1-5
  ?collection)
```

Figure 9: Query for the question in Figure 7

True/False and text query questions primarily draw on the information from the text content of the chapter and, for the most part, do not utilize the integrated cases created by the multimodal knowledge capture system. These types of questions serve to demonstrate the effectiveness of EA NLU in capturing the knowledge in the natural language component of the input. The other types of questions, diagram concept and diagram measurement queries, however, both rely heavily on the integration portion of the knowledge capture. Often the labels or tags that are used to find the answer to the questions come directly from the text accompanying a diagram.

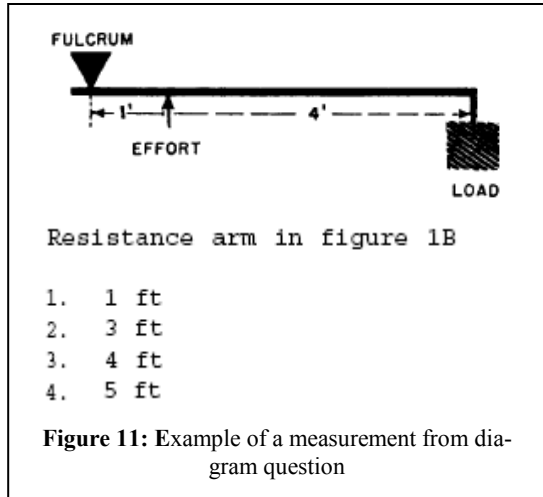
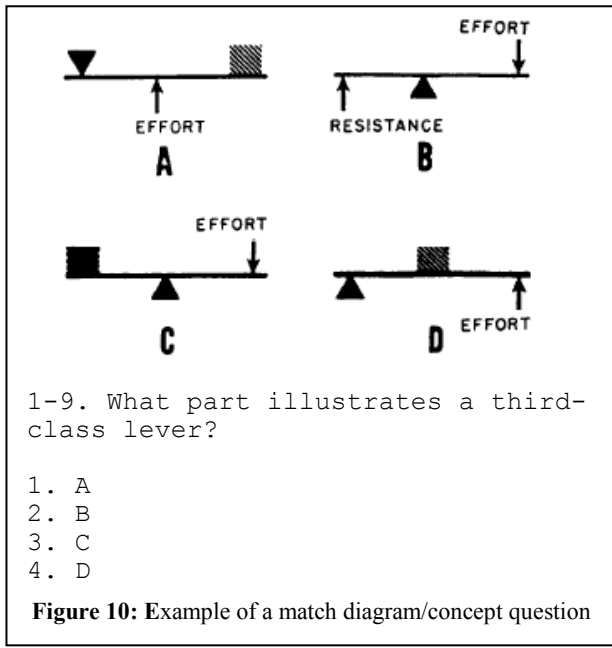


Diagram concept queries and diagram measurement queries are both special case queries that are handled using SME. Examples of diagram concept and diagram measurement queries are shown in Figure 10 and Figure 11 respectively.

To solve a diagram concept query the system looks for cases that contain a fact tying either an integrated case or discourse-only case to the concept being queried for, with preference given to integrated cases (in an integrated case, these facts are created automatically during integration, based on sentences in the text such as "...figure 1-4 is a good example of a second-class lever"). In the example in Figure 10, the system would look for a case that has a diagram that has been tied to the concept "third class lever". Once a suitable diagram has been retrieved, the system compares the known diagram to each of the diagrams in the question using SME. The diagram with the highest structural evaluation score is returned as the system's answer to the question.

