Large-Scale Analogical Reasoning

Vinay K. Chaudhri, Stijn Heymans, Aaron Spaulding, Adam Overholtzer, Michael Wessel
Artificial Intelligence Center, SRI International
Menlo Park, CA 94025, USA

Abstract

Cognitive simulation of analogical processing can be used to answer comparison questions such as: What are the similarities and/or differences between A and B, for concepts A and B in a knowledge base (KB). Previous attempts to use a general-purpose analogical reasoner to answer such questions revealed three major problems: (a) the system presented too much information in the answer, and the salient similarity or difference was not highlighted; (b) analogical inference found some incorrect differences; and (c) some expected similarities were not found. The cause of these problems was primarily a lack of a well-curated KB and, and secondarily, algorithmic deficiencies. In this paper, relying on a well-curated biology KB, we present a specific implementation of comparison questions inspired by a general model of analogical reasoning. We present numerous examples of answers produced by the system and empirical data on answer quality to illustrate that we have addressed many of the problems of the previous system.

Introduction

Analogical reasoning and similarity reasoning both rely on an alignment of relational structure, but they differ in that, in analogy, only relational predicates are shared, whereas in literal similarity, both relational predicates and object attributes are shared (Gentner and Markman 1997). As an example: a comparison between an atom and a solar system is considered an analogy, but a comparison between a red door with a red key and a blue door with a blue key is considered a similarity. It has also been argued that the comparison process involves a sophisticated process of structural alignment and mapping over rich complex representations, which can be computationally realized using a Structure Mapping Engine (SME) (Falkenhainer, Forbus, and Gentner 1989).

A practical need for such reasoning arises in education, where the process of making sense of scientific concepts is strongly related to the process of understanding relationships among concepts (Bransford et al. 2000). *Inquire* is an intelligent textbook that embeds key semantic relationships among concepts (Chaudhri et al. 2013a). Students explore those relationships by navigating the textbook and posing a variety of questions. This form of intervention leads to significant learning gains. A key question format supported in *Inquire* is the comparison question: What are the similarities/differences between A and B? For example: What are the differences between alpha-glucose and beta-glucose?

In a previous effort to use analogical reasoning for comparison questions, three problems were found (Nicholson and Forbus 2002): (a) the system answered with too much information, and the salient similarity or difference was not highlighted; (b) analogical inference found some incorrect differences; and (c) some expected similarities were missing. As an illustration of these problems consider the comparison between a Eukaryotic-Cell and a Prokaryotic-Cell[1]. Even though the answer to this question contained an overwhelming amount of information, it did report the salient difference that a Eukaryotic-Cell has a Nucleus but the Prokaryotic-Cell does not. Unfortunately, the system mapped the Nucleus to Cell-Wall, which is incorrect. A major contributor to these problems was a lack of a well-curated KB. Due to errors and omissions in the KB, the overall quality of the results produced by the system suffered. Some weaknesses in the algorithm led to too much information and, sometimes, incorrect information in the answer.

In this paper, we describe our approach for answering comparison questions. Our reasoning algorithms are modeled after analogical reasoning (Falkenhainer, Forbus, and Gentner 1989), but were customized to the questions at hand. We relied on a KB called `KB_Bio_101`, which was created from a biology textbook by domain experts using a state-of-the-art knowledge-authoring system called AURA (Gunning et al. 2010). `KB_Bio_101` contains more than 100,000 axioms, and has been well tested for quality. We explain the algorithms, give examples of questions answered, give experimental data on answer quality, and summarize the open challenges. We consider this work to be large scale because of the size and the complexity of the KB, and a large variety of comparisons that must be drawn.

[1]The actual answer output is available at `www.ai.sri.com/%7Ehalo/public/2014-aaai/` under file name `Q0-NF.pdf`

Answering Comparison Questions

We treat a comparison question as two sub-questions: *What are the similarities between* A *and* B*?* and *What are the differences between* A *and* B*?* The structure of the computation is the same as that used in (Nicholson and Forbus 2002): case construction, candidate inference computation and summarization. We explain our implementation of these steps and discuss how they differ from previous work. Given our set-theoretic approach, we refer to the candidate inference computation step as comparison computation.

Case Construction

The knowledge representation used in AURA has many standard features such as classes, individuals, class-subclass hierarchy, disjointness, slots, slot hierarchy, necessary and sufficient properties, and Horn rules (Chaudhri et al. 2013b). We assume that the relation hierarchy is pre-defined and fixed. The concepts A and B to be compared are represented as classes. Given an individual I, its case $\mathcal{I}$ is its description in terms of its types, slot values, and constraints. To construct a case for a concept, we create its Skolem individual instance and then compute its slot values and constraints.

