International Review of Research in Open and Distributed Learning

Instructor-Aided Asynchronous Question Answering System for Online Education and Distance Learning

Dunwei Wen, John Cuzzola, Lorna Brown et Dr. Kinshuk

Citer cet article

Wen, D., Cuzzola, J., Brown, L. & Kinshuk, D. (2012). Instructor-Aided Asynchronous Question Answering System for Online Education and Distance Learning. *International Review of Research in Open and Distributed Learning*, *13*(5), 102–125. https://doi.org/10.19173/irrodl.v13i5.1269

Résumé de l'article

Question answering systems have frequently been explored for educational use. However, their value was somewhat limited due to the quality of the answers returned to the student. Recent question answering (QA) research has started to incorporate deep natural language processing (NLP) in order to improve these answers. However, current NLP technology involves intensive computing and thus it is hard to meet the real-time demand of traditional search. This paper introduces a question answering (QA) system particularly suited for delayed-answered questions that are typical in certain asynchronous online and distance learning settings. We exploit the communication delay between student and instructor and propose a solution that integrates into an organization's existing learning management system. We present how our system fits into an online and distance learning situation and how it can better assist supporting students. The prototype system and its running results show the perspective and potential of this research.

Instructor-Aided Asynchronous Question Answering System for Online Education and Distance Learning

Dunwei Wen, John Cuzzola, Lorna Brown, and **Dr. Kinshuk**
Athabasca University, Canada

Abstract

Question answering systems have frequently been explored for educational use. However, their value was somewhat limited due to the quality of the answers returned to the student. Recent question answering (QA) research has started to incorporate deep natural language processing (NLP) in order to improve these answers. However, current NLP technology involves intensive computing and thus it is hard to meet the real-time demand of traditional search. This paper introduces a question answering (QA) system particularly suited for delayed-answered questions that are typical in certain asynchronous online and distance learning settings. We exploit the communication delay between student and instructor and propose a solution that integrates into an organization's existing learning management system. We present how our system fits into an online and distance learning situation and how it can better assist supporting students. The prototype system and its running results show the perspective and potential of this research.

Keywords: Automated question answering; natural language processing; information retrieval; LMS; distance education; online learning

Technology tools such as wikis, blogs, and podcasts can foster student interaction in online learning (Beldarrain, 2006). However the focus of instructional design should be on customization of content where the learner's needs become the center of attention (Reigeluth, 1999). Learners of all age groups are sensitive to the applicability of content and ease of use of an online learning portal. Denvir, Balmer, and Pleasence (2011) discovered that people aged 18-24, although they have grown up in the digital age, have reduced success in finding information online. Their results indicate that young people are less likely to utilize the Internet for obtaining advice, possibly due to frustration they experience when searching on the Internet. However, online learning has broadened the accessibility of education by the reduction of time-zone, employment commitment, and family obligation constraints, particularly with students who have left the traditional learning institutions. Nonetheless, the transition from traditional to online delivery isn't trivial for the student. Likewise, perceived usefulness and ease of use are critical factors in determining a teacher's acceptance and use of an educational portal (Pynoo, Balmer, van Braak, Duyck, Sijnave, & Duyck, 2012).

Question answering (QA) technology may be part of the solution. QA aims to automatically answer some of these questions in a way that the current search engines such as Google and Bing do not. A QA system attempts a deeper understanding of a query such that the retrieval of text/documents best answers the student's question. This is particularly difficult in the context of natural language where ambiguity in meaning is inherent in speech. Consider the following example query (Q) and the possible candidate answers (CA) available for selection.

Q*: Should I keep my dog away from eating flowers?*

CA1: More than 700 plants have been identified as producing physiologically active or toxic substances in sufficient amounts to cause harmful effects in animals.[1]

CA2: Poisonous plants produce a variety of toxic substances and cause reactions ranging from mild nausea to death. [2]

CA3: Certain animal species may have a peculiar vulnerability to a potentially poisonous plant. [3]

CA4: A hot dog is one of my favorite things to eat.

Removing stop words and focusing on the core words of the query *dog*, *away*, *eating*, and *flowers*, one can see that CA4 would be the most likely answer when strictly a bag-of-words approach is used since none of the core words appear elsewhere. However, clearly CA4 is also the least desirable. Employing further mechanical techniques such as stemming and lemmatization would also strengthen the choice of CA4 as the best answer (the word "eating" would be stemmed to "eat").