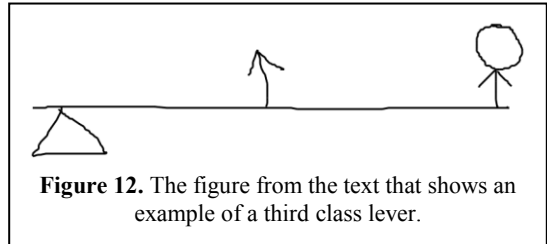
Measurement from diagram queries are evaluated in a similar way to diagram concept queries. The system looks for an existing case that has a diagram (or discourse) containing the type of object that the measurement is requested for. In the example here, the system would be looking for resistance arms. Once a sketch is retrieved, it is mapped to the sketched diagram from the problem and the mapping is examined. The glyph in the problem sketch that matches to the glyph in the retrieved sketch that is labeled with the target term is assumed to be the corresponding element in the query diagram. If the selected glyph is an annotation, the corresponding numerical value is returned. If it is not an annotation, or does not have a value associated with it, then the glyph is returned as the answer to the query.

**Results**

The system was run over all 15 questions in the evaluation set and the results were checked for correctness. Of the 15 questions in our evaluation set, the system gets 12 of them correct (80%). This is statistically significant ( $P < 10^{-5}$ ).

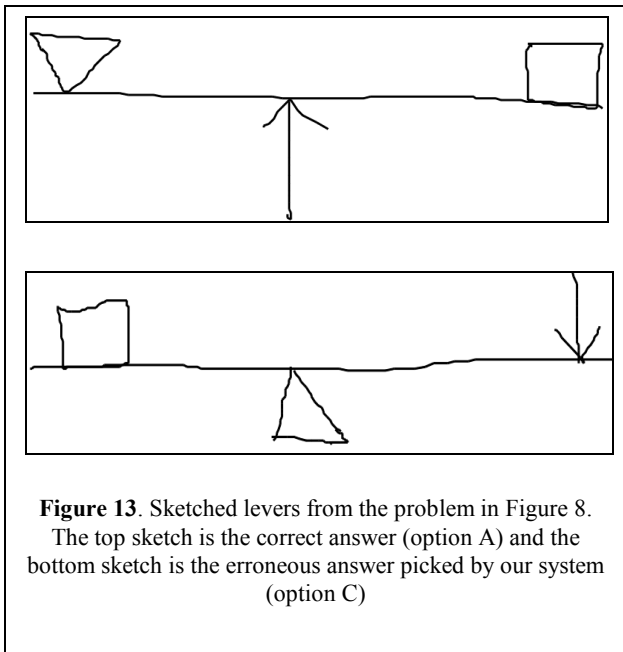
**Failure Analysis**

A close examination of the failures provides additional insights and suggests improvements. One of the three failed questions is a diagram/concept matching question, while the other two are measurement from diagram questions. The diagram/concept matching question that fails is the example in Figure 10. Figure 12 shows the diagram that is retrieved as the example of a third class lever from the text. It is compared (via SME) to the sketched versions of the levers in the problem in Figure 10.



Two of the sketches are shown in Figure 13. The system should match to the top sketch (option A - which is an example of a third class lever) but instead matches to the bottom sketch (option C - which is an example of a first class lever). This mistake occurs because our system currently cannot recognize that the lever in the top of Figure 13 needs to be flipped over the horizontal axis, so that the fulcrum is underneath the lever in order for the comparison to work correctly. As the pictures are currently drawn, the matching system is overwhelmed by the number of spatial relationships in common between the retrieved sketch and the sketch in the bottom of Figure 13 (e.g. the fulcrum being below the lever and the force and weight being above) and those relations override the important one, which is the placement of the fulcrum relative to the weight and effort. This suggests verifying properties of the match, i.e., that forces are being applied in ways consistent with the learned

definition. If verification fails, then rerepresentation techniques based on spatial properties, like flipping or rotating one of the sketches, could be tried.



The other two failures, the diagram measurement queries, illustrate another shortcoming of our problem solving strategy of using a simple, unevaluated match to produce an answer. The problem in Figure 11 is the one of the two failed questions of this type. Our current problem solver expects there to be a single annotation glyph it can match against to compute the answer. Here, what is required is to realize that the length is the sum of the two glyphs that are in the diagram. Again, this is more a failure of our problem solver than of the knowledge capture process.