Because the slot values can be other individuals, the description of I, will contain a set of individuals $S(I)$. $\mathcal{I}$ then has the following three parts: (1) a set $\texttt{types}(I)$ of pairs $\langle individual, type\rangle$, where $individual$ is an individual from $S(I)$, and $type$ is a class that is an immediate type of the $individual$; (2) a set $\texttt{slov}(I)$ of slot value triples $\langle individual, name, value\rangle$ where $individual$ is an individual from $S(I)$, $name$ is a *slot name* and $value$ is an individual from $S(I)$ or a literal value from some pre-defined set $\mathcal{D}$ (the *concrete domain*); and (3) a set of *constraints* $\texttt{sloc}(I)$ on I. Constraints are of the form, $eq\ n\mathrm{R}.\mathsf{C}$, where $eq \in \{=, \leq, \geq\}$, n is a positive integer, R is a relation name, and C is a concept name.

Example 1. *Assume we have the following set of classes in the KB. The indentation implies a subclass relationship (e.g.,* Eukaryotic-Cell *is a subclass of* Cell*.*

Class Hierarchy	
Cell	Ribosome
Eukaryotic-Cell	Chromosome
Prokaryotic-Cell	Cell-Wall
Enzyme	
DNA-Polymerase	
DNA-Polymerase-I	
DNA-Polymerase-II	
DNA-Polymerase-III	

Suppose C1*,* P1 *and* E1 *are Skolem individuals that are respectively instances of* Cell*,* Prokaryotic-Cell *and* Eukaryotic-Cell*. The following case descriptions can be read in an obvious manner (for example,* C1 *has part a ribosome and has a constraint that it has at least one* Chromosome*). The expression* $= 3\text{has-part}.\text{DNA-Polymerase}$ *in the* $\texttt{sloc}(\mathsf{E1})$ *indicates that a Eukaryotic cell has exactly (=) 3 parts that are a* DNA-Polymerase*. This case description has been over-simplified from the knowledge in the KB for explanation purposes.*

$\texttt{types}(\mathsf{C1})$	(C1,Cell) (R1,Ribosome)
$\texttt{slov}(\mathsf{C1})$	$\langle \mathsf{C1}, \text{has-part}, \mathsf{R1}\rangle$
$\texttt{sloc}(\mathsf{C1})$	$\geq 1\text{has-part}.\mathsf{Chromosome}$
$\texttt{types}(\mathsf{P1})$	(P1,Prokaryotic-Cell) (W1,Cell-Wall) (E2,Enzyme)
$\texttt{slov}(\mathsf{P1})$	$\langle \mathsf{P1}, \text{has-part}, \mathsf{W1}\rangle$ $\langle \mathsf{P1}, \text{has-part}, \mathsf{E2}\rangle$
$\texttt{sloc}(\mathsf{P1})$	$\emptyset$
$\texttt{types}(\mathsf{E1})$	(E1,Eukaryotic-Cell) (M1,DNA-Polymerase-I) (M2,DNA-Polymerase-II) (M3,DNA-Polymerase-III)
$\texttt{slov}(\mathsf{E1})$	$\langle \mathsf{E1}, \text{has-part}, \mathsf{M1}\rangle$ $\langle \mathsf{E1}, \text{has-part}, \mathsf{M2}\rangle$ $\langle \mathsf{E1}, \text{has-part}, \mathsf{M3}\rangle$
$\texttt{sloc}(\mathsf{E1})$	$= 3\text{has-part}.\text{DNA-Polymerase}$

A challenge in building the case description is to determine what facts to include. Because E1 is an instance of Cell, by inheritance, we should expect a fact $\langle \mathsf{E1}, \text{has-part}, \mathsf{R2}\rangle$, where R2 is an instance of Ribosome. We use the principle that the case description of a class C should include only those facts that are *local* to C. A fact is local to a class C, if it cannot be derived if all the axioms defined for C were to be removed from the KB. In AURA, this information is computed from a justification system similar to the ones studied by others (Doyle 1979). For every assertion in the KB, it is possible to query the justification system to check if it was directly asserted or derived, and if it was derived, from which concept it was derived. If the only justification for it is the concept C, we assume that the assertion is local to C. In previous work, the problem of what to include in a case was solved by relying on either an arbitrary depth bound on inference or based on what the domain expert chose to keep visible while saving the concept (Nicholson and Forbus 2002). Neither of these approaches was found to be effective.