---

[1]   www.humanesociety.org/animals/resources/tips/plants_poisonous_to_pets.html

[2]   www.calgaryhumane.ca/document.doc?id=78

[3]   www.treehouseanimals.org/Tree%20House%20Site-E-CPP.htm

This leads to the need of more thorough natural language processing (NLP). While many NLP techniques such as sentence segmentation, tokenization, name entity recognition, chunking, POS tagging, and syntactic parsing become necessary parts of QA systems, deeper NLP techniques including syntactic parsing and semantic role labeling (SRL) have increasingly attracted the attention of QA researchers. Semantic role labeling (SRL) is one method of achieving a deeper understanding of the query to better match the question with the candidate answers. Role labeling maps parts of a sentence to abstract themes (frames) and their supporting metadata (frame elements).
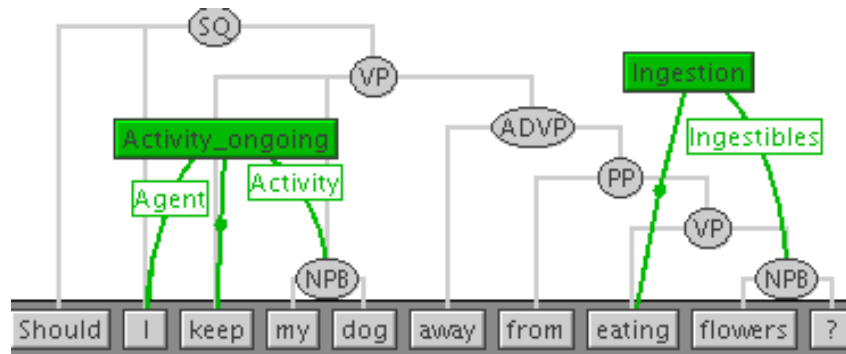

Figure 1. An example of SRL.

Figure 1 demonstrates the idea of SRL using the *FrameNet* (Baker, Fillmore, & Lowe, 1998) lexical semantic database. Two frames (concepts) have been identified in the query: [activity_ongoing] and [ingestion]. The frame elements provide supporting data for the frame. For example, flowers have been identified as the entity being ingested. However, contrary to intuition, these deeper NLP processing techniques are not always of much help as we will discuss below. Further research into the effective use of them in QA is needed. Moreover, as we can imagine, syntactic and semantic processing are much more computationally intensive (requiring more computing time and resources for performing the tasks) than other lower level NLP processing tasks and thus impose a big burden to online or synchronous QA that is supposed to provide answers to a question immediately.

Our current research is focused on these two problems and aims to reach our own solutions to produce more efficient and relevant answers particularly in an educational setting. To face the intensive computing challenge, we exploit the communication latency between student and instructor in online and distance education environments, and propose an asynchronous QA framework that makes the deep NLP analysis workable and acceptable in reality. This paper complements our recent work in exploiting semantic roles for asynchronous question answering (Wen, Cuzzola, Brown, & Kinshuk, 2012), with emphasis on the student and instructor interaction through the QA interface. We extend our contribution through new evaluation results using a large commercial corpus and compare the results with our alternative solution further validating our proposed technique.

An Asynchronous QA System

In this section, QA's processing time issue when integrating deep NLP is examined. We begin by discussing the time complexity challenge then present an overview of the major parts of a QA system and finally offer the particulars of our proposed QA prototype system. We describe how our implementation of such a system can fit into an online learning situation despite this time obstacle.

The Trouble with Time

Modern QA systems incorporate NLP such as syntactic and semantic query analysis in an attempt to find the most relevant answers. Syntactic techniques examine the individual elements and structure of the sentence. These strategies include part-of-speech (POS) tagging where words are classed into their grammatical categories – noun, pronoun, verb, adverb, adjective, and so forth. Parse trees extend POS tagging with a structural representation of the sentence. In contrast, semantic techniques attempt to derive the context or meaning of a sentence. Such strategies include named entity (NE) recognition where parts of a sentence are generalized as entities such as country, person, date, temperature, weight, and so forth. Other more recent strategies involve labeling semantic roles and collecting them as reusable resources, such as *PropBank* (Palmer, Gildea & Kingsbury, 2005) that centers around the verb of the sentence and *FrameNet* which deconstructs the sentence into concepts (semantic frames) and supporting frame elements. The NLP research community has developed many very useful software tools to make the above techniques accessible and applicable to researchers for constructing new NLP projects. Such powerful tools include *ASSERT* (Pradhan, Ward, Hacioglu, & Martin, 2004), *SEMAFOR* (Das, Schneider, Chen, & Smith, 2010), and *Shalmaneser* (Erk & Pado, 2006), which are readily exploitable to those who can mitigate the computational time obstacle.

Table 1

Speed Comparison between ASSERT/SEMAFOR/Shalmaneser per 1,000 Sentences

ASSERT (PropBank)	SEMAFOR (FrameNet)	Shalmaneser (FrameNet)
19 Minutes	30 Minutes	77 Minutes

Table 1 shows the observed processing time required to semantically parse 1,000 sentences from the Reuters 21,578 corpus using PropBank and FrameNet. The speed comparisons were done using the software of ASSERT, SEMAFOR, and Shalmaneser on a 2.4 GHz Intel Core2 Duo with 8 GB RAM on a 64-bit Linux OS.

This overhead makes the use of semantic role labeling troublesome in synchronous QA where near-instantaneous answers are expected. Precomputation of the corpus in advance may avoid lengthy delays at the time of the query. The aforementioned Reuters corpus consisted of a total of 104,410 sentences which in the context of Table 1 may be feasible for

preprocessing and storage. However, a large corpus of millions of lines (and larger) makes beforehand labeling unpractical. Furthermore, this option is unavailable for dynamically changing content such as that accessible through search engines, Wikipedia, online forums, and similar World Wide Web artifacts.