## RELATED WORK

Our work is related to a variety of systems from the knowledge capture and learning by reading communities. Ferguson's JUXTA [9] system reasoned about juxtaposition diagrams, diagrams that use comparison to illustrate physical principles, by using information from both the diagram and the caption. JUXTA relied on a pre-defined mapping from shapes to domain-specific meaning and required all of the captions to be hand-translated to qualitative physics representations. Bulko's BEATRIX [4] system was able to solve the coreference problem for physics problems that contained both text and a diagram. It relied on a blackboard architecture to align objects in the diagram with their references in the text. BEATRIX relied on hand-coded knowledge sources to identify potential objects in the diagrams in its system. The Figure Understander [17] was also developed to integrated text and diagram representations in the physics domain. Figure Understander was used to input problems into a magnetic fields problem solving system. The system relied on a system of figure semantics that re-

lated shading and patterns in the diagram with a semantic interpretation for the items (for example, a circle with white shading represented a loop of conducting wire, while black shading represented an immobile supporting object). In contrast to these approaches, our system uses the more flexible concept labeling feature of CogSketch to allow users to enter objects and attach meaning without relying on a pre-defined library of mappings.

Watanabe and Nagao [19] used a combination of spatial information and simple parsing rules to categorize the text associated with diagrams in a Japanese pictorial book of flora. Their method was specifically aimed at being able to classify the type of text (whether it described a plant species, plant part, etc) based on a combination of textual and spatial information and was limited to the domain of Japanese wild flowers. They also hand-coded the spatial relationships between text and diagrams. Currently we do not take advantage of this type of information since we do not capture information about the placement of the conceptual label text. This suggests an interesting area of future work for our system, i.e., developing methods to capture the information that can be gleaned from the placement of a label in a textbook diagram.

The HALO project [6][5] also addressed learning from textbooks and solving problems with the captured knowledge. The AURA system provided an interface for subject matter experts to input the knowledge from 50 pages of textbooks in each of physics, chemistry, and biology. The associated question answering system used a controlled language to allow users to input AP-like test questions for the system to solve. The approach used in AURA focused on human-generated knowledge and on conceptual knowledge and tables (diagrams were excluded). We view this approach as complementary to ours.

Our work can be viewed as a particular form of learning by reading. The closest systems are Mobius [2] and Learning Reader [10]. Mobius was used to see how existing NL and KR components could be combined to learn from text. It focused on two narrow domains (how human hearts and simple engines work), but was tested with a variety of paragraphs written by different people about those topics. Its knowledge base was small and hand-coded for the domain, and its learned knowledge was evaluated by hand inspection. Learning Reader, like our work, uses simplified English and ResearchCyc KB contents, but a DMAP parser instead of the more traditional EANLU system used here. Learning Reader was tested via automatically generated quizzes, and incorporated a process of rumination, where the system asked itself questions off-line to improve its performance later on. Both Mobius and Learning Reader were purely text-based, unlike our multimodal approach. The system described here will be part of a next-generation learning by reading system, incorporating ideas from Learning Reader.



## CONCLUSIONS AND FUTURE WORK

We have described our model of multimodal knowledge capture that is inspired by Mayer's theory of multimedia understanding. We have also described the results of an experiment indicating that structure mapping is a powerful tool for integrating knowledge from multiple modalities. Our first step in future work is to integrate an existing algebraic problem solver into our system to expand the range of questions that the system can handle.

There are many additional improvements that we plan to make, including automating the translation of example problems and test problems and perhaps even some of the text-diagram chunking. This project began using a hard copy of the source textbook, in future work we hope to examine electronic sources to see what additional leverage we can get from tags and links within the documents.

As mentioned above, we plan to investigate further how we process the sketched diagrams. One avenue for this is to process spatial information about label location, which will be incorporated into CogSketch. Also, the sketches in this source were fairly simple. However, prior work [15] has shown that parsing diagrams can be much more complex and require the use of functional knowledge about the entities in a sketch and even naïve physics. We hope to develop a robust algorithm for diagram parsing and to integrate it into our multimodal knowledge capture system.