Comparison Computation

We will illustrate our approach by discussing how we compute differences between two concepts. We will then discuss how this computation generalizes to computation of similarities. The *difference description* of individuals F and G w.r.t. a KB, denoted $\Delta(\mathsf{F}, \mathsf{G})$, is a pair $(\Delta_{\mathsf{G}}(\mathsf{F}),\Delta_{\mathsf{F}}(\mathsf{G}))$ where $\Delta_{\mathsf{G}}(\mathsf{F})$ is defined as the *difference* of F w.r.t. G.

We begin the difference computation by comparing the two Skolem individuals that are instances of the classes being compared. $\Delta_{\mathsf{G}}(\mathsf{F})$ contains a slot value triple from $\mathcal{F}$ if (1) it contains a non-Skolem value that does not also appear in $\mathcal{G}$; (2) if it contains a slot value that cannot be paired with any slot value for the same slot in $\mathcal{G}$. Two values can be paired (2a) if they have exactly the same types in both $\mathcal{F}$ and

$\mathcal{G}$, or if the types of the value in $\mathcal{F}$ subsume the types of the value in $\mathcal{G}$; (2b) if the values have further slot values, then the comparison must be recursively performed. $\Delta_{\mathsf{G}}(\mathsf{F})$ contains a constraint value from $\mathcal{F}$ if (1) the same constraint does not appear in $\mathcal{G}$, and (2) the constraint in $\mathcal{G}$ cannot be derived from some constraints in $\mathcal{F}$. More general constraint simplification techniques can be used for this computation (Abdennadher, Frühwirth, and Meuss 1999). We do not compute the difference between the types of the two individuals because all the types of the individual are shown regardless of whether we are reporting similarities or differences.

Example 2. *Take the question:* What are the differences between a Prokaryotic Cell and a Eukaryotic Cell?*; to answer this we calculate both* $\Delta_{\mathsf{E1}}(\mathsf{P1})$ *and* $\Delta_{\mathsf{P1}}(\mathsf{E1})$.

In the calculation of the difference slot value pairs $\Delta_{\mathsf{E1}}(\mathsf{P1})$*, we note that* $\langle P1, \text{has-part}, E2\rangle \in \texttt{slov}(P1)$*,* $\langle E1, \text{has-part}, \mathsf{M1}\rangle \in \texttt{slov}(E1)$*, but because a* DNA-Polymerase-I *is a subclass of* Enzyme*, these two triples are not a difference. Indeed, prokaryotic cells have as a part an enzyme and eukaryotic cells have as part a DNA polymerase I: because the latter is actually an enzyme, a eukaryotic cell also has a part enzyme, and therefore, it is not a difference. Additionally,* $\langle P1, \text{has-part}, W1\rangle$ *is a difference. We can similarly calculate* $\Delta_{\mathsf{P1}}(\mathsf{E1})$ *resulting in the following difference description:*

	$\Delta_{\mathsf{E1}}(\mathsf{P1})$	$\Delta_{\mathsf{P1}}(\mathsf{E1})$
S	$\langle P1, \text{has-part}, \mathsf{W1}\rangle$	$\langle E1, \text{has-part}, \mathsf{M1}\rangle$ $\langle E1, \text{has-part}, \mathsf{M2}\rangle$ $\langle E1, \text{has-part}, \mathsf{M3}\rangle$
C		$= 3has\ part.DNA\ Polymerase$

We need to be careful to consider the pairing of slot values further than just one level deep. For example, both Eukaryotic-Cell *and* Prokaryotic-Cell *have as* has-part Cytoplasm*; but for a* Eukaryotic-Cell*, the* Cytoplasm *is between a* Nucleus *and* Plasma-Membrane*, which is a difference and, thus, cannot be paired and considered as similar.*

Next we consider how the above computation generalizes to similarities. A relation S for concepts A and B has a similarity if (a) A and B both have a non-Skolem values for S that are equal, or (b) A and B both have Skolem values for S which can be paired (pairing is done similarly as for differences). The similarity computation factors out inherited information and only reports what is specific to a class relative to the nearest common ancestor. The similarity results are grouped and ranked, but there is no need for alignment.

Summarization

The goal of summarization is to present the previous step's computational results in an easy-to-understand manner. More specifically, we want to address the problems faced in the previous work when the salient differences were not always highlighted or too much information was displayed. We use three techniques to support summarization: grouping, ranking, and alignment. We explain these techniques, by considering the system output to *What are the differences between a glycoprotein and a glycolipid?* (see Figure 1).