Our research investigates the offline interaction between student and instructor that inherently involves a response delay between the time a question is asked and the time of an expected answer. It is within this delay where we propose a QA system that can benefit from semantic role labeling while side-stepping the time complexity problem.

The Components of a QA System

The components of a question answering type system are commonly comprised of four stages: *question analysis*, *query generation*, *search*, and *filtering/ranking*. A syntactic and semantic analysis of the question is performed to generate suitable queries for the search. The returned results also undergo syntactic and semantic evaluation to determine their relevancy with respect to the question and their ranking with respect to each other. Non-relevant answers are discarded (filtered). We utilized the open source framework *OpenEphyra* which provides the four stage pipeline in a modular and extensible implementation (Schlaefer, Gieselmann, & Sautter, 2006). OpenEphyra marries *INDRI's* very fast search engine for large lexicons with various syntactic capabilities that include part-of-speech tagging, sentence detection, and named entity recognition through the *OpenNLP* package (http://opennlp.sourceforge.net). Additionally, a lexical database of related concepts known as *WordNet* (Miller 1995) is also available for syntactic use. PropBank semantic analysis is performed through the ASSERT software. We are currently developing an interface module to either Shalmaneser or SEMAFOR that would give OpenEphyra the ability to perform subtler semantic analysis through FrameNet.
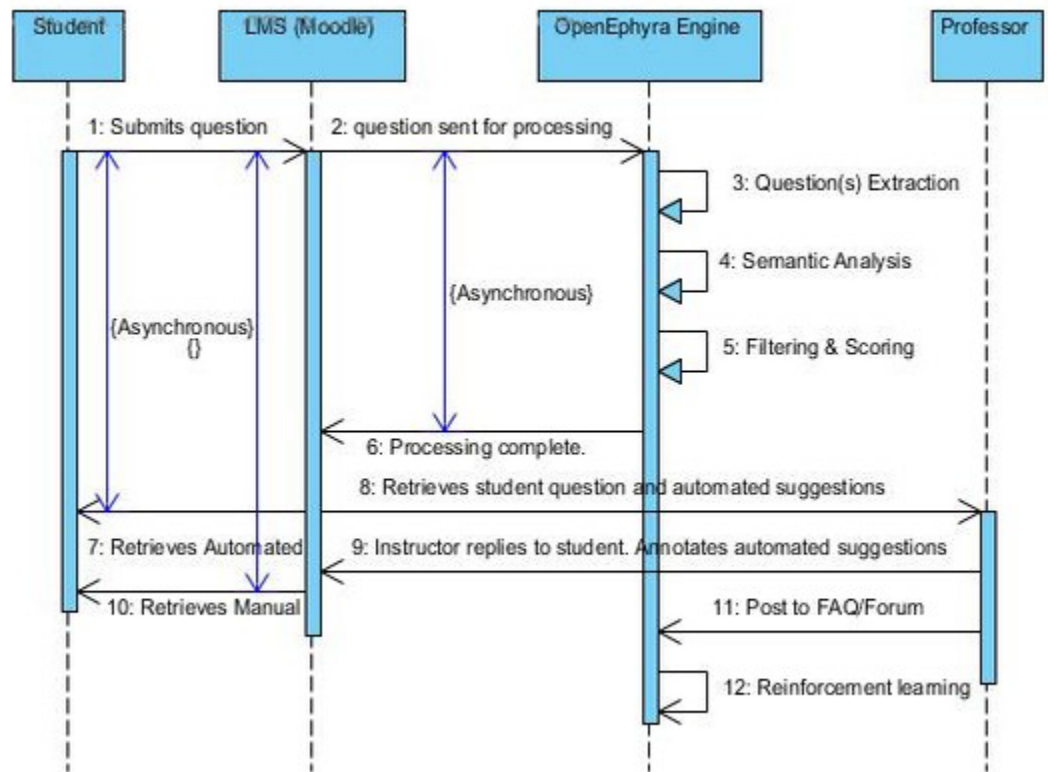
Figure 2. Asynchronous semantic QA model (Wen, D., Cuzzola, J., Brown, L. & Kinshuk, 2012).

