

# Answering Biology Questions using Textual Reasoning

Peter Clark, Phil Harrison (Vulcan Inc), Niranjana Balasubramanian (Univ Washington)  
{peterc,philipha}@vulcan.com, niranja@cs.washington.edu

## 1. Introduction

As part of Project Halo, we are seeking to build an (iPad based) "knowledgeable textbook" about biology that users can not only browse, but also ask questions to and get reasoned or retrieved answers back. While our previous work has relied on a hand-crafted knowledge base, we have a new effort this year to add a textual question-answering (QA) module that will answer some classes of questions directly from the textbook itself, combining techniques from NLP, textual entailment, and paraphrasing together. Specifically, we are interested in answering questions where the required knowledge is difficult to encode formally, but there is a surface-level answer expressed explicitly in text, for example the questions:

- What are the most common lipids in membranes?
- What limits the size of cells?
- What molecule does aquaporin transport?

have explicit answers in the textbook from the sentences:

- The most abundant lipids in most membranes are *phospholipids*.
- *The logistics of carrying out metabolism* sets limits on cell size.
- Aquaporins selectively conduct *water molecules* in and out of the cell.

Identifying such answers is not straightforward, though, and requires a linguistic analysis of both the question and book sentences, and the use of general lexical and world knowledge, in order to be reliable. For instance, the above examples benefit from knowing that (a) "common" and "abundant" are synonyms; (b) "set limits on" and "limits" are paraphrases, and "the size of cells" and "cell size" have similar meaning; and (c) "conducts" and "transports" can be synonyms, and that a water molecule is a molecule.

While we are using a somewhat standard QA architecture for this task, there are several novel challenges that we face: First, we have a highly domain-specific and closed corpus, and the requirement that all answers must come from within the book (although we can use other resources to help locate those answers). Second, our question set is diverse and includes questions requiring both lookup and simple inference. Third, the cost of providing wrong answers is high. Together these raise the importance of using simple "textual reasoning" as well as retrieval to overcome sparsity and limited redundancy, as well as creating good validation and confidence models for the answers.

## 2. Approach

### Pre-Runtime: Resource Construction

While one can think of this style of QA as "searching the textbook" for answers, in practice, and for efficiency, it involves searching resources lexically derived from the textbook reflecting the "facts" within it. We are currently using two such textbook-derived resources:

- **Parses:** A parse-based logical form (LF) representation of the book sentences, e.g.,:  
From "*Metabolism sets limits on cell size*":  
(S (SUBJ ("metabolism")) (V ("set")) (SOBJ ("limit" ("on" ("size" (MOD ("cell"))))))))
- **Triples:** A set of arg1-predicate-arg2 triples extracted via a chunker applied to the book sentences, using Univ. Washington's ReVerb system, e.g.,  
From "*Free ribosomes are suspended in the cytosol and synthesize proteins there.*":  
["ribosomes"] ["are suspended in"] ["the cytosol"]

### 1. Question Analysis

Given a question, question analysis involves identifying its intent and converting it into suitable form(s) for querying the textbook resources. While each resource may have its own idiosyncratic query mechanism, much of the general question analysis mechanism can be shared. In some cases, the query to

resources will be a simple transformation of the original question. However, in other cases, more sophisticated processing is needed, and requires writing simple "textual QA methods" that convert a question into one or more queries to the text. The resulting methods form a thin software layer that mediate between the question and the engines that query the text resources.

## 2. Answer Generation

Although the resources have different forms, the query mechanisms are all performing essentially the same task, namely seeing if some (declarative representation of) a textual query is present in the resource. Given the variability in language, and the low redundancy in the textbook (as opposed to the Web), many query patterns are unlikely to be found verbatim in the book. However, a pattern may be inferred to match using simple lexical and world knowledge. We are drawing this knowledge from four resources: WordNet, ParaPara (a paraphrase database from Johns Hopkins), the DIRT paraphrase database, and our hand-built formal biology KB itself.

To search the parses (LFs), we search for sentence parses subsumed by the question parse. The answer is then the phrase in the sentence corresponding to the "what" node in the question. In the simplest case this is just structure matching, but more commonly we use textual entailment, exploring rewrites of the sentence using the synonym and paraphrase resources above. For example,

Q. Which proteins help move molecules through the membrane?

can be answered from the (parse of the) book sentence:

- *Channel proteins* facilitate the passage of molecules across the membrane.

using knowledge that **IF** X facilitates Y **THEN** X helps Y, that "passage" is a nominalization of "move", and that "through" and "across" are approximate synonyms, drawn from the above knowledge resources.

To search the ReVerb triples, a question is converted into a search query over the extractions by matching it against a small set of hand-built question patterns, each with a corresponding extraction query. The query is then run against the extraction database. Again, the linguistic resources above can be employed to improve recall.

## 3. Answer Aggregation and Scoring

A particular answer may be supported by multiple sentences, each sentence may have needed different paraphrases with different confidences to conclude the answer, and each textbook resource (parses, triples) may contribute to the same answer. To derive an overall confidence in an answer, we are using machine learning techniques to build a confidence model computed from various answer features (number of supporting sentences, paraphrases used, resource used, section of the book, syntactic properties of the answer, etc.). To obtain training data, a group of biology students involved in the project are scoring the system-generated answers.

## 3. Status, Discussion, and Plans

This work is still in its early stages (the system currently answers about 10% of our question corpus, obtaining about 70% accuracy on those 10%), but provides an exciting complement to the logic-based QA that we have been using in the past. Its strengths are that it is cheap, scalable, and can answer some difficult-to-represent questions by working directly at the textual level. The primary challenge is less "Does it work?", but rather "Can it be made to work reliably enough?". There is noise at each stage of the process, and the task is thus to create something robust from somewhat brittle individual pieces.

We are expanding the system in various ways, including using additional biology texts to help locate/validate answers in the target textbook, validating answers against a Web-based corpus, and exploring partial matching techniques as well as full textual entailment for QA. We are also seeking to make the pre-runtime databases derived from the textbook be more of a unified "textual knowledge base" in its own right, by using the same QA technology to merge evidence together from the individual parses and extractions. Our plan is that this can then be crowd-maintained, corrected, and validated by biologists, potentially resulting in better QA performance of the overall system.