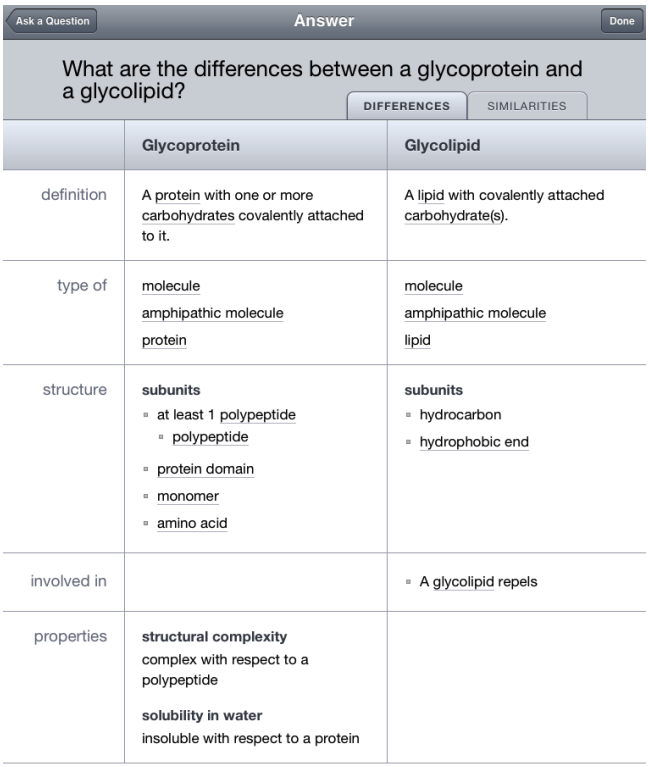

Figure 1: A sample answer to a differences question

The results are organized into a table. The first row shows the human-authored definition of two concepts, and the second row shows all of their superclasses. In earlier versions, we had included only those superclasses that were different, but users preferred to see all the subclasses so that they could easily spot the ones that were different.

Grouping the results Although the KB contains more than 100 slots, only 6–12 slots appear in an answer to most comparison questions. Instead of individually showing the slots, we group the slots into coarser categories. The coarse categories also serve as an abstraction mechanism (Mylopoulos 1998) over the KB's finer-grained distinctions that are necessary for inference.

For example, in Figure 1, the first grouping of slots is labeled as *structure*. This grouping includes slots such as has-part, has-region, possesses, etc. The categories also aim to capture the salient aspects of a difference. In this case, structure corresponds to a core theme in biology expected to know (Reece et al. 2011). The other groups (*involved in*, and *properties*) collect the other slot names. For example, *structural complexity* and *solubility in water* are slots that are both *properties*. These groupings are directly represented in the KB in a relation hierarchy, but the domain experts control the labels. The groupings are also used to answer more specific forms of comparison questions (e.g., What are the structural differences between A and B?). Such specific questions are discussed elsewhere (Chaudhri, Dinesh, and Heller 2013). The values for the slots as well as

the constraints on these slots are presented as a list under the corresponding slot name: the constraint *at least 1 polypeptide* and the value *protein domain* are both listed as a *subunit*.

Ranking the results We further used a ranking of groups and slots to determine the order with which to present them, and within each group the slot values. In the example, the groups have the order $structure < involved\ in < properties$ and for slot names we have $structural\ complexity < solubility\ in\ water$. Such an order is created and maintained by biology domain experts, and can be input to the algorithm as a parameter.

Aligning the results For each grouping and for each slot, we align the slot values. The objective is to show the salient comparisons first. For the *subunits* slot in Figure 1, polypeptide is aligned with hydrocarbon, and protein domain is aligned with hydrophobic end. Because a large number of slot value combinations may exist, it is impractical to determine which comparisons are interesting ahead of time. We devised three different criteria to determine the *interestingness* of comparing a pair of slot values.

The first criterion is based on determining the interestingness of a comparison. We assign an interestingness score to each individual value and then compare an overall score by comparing two values to each other. The interestingness score of a value v, is a number $i(v)$, $0 \leq i(v) \leq 1$, which is computed by using the following heuristics: (1) A slot value is interesting if it is mentioned in the definition of that concept; (2) A slot value is interesting if it is part of a sufficient definition of that concept; (3) A slot value is interesting if it is mentioned frequently in other parts of the KB. After an interestingness score of all values has been considered, we define an interestingness of comparing two values u and v, $score_1(v, u)$, as the average of their individual scores.