Figure 2 outlines the proposed model. The primary actors involve the student, the learning management system (LMS), the OpenEphyra framework, and the course instructor (professor). A student first submits a question to the course instructor/tutor through an interface made available through the organization's learning management system. This allows for seamless, integrated and familiar access for the student thus encouraging its use. We use *Moodle* (http://moodle.org) as the LMS, a popular choice for a significant number of educational organizations. Once a student's question is posted, the OpenEphyra framework begins the process of question extraction, semantic analysis, and finally filtering and scoring of the top ten results (Steps 3-5). It is through this phase of the process that the proposed asynchronous solution is required. The time complexity of these three steps significantly exceeds the patience of any user expecting results as quickly as the traditional synchronous search engine (see Table 1). However, since this communication model between the time a student submits the question and the instructor's response is anticipated to be delayed in an asynchronous learning environment, the problem of time complexity can be mitigated. Once the processing is complete, a ranked top 10 result, similar in output to that of a search engine, awaits the student and the course instructor in their LMS mailboxes for retrieval (Steps 7 and 8). The students may now investigate the automated results for potential answers to their questions while they wait for the instructors' feedback. The instructor receives the student's query as well as the automated suggested answers. The

instructor can give direct feedback to the student, annotate or modify the automated results list, and even give preference to specific result entries over other less-relevant automated suggestions. This instructor reply then awaits the student in his or her LMS mailbox as a follow up to the automated suggestions. The instructor may also decide that the revised questions/answers may be of use to other students in the class and can direct the LMS to post this information in the course FAQ or forum. Lastly, the model includes a reinforcement learning component for continuous improvement of the accuracy of the automated results (Step 12) by leveraging the annotations offered by the course instructor, a domain expert, to learn how the rankings of similar QA answers in the future should be adjusted.

Prototype Test Results and Discussion

This section introduces our proof-of-concept prototype of the proposed system and the preliminary results. This prototype was developed at Athabasca University for research and test purposes.

The QA Prototype

We have developed a prototype user interface as described in Figure 2 for Athabasca University. Figure 3 shows the Moodle plugin as displayed to the student. The plugin reveals the number of questions asked by the student that are still awaiting answers, the number of responses with unread automated answers (Step 7), and the number of replies from the course instructor yet to be read (Step 10).
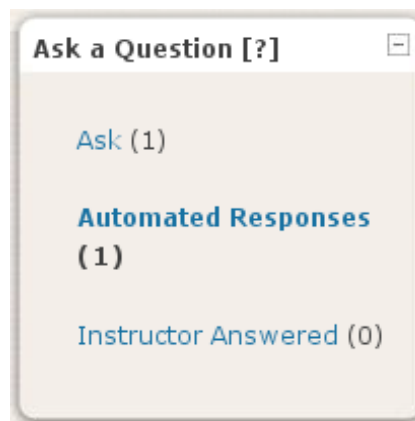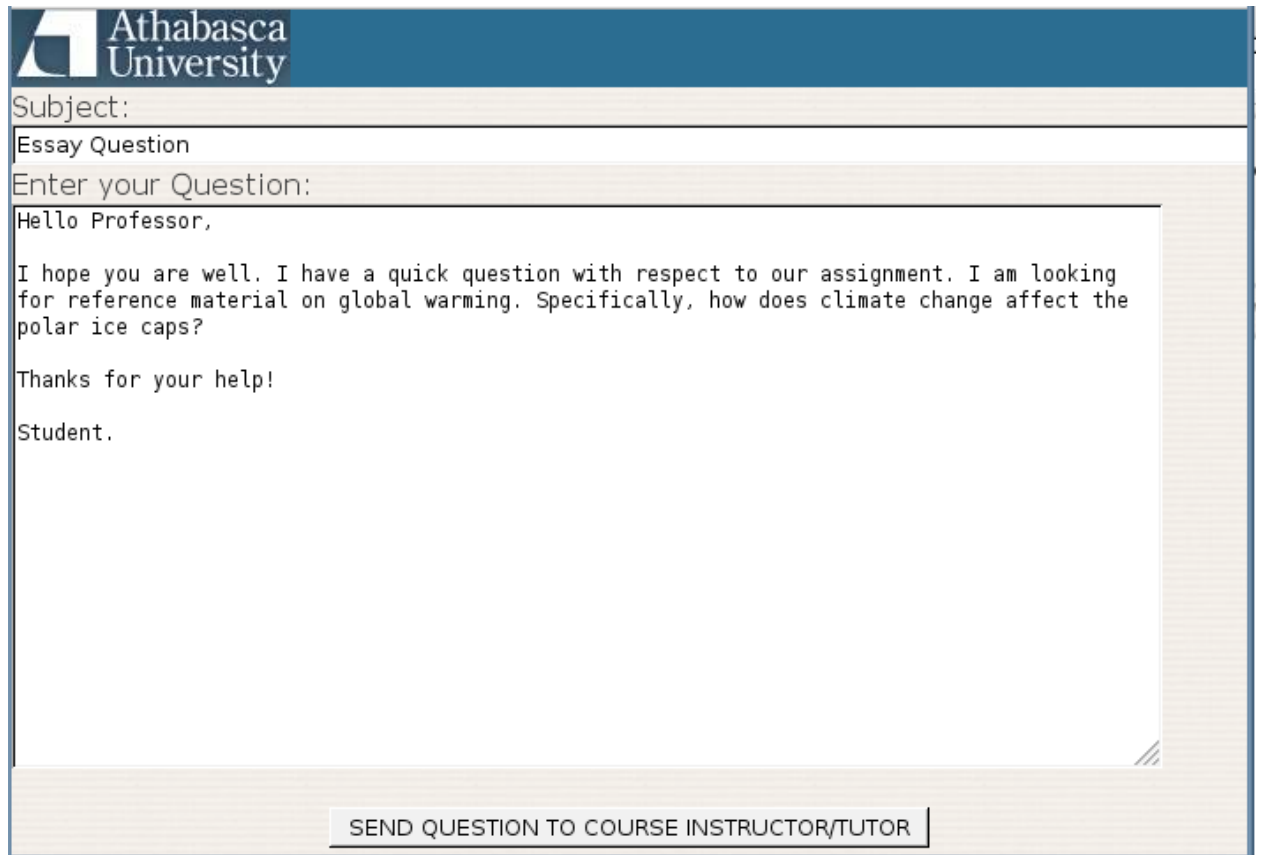