## ACKNOWLEDGMENTS

This work is sponsored by a grant from the Intelligent Systems Division of the Office of Naval Research and an AAUW dissertation year fellowship.

## REFERENCES

- [1] Allen, J. F. *Natural Language Understanding*. (2nd ed). Redwood City, CA.: Benjamin/Cummings. (1995).
- [2] Barker, K. *et al.* Learning by Reading: A Prototype System, Performance Baseline and Lessons Learned. *Proceedings of AAAI 2007*. (2007)
- [3] *Basic Machines*. Navy Education and Training Professional Development and Technology Center. (1994).
- [4] Bulko, W.C. Understanding Text with an Accompanying Diagram. *Proceedings of the 1<sup>st</sup> International Conference on Industrial Engineering Applications of Artificial Intelligence and Expert Systems – Volume 2*, 894-898. (1988).
- [5] Chaudhri, V. *et al.* Enabling Experts to Build Knowledge Bases from Science Textbooks. *KCap'07*. (2007).
- [6] Clark, P. *et al.* Capturing and Answering Questions Posed to a Knowledge-Based System. *KCap'07*. (2007).
- [7] Cohn, A. Calculi for qualitative spatial reasoning. In *Artificial Intelligence and Symbolic Mathematical Computations*, LNCS 1138, eds: J Calmet, J.A. Campbell, J Pfalzgraph. Springer Verlag 124-143. (1996).
- [8] Falkenhainer, B, Forbus, K., & Gentner, D. The Structure-mapping Engine: Algorithm and Examples. *Artificial Intelligence*, 42, 1-63. (1989).
- [9] Ferguson, R. W., & Forbus, K. Understanding Illustrations of Physical Laws by Integrating Differences in Visual and Textual Representations. *AAAI Fall Symposium on Computational Models for Integrating Language and Vision*. (1995).
- [10] Forbus, K., Riesbeck, C., Birnbaum, L., Livingston, K., Sharma, A., & Ureel, L. (2007). A Prototype System that Learns by Reading Simplified Texts. AAAI Spring Symposium on Machine Reading. Stanford University, California.
- [11] Forbus, K., *et al.* CogSketch: Open-domain sketch understanding for cognitive science research and for education. *Proceedings of the Fifth Eurographics Workshop on Sketch-Based Interfaces and Modeling*. (2008).
- [12] Gentner, D. Structure-mapping: A Theoretical Framework for Analogy. *Cognitive Science*, 7, 155-170. (1983).
- [13] Hegarty, M. & Just, M.A. Constructing Mental Models of Machines from Text and Diagrams. *Journal of Memory and Language*. 32, 717-742 (1993).
- [14] Kuehne, S. and Forbus, K. Capturing QP-Relevant Information from Natural Language Text. *Proceedings of QR04*. (2004).
- [15] Lockwood, K., *et al.* A Theory of Depiction for Sketches of Physical Systems. *Proceedings of the Qualitative Reasoning Workshop*. (2008).
- [16] Mayer, R.E. *Multimedia Learning*. Cambridge University Press. (2001).
- [17] Rajagopalan, R. & Kuipers, B. The Figure Understander: A System for Integrating Text and Diagram Input to a Knowledge Base. *Proceedings of 7<sup>th</sup> Annual Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*. (1994).
- [18] Tomai, E. and Forbus, K. EA NLU: Practical Language Understanding for Cognitive Modeling. *Proceedings of the 22nd International Florida Artificial Intelligence Research Society Conference*. Sanibel Island, Florida. (2009).
- [19] Watanabe, Y., & Nagao, M. Diagram Understanding using Integration of Layout Information and Textual Information. *Proceedings of the 36<sup>th</sup> Annual Meeting of the Association for Computational Linguistics and 17<sup>th</sup> International Conference on Computational Linguistics – Volume 2*, 1374-1380. (1998)