The second criterion is based on the syntactic similarity of values. We use the (normalized) Levenshtein string distance between individual names to attach more weight to syntactically similar concepts (Levenshtein 1966). The syntactic similarity matters in a biology textbook as many similar concepts have syntactically similar names. For example, *Nucleus* and *Nucleoid* are syntactically similar and score high. In general, for a pair of values (v, u) the syntactically similarity $score_2(v, u)$ yields a value between 0 and 1.

The third criterion is based on the *Semantic similarity of values*. Values that are semantically similar should be grouped. For example, *Prokaroytic Cell* and *Eukaroytic Cell* are semantically similar because they are both subconcepts of *Cell*, and hence, *Prokaryotic Cell* should rather be paired with *Eukaryotic Cell* than, for example, with *Nucleus*. The score $score_3(v, u)$ thus assigned is also between 0 and 1.

We compute the overall score by taking an average of $score_1(v, u)$, $score_2(v, u)$ and $score_3(v, u)$. Using this scoring function, we can define a score of a particular alignment of values as the sum of the individual alignments. The number of possible alignments is proportional to the number of values within each slot category. Finding the best alignment is an optimization problem, and we solve it using a best-first search (Russell et al. 1995), and show the best alignment.

Example 3. *In the answer in Figure 1, we can see that polypeptide and hydrocarbon are paired. Polypeptide and hydrocarbon are both organic molecules while hydrophobic end is a region. It thus makes sense not to pair polypeptide with hydrophobic end. Similar arguments can be made as to why monomer was not aligned to either hydrocarbon or hydrophobic end. Protein domain and hydrophobic end both have a superclass* Region *that makes this pairing sensible.*

Experimental Feedback

The system went through a series of studies with end-users that included both teachers and students. A small-scale educational usefulness study that included comparison questions, but focused on student learning gains instead of quality of answers, has been previously published (Chaudhri et al. 2013a). Our focus in the current paper is on the answer quality of comparison questions.

An ideal evaluation would have compared the outputs from the current system with the outputs from (Nicholson and Forbus 2002). We loaded the current KB into the prior system, but since the KB is much bigger it crashes that system. If we pose the current set of questions on the prior system with the KB at that time, most of the questions fail due to lack of knowledge. A fair comparison between the two systems requires a non-trivial amount of additional work.

Our evaluation goal was to test the cognitive validity of the techniques considered here. More specifically, we test if we can we capture the salient similarities and differences and rank them in an order that matches the user's understanding. To test this hypothesis, we assembled a suite of 158 comparison questions uniformly spread over the first eleven chapters of the textbook. Each answer was rated by a biologist who encoded the knowledge and a biology teacher. A subset of answers was graded by end-user students to ensure that the student scores were comparable to the scores of encoder biologists and teacher biologists. An answer was considered of high quality if it included salient similarities and differences at the top. Overall, 97% of the questions produced an answer. From this set, 57% of the questions were considered of very high quality with no major issues. A score of 57% indicates progress but it is not as good as a score of 90-100% that a human is capable of. Many issues were easily correctable, e.g., some questions suffered from the natural language presentation, while others suffered due to KB gaps. To offer deeper insight, we consider sample answers and then identify the cognitive issues and challenges.

Example Answers

For each question below, we indicate the identifier, the chapter title, whether it asks for a similarity or difference, the concepts being compared, and the answer. In question Q1 below, the question identifier is Q1; it is from a chapter on Carbon; it is a difference question over Aldehyde and Alcohol, followed by the answer.[2]. We summarize because, to convey the breadth of issues, we need multiple questions, and we could not include all the outputs in the paper. The

[2]We have summarized the answers; for the actual output, see www.ai.sri.com/%7Ehalo/public/2014-aaai

summarization is based on the salient result reported by the system (i.e., reported at the top). Wherever that is not the case, it is pointed out.

Q1. Carbon. Difference. Aldehyde. Alcohol. An aldehyde has a carbonyl group as subunits while alcohol has hydroxyl group as subunit. Alcohol has hydrogen and oxygen as subunits and possesses covalent bonds

In Q1, while computing the difference, we are correctly told that an aldehyde has a carbonyl group while alcohol has a hydroxyl group. We are further told that alcohol has hydrogen and oxygen as subunits, and possesses a polar covalent bond. This illustrates a limitation of the tabular structure for presenting the answers as well as a challenge for natural language generation (NLG). The concept representation of alcohol in the KB shows that the polar covalent bond is between hydrogen and oxygen, thus, relating three different entries that are values of different slots: two of which as subunits (i.e., hydrogen and oxygen), and one under possesses (covalent bond). This kind of relationship is very difficult to capture in a tabular structure. An alternative is to synthesize English sentences that capture such relationships. Although AURA includes a NLG capability (Banik, Kow, and Chaudhri 2013), we have not yet applied it in such situations.