Figure 3. Moodle student plugin to QA system.

The student submits a question by clicking on "Ask" and entering a subject and message body to his/her question for the instructor. Figure 4 illustrates the question submission interface including a possible question asked by a student.
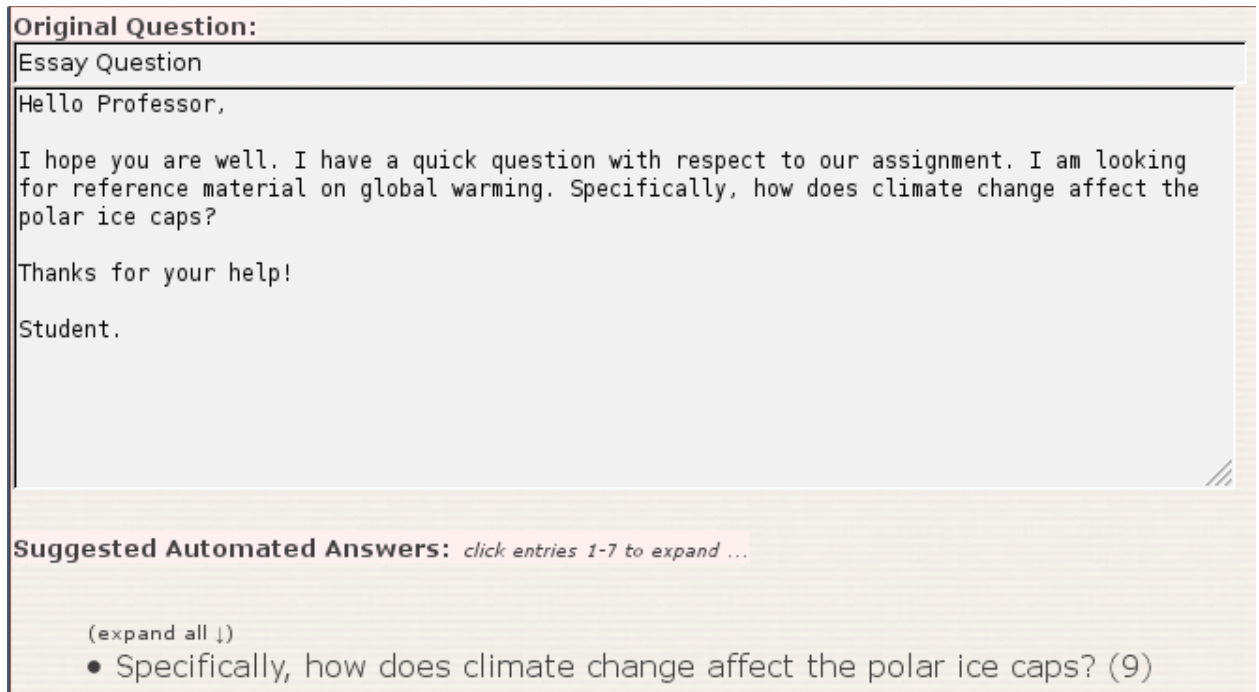
Figure 4. Moodle interface for submitting a question to the QA system.

The simple interface resembles the composition form of a typical email program. Note that these types of questions pose an extra challenge to the QA system for a number of reasons. First, the colloquial nature of the message (i.e., "Hello Professor") introduces informal language not often included in a query to a search engine. Second, multiple questions can be asked within a message that may be intermixed with nonquestions (i.e., "I hope you are well"). The system must identify the questions while ignoring the superfluous parts of the message. The task of precisely identifying questions is very much an open problem. Our implementation assumes that each sentence is a potential question. The message is decomposed into individual sentences using the sentence detector of OpenNLP; each sentence is then submitted for candidate answers. Once the automated suggested answers are formulated, the student is informed through the Moodle plugin (see Figure 3). The query portion of the message is identified and displayed to the student underneath his/her original message.

Figure 5. Automated response returned by the QA system.

Figure 5 shows the primary user interface for retrieving the suggested answers from the automated system and from the course instructor. The number in the bracket (9) after the question indicates the count of retrieved documents that the system believes is relevant to the student's query. Clicking on the question reveals the list, which includes a summary paragraph, a relevancy score, and a link to the source document. Figure 6 shows the suggested second answer and its score from the list of nine answers.

Figure 6. Suggested second answer from the list of nine.

A student can verify his/her question was fully considered by the QA system by clicking the "expand all" link which reveals the decomposition of the query, as shown in Figure 7.

Suggested Automated Answers: *click entries 1-7 to expand …*

- Essay Question (0)

- Hello Professor, I hope you are well. (0)

- I have a quick question with respect to our assignment. (0)

- I am looking for reference material on global warming. (0)

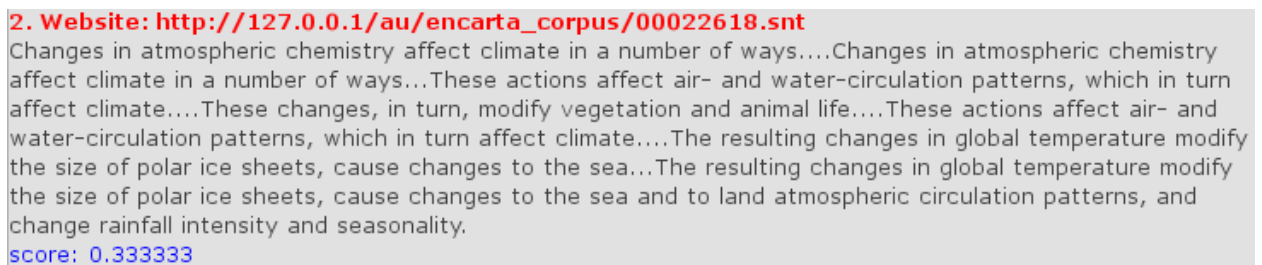- Specifically, how does climate change affect the polar ice caps? (9)

- Thanks for your help! (0)

- Student. (0)
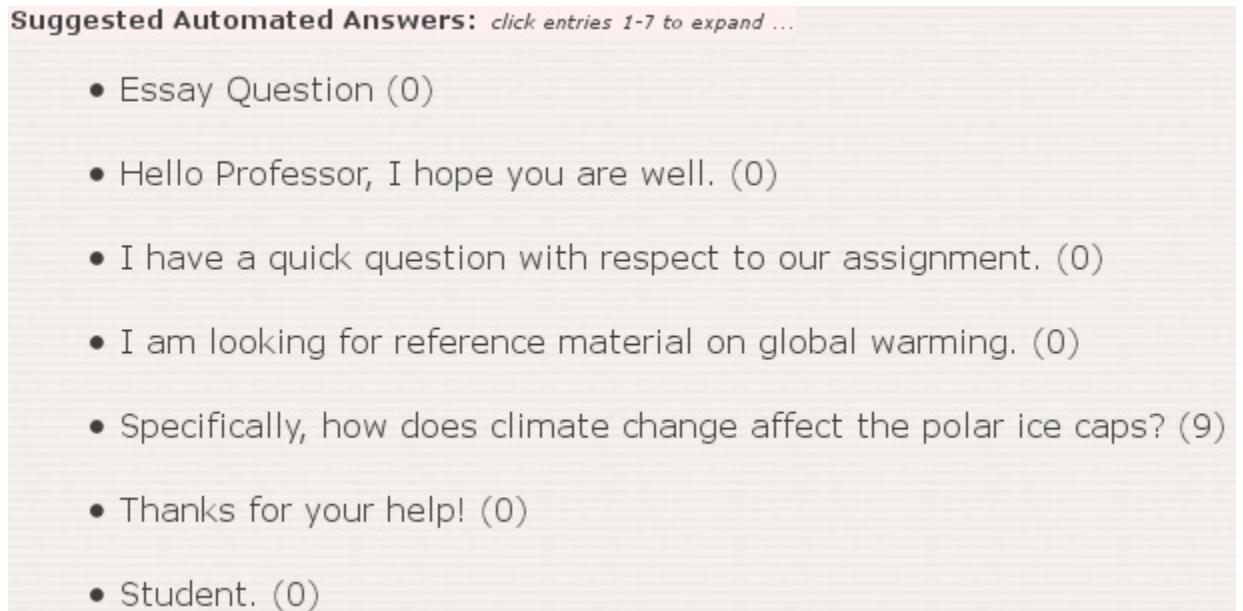
Figure 7. Submitted message split into sentences and answered by the QA system.

By default, sentences with zero answers are not displayed and this method is used as a simple filter to separate questions from anecdotal speech. In the future we plan to investigate state-of-the-art methods to differentiate questions from statements. In the interim, our simple no-zero answer filter policy has proven satisfactory. Also noteworthy is that the subject of the message is included as part of the query. This is intentional as in practice the subject line of a message often contains a synopsis of the message and usually excludes the colloquial extras that may cause confusion to the machine. Due to the concise nature of a subject line, we plan to incorporate the subject directly into each of the queries in an attempt to further improve the answers returned.

Ask a Question [?]

Student Questions (3)
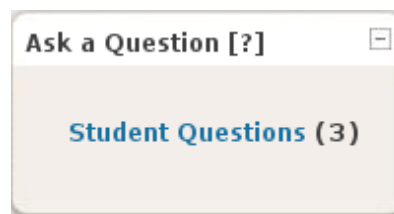
Figure 8. Moodle instructor plugin to the QA system.