Q2. Cell. Difference. Rough Endoplasmic Reticulum. Smooth Endoplasmic Reticulum. The rough endoplasmic reticulum has ribosomes in its structure while the smooth endoplasmic reticulum does not.

In the answer to Q2, we are told that a rough endoplasmic reticulum contains ribosomes while a smooth endoplasmic reticulum does not. This answer raises several issues. First is the problem of empty entries in the table. In the answer, we are shown that rough endoplasmic reticulum has a bound ribosome as a subunit, but the corresponding entry for smooth endoplasmic reticulum is empty. We must assume that the absence of the value implies that the smooth endoplasmic reticulum does not have a bound ribosome. But, in many cases, omissions exist in the KB, and an absence of a value can be confusing to the user. The second problem is how to arrive at an appropriate ranking. Whenever we have a comparison such that only one of the concepts has the value for a slot, but the other one does not, we rank it lower. In this example, this difference gets ranked the lowest in spite the fact that the bound ribosome is mentioned in the text definition of both of these concepts. This illustrates a complex interaction between two competing heuristics, which can be quite challenging to work out across a large number of examples.

Q3. Cellular Respiration. Difference. Citric Acid Cycle. Calvin Cycle. Calvin Cycle is an anabolic pathway and produces sugar, whereas Citric Acid Cycle is a catabolic pathway and produces carbon dioxide.

Q4. Photosynthesis. Difference. Light Reaction. Calvin Cycle. Light reaction produces ATP and consumes ADP; Calvin Cycle produces ADP and consumes ATP.

In Q3 and Q4, we compare a Calvin cycle with citric acid cycle and light reaction. In the answers to both of these questions, the salient differences are derived by comparing their respective inputs and outputs. The main difference between the answers to these two questions is in the way that the differences are aligned and ranked. In the tabular presentation of the answer comparing Calvin cycle and light reaction, ADP and ATP are aligned with each other based on their similarity scores: these differences are properly identified as salient and shown first.

Q5. Cell Communication. Difference. Signal Transduction. Cell Signaling. Signal transduction is a step of Cell Signaling.

In Q5, we compare signal transduction and cell signaling. Here, cell signaling happens to have a series of steps in which signal transduction happens to be one of them. In the tabular presentation of the answer, all the steps of cell signaling are shown, and signal transduction is highlighted in bold to emphasize the salience of the comparison.

General Lessons and Open Challenges

The previous section illustrated comparisons between entities and processes. Although a question's syntactic form is the same, its instantiation for different concepts presents diverse issues that must be accounted for. Generating answers for such questions is more complex than simply returning a short phrase, which is typical of state of the art question answering systems (Ferrucci et al. 2010; Voorhees, Harman, and others 2005). We next discuss general lessons and challenges in presenting answers to comparison questions that we distilled from user feedback.

The first issue is whether the comparison questions can be separated into similarities and differences questions. Often, the users wanted to see the differences and similarities together, or they expected a similarity to appear, but computationally it was a difference. It is often not easy to separate both, e.g., for the comparison between a eukaryotic and prokaryotic cell, both have a cytoplasm (a similarity), but for prokaryotic cell, the cytoplasm is between the nucleus and the cell membrane (a difference). Our solution is to show both similarities and differences regardless of the question. The two answers are shown in different tabs in the user interface (see Figure 1). An alternative summarization strategy is to show similarities and differences together in one single display. Investigating this is a topic for future research.

The users uniformly liked the tabular presentation, but, as we saw in the answer to Q1, such a presentation is incomplete. Most concepts in `KB_Bio_101` can be several levels deep. That level of depth of knowledge is not leveraged and presented by the current system. Most of the comparisons of interest for the educational application may be shallow, but we do not have any well-formed evidence to support or refute this observation. Considering comparisons that leverage deeper aspects of concept representations and presenting them in the answers is open for future research.

We make judicious use of NLG, e.g., we render the steps of a process in English. This enhances the usability of the output as compared to using the KB names of steps. A natural question is whether we should move away from a tabular presentation and present the comparisons textually. The users have preferred minimal use of English sentences so

that the answers provide a different presentation of information than is already available in textual form in the textbook. Generating good sentences is still a challenge for NLG systems (Banik, Gardent, and Kow 2013). Perhaps the ideal solution is in the middle; further exploration is future research.