Figure 8 shows the view of the Moodle QA from the perspective of the course instructor. A count of the number of unanswered questions pending is given in the brackets. The instructor can reply with his/her own message, annotate the automated answers, and/or post the reviewed solution directly to the course FAQ or forum.

QA Matching Theory and Methodology

With respect to the task of associating answers with questions there are typically two classes of queries: factoid based and information retrieval. Factoid queries give matter-of-fact type answers of a simple nature. Examples of such queries include "what are the first four

digits of PI?," "who was the second president of the United States?," and "what is the ratio of hydrogen to oxygen in water?". In contrast, information retrieval questions attempt to locate supporting documents for a specific concept, topic, or idea. Such examples are: "how does the warming of the oceans affect weather patterns?" and "what is the link between smoking and heart attack?" For educational purposes, our focus is on information retrieval type queries. Unfortunately, OpenEphyra out-of-the-box is geared toward factoid answers only. Our first task was to modify the pipeline to allow for the latter. Our FrameNet inspired solution replaces the OpenEphyra pipeline. This pipeline includes our own answer filtering technique that involves incremental clustering of the top answers based on a dynamically adjusted boundary. Figure 9 shows the new pipeline as an alternative to the OpenEphyra pipeline.

Figure 9. Alternative pipeline based on FrameNet.

The FrameNet lexical database consists of more than 1,000 semantic frames wherein each frame represents an event or concept. Associated with each frame are specific f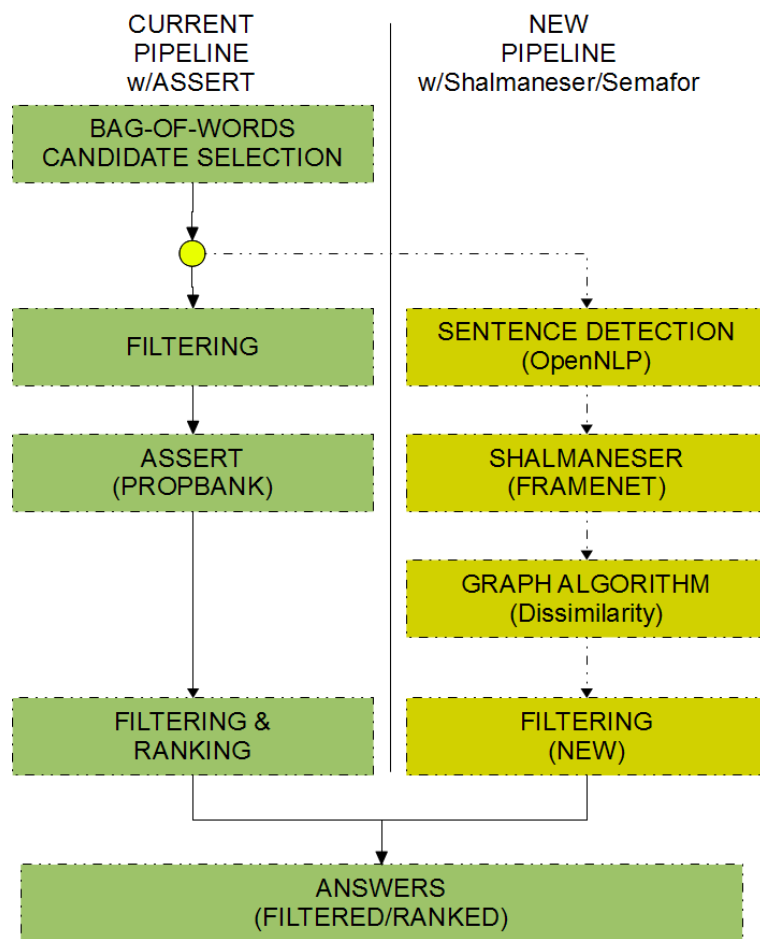rame elements, sometimes referred to as roles, which represent the entities or participants of the frame. Additionally, the FrameNet database is relational in which frames may inherit from other frames.

Consider the act of eating which consists of what is being eaten (ingestibles) and who is doing the eating (ingestor). This act may or may not involve a utensil (instrument). In FrameNet, this event is represented by the frame [Ingestion] and includes frame elements ingestor, ingestibles, instrument, and others.

example: $\underline{\text{John}}_{[\text{Ingestion:ingestor}]}$ tried to eat his $\underline{\text{soup}}_{[\text{Ingestion:ingestibles}]}$ with a $\underline{\text{fork}}_{[\text{Ingestion.instrument}]}$.