When information about two concepts is put next to each other, it can create surprises. An empty entry was an instance of this problem. In some cases, an empty entry may be seen because the textbook itself does not directly mention a fact. For example, when we compare a ribosome and a chromosome, the system responds by saying that the function of a ribosome is protein synthesis, but the corresponding function for a chromosome is left empty. The textbook does not explicitly offer that information, and therefore, it is not present in the KB. This kind of textbook deficiency is rarely caught and is easily forgiven by human readers, but when it is noticed through a computational tool such as the one we presented, the user feedback is extremely harsh. This observation implies that comparison questions can serve as a powerful tool during knowledge engineering to test and debug the KB as well as to improve the textbook.

Comparison to Related Work

Several subprocesses constitute analogical thinking (Keane, Ledgeway, and Duff 1994): representation, retrieval, mapping, adaptation and induction. We have focused on the representation and mapping steps, and to a limited degree on induction. The techniques used in summarization (e.g., grouping the relations into categories) can be viewed as limited form of induction. Retrieval of an appropriate base case and adapation are not required; indeed, the concepts to be compared are provided and the primary goal is to improve student comprehension.

Focusing on the mapping step, we note that during analogical mapping, two computations occur: (1) corresponding concepts in both domains are matched; and (2) a portion of the conceptual structure of one domain is transferred into the other. In our matching process only those concepts that have a common relation are matched. For example, a Nucleus will be matched to Nucleoid only if both are in a has-part relationship to the two cells being compared. In general analogical reasoning, the two concepts could be matched even if they are in different relationships to the concepts being compared. Furthermore, for comparison computation, there is no need to transfer the conceptual structure. This makes our algorithm more restricted than general analogical reasoning, but customized to the task of performing comparisons.

Different approaches for analogical mapping can be understood in terms of constraints on information, behavior and hardware (Keane, Ledgeway, and Duff 1994). Information constraints can be further sub-categorized into structural constraints, similarity constraints and pragmatic constraints. Just like SME (Falkenhainer, Forbus, and Gentner 1989) and Analogical Constraint Mapping Engine (ACME) (Holyoak and Thagard 1989), our approach relies on the following structural constraints: make matches only between entities of the same type, exploit structural consistency and favor a systematic set of matches. Indeed, SME relies on the principle of *systematicity*: based on the presence of higher-order structures to describe relations, it favors mappings that map larger such structures. In contrast, our representation is flat and higher-orderness is handled by reification: we handle systematicity by preferring matches with *more* similarities.

Further, we use similarity constraints that are based on both syntactic and semantic similarity. The notion of interestingness considered by us can be viewed as a pragmatic constraint. Two kinds of behavioral constraints have been considered in the prior work (Keane, Ledgeway, and Duff 1994): working memory constraints and background knowledge. Both SME and ACME deal with working memory constraints in a heuristic manner. In contrast, our case construction considers only *local information*, thus scoping the size of the cases, and controlling the working memory requirements. Even though we do not use background knowledge to validate the computed inferences, we do use the class taxonomy and slot hierarchy for summarizing the results. Our work does not consider hardware constraints.

Our work overlaps with work on using analogies for biologically inspired designs (Helms and Goel 2014). They also use high level categories such as functions and performance criteria for organizing the similarities and differences and use one unified tabular display for presentation. However, the similarities are manually determined by humans and the goal is to find analogical concepts, whereas in our approach the concepts to compare are given.

In the knowledge representation and reasoning literature, techniques to compute differences have been explored (Brandt, Küsters, and Turhan 2002; Baral and Liang 2012), but no large-scale experimentation has been attempted to verify that the results that are cognitively valid.

Summary

We presented a system to answer comparison questions, inspired by a general model of analogical processing, but offering several novel features. First, we leveraged KB_Bio_101, a well-curated KB that has undergone extensive quality control. Second, we used a case-construction scheme that focuses on information local to a concept, ensuring salient differences. Third, our computation accounted for class hierarchy and the graph structure of the case description, and could simplify the constraints. Finally, we customized our implementation to the specific domain and devised a presentation scheme that used judicious grouping, ranking, and alignment of the results. The resulting system is a substantial advance over the previous implementation for this task (Nicholson and Forbus 2002). Our implementation can give good answers to comparisons over concepts ranging from chemicals, molecules, cell organelles, to processes. This experience highlights many challenging research topics that must be addressed to continue to capture the cognitive processes that underlie the human ability to perform comparisons, focus on salient aspects, and succinctly explain results. Because the implementation has been embedded in an intelligent textbook that has been proven to improve student learning, we expect that further progress on this topic will have substantial impact on the use of computational tools based on analogical reasoning in education.

Acknowledgments

This work has been funded by Vulcan Inc. and SRI International. We gratefully acknowledge the efforts of the KB development team, the biology teachers and the software engineering team who contributed to this effort.

References

Abdennadher, S.; Frühwirth, T.; and Meuss, H. 1999. Confluence and semantics of constraint simplification rules. *Constraints* 4(2):133–165.

Banik, E.; Gardent, C.; and Kow, E. 2013. The KBGen challenge. In *ENLG 2013 : 14th European Workshop on Natural Language Generation*.

Banik, E.; Kow, E.; and Chaudhri, V. K. 2013. User-controlled, robust natural language generation from an evolving knowledge base. In *ENLG 2013: 14th European Workshop on Natural Language Generation*.

Baral, C., and Liang, S. 2012. From knowledge represented in frame-based languages to declarative representation and reasoning via asp. In *KR*.

Brandt, S.; Küsters, R.; and Turhan, A.-Y. 2002. Approximation and difference in description logics. In *Proc. of KR-02*. Citeseer.

Bransford, J. D.; Brown, A. L.; Cocking, R. R.; et al. 2000. *How people learn*. National Academy Press Washington, DC.

Chaudhri, V. K.; Cheng, B.; Overholtzer, A.; Roschelle, J.; Spaulding, A.; Clark, P.; Greaves, M.; and Gunning, D. 2013a. *Inquire Biology*: A textbook that answers questions. *AI Magazine* 34(3).

Chaudhri, V. K.; Heymans, S.; Wessel, M.; and Tran, S. C. 2013b. Object-oriented knowledge bases in logic programming. In *Technical Communication of International Conference in Logic Programming*.

Chaudhri, V. K.; Dinesh, N.; and Heller, C. 2013. Conceptual models of structure and function. In *Second Annual Conference on Advances in Cognitive Systems*.

Doyle, J. 1979. A truth maintenance system. *Artificial intelligence* 12(3):231–272.

Falkenhainer, B.; Forbus, K.; and Gentner, D. 1989. The structure-mapping engine: Algorithm and examples. *Artificial Intelligence* 41:1–63.

Ferrucci, D.; Brown, E.; Chu-Carroll, J.; Fan, J.; Gondek, D.; Kalyanpur, A. A.; Lally, A.; Murdock, J. W.; Nyberg, E.; Prager, J.; et al. 2010. Building Watson: An overview of the DeepQA project. *AI magazine* 31(3):59–79.

Gentner, D., and Markman, A. B. 1997. Structure mapping in analogy and similarity. *American Psychologist* 52:45–56.

Gunning, D.; Chaudhri, V. K.; Clark, P.; Barker, K.; Chaw, S.-Y.; Greaves, M.; Grosof, B.; Leung, A.; McDonald, D.; Mishra, S.; Pacheco, J.; Porter, B.; Spaulding, A.; Tecuci, D.; ; and Tien, J. 2010. Project Halo update: Progress toward digital Aristotle. *AI Magazine*.

Helms, M., and Goel, A. 2014. The four-box method of analogy evaluation in biologically inspired design. In *ASME International Design Engineering Technical Conferences and Computer and Information Engineering Conference*.

Holyoak, K. J., and Thagard, P. 1989. Analogical mapping by constraint satisfaction. *Cognitive science* 13(3):295–355.

Keane, M. T.; Ledgeway, T.; and Duff, S. 1994. Constraints on analogical mapping: A comparison of three models. *Cognitive Science* 18(3):387–438.

Levenshtein, V. I. 1966. Binary codes capable of correcting deletions, insertions and reversals. In *Soviet Physics Doklady*, volume 10, 707.

Mylopoulos, J. 1998. Information modeling in the time of the revolution. *Information Systems* 23(3):127–155.

Nicholson, S., and Forbus, K. D. 2002. Answering comparison questions in SHAKEN: A progress report. In *AAAI Spring Symposium on Mining Answers from Texts and Knowledge Bases*.

Reece, J. B.; Urry, L. A.; Cain, M. L.; Wasserman, S. A.; Minorsky, P. V.; and Jackson, R. B. 2011. *Campbell biology*. Boston: Benjamin Cummings imprint of Pearson.

Russell, S. J.; Norvig, P.; Canny, J. F.; Malik, J. M.; and Edwards, D. D. 1995. *Artificial intelligence: a modern approach*, volume 74. Prentice hall Englewood Cliffs.

Voorhees, E.; Harman, D. K.; et al. 2005. *TREC: Experiment and evaluation in information retrieval*, volume 63. MIT press Cambridge.