Our graph algorithm involves locating matching frames, either directly or through inheritance relationships, in order to translate this intersection into a heuristic function that returns a dissimilarity measure between the query and candidate answers. To illustrate, consider the query "are potential and kinetic energy related" and two possible candidate answers: "there is ample data relating energy drinks to headaches" and "energy related to position is called potential energy." Although both candidate answers contain the matching words *energy* and *relate*, the candidate answer "energy related to position is called potential energy" is clearly preferred. In order to resolve this ambiguity, our algorithm first performs a FrameNet semantic analysis of the query and candidate answers as shown in Figure 10. The scoring is obtained through a relative distance calculation between weighted paths from matching words of the query and candidate answer that share the closest intersecting frame. The further apart the intersection occurred, the less similar an answer is to the question.

These paths incorporate the inheritance model of FrameNet where each super-frame is a more generalized abstraction of a child frame concept. The edges are weighted with values $\alpha_i v^r$ where (v) is some arbitrarily chosen base raised to an incrementally larger exponent (*r*) at higher graph levels. Hence, weights at lower levels (more specific frames) are smaller than those at higher levels (generalized frames). A learning vector (α), derived through a supervised machine learning algorithm and a training set, provides weight adjustments for a better fitted model. Path values are computed through the product of the traversed edges and log normalized to account for large exponential values. Figure 11 illustrates such paths constructed from the semantic analysis of Figure 10.
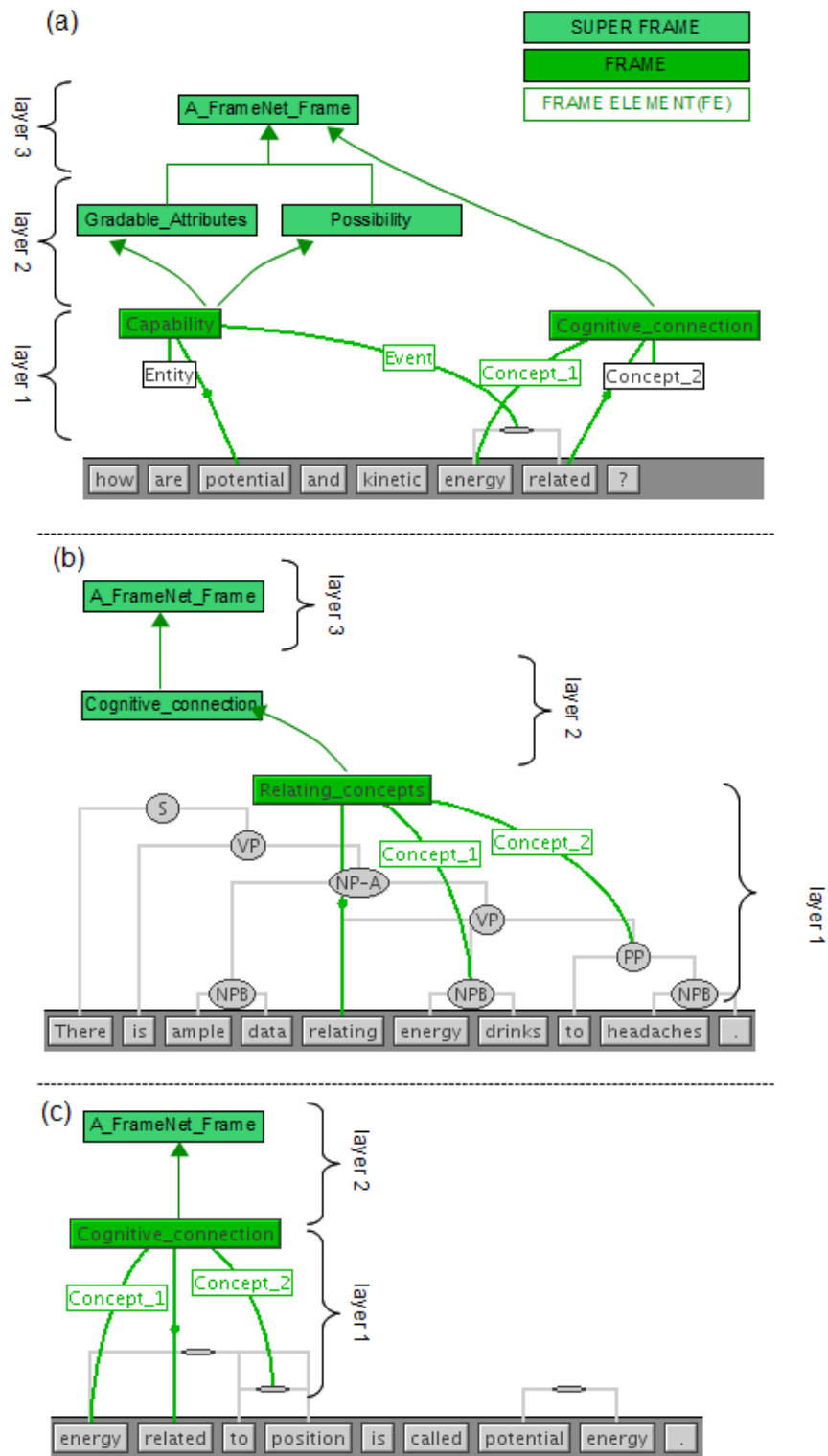
Figure 10. FrameNet semantic role labeling of query (a) and candidate answers (b), (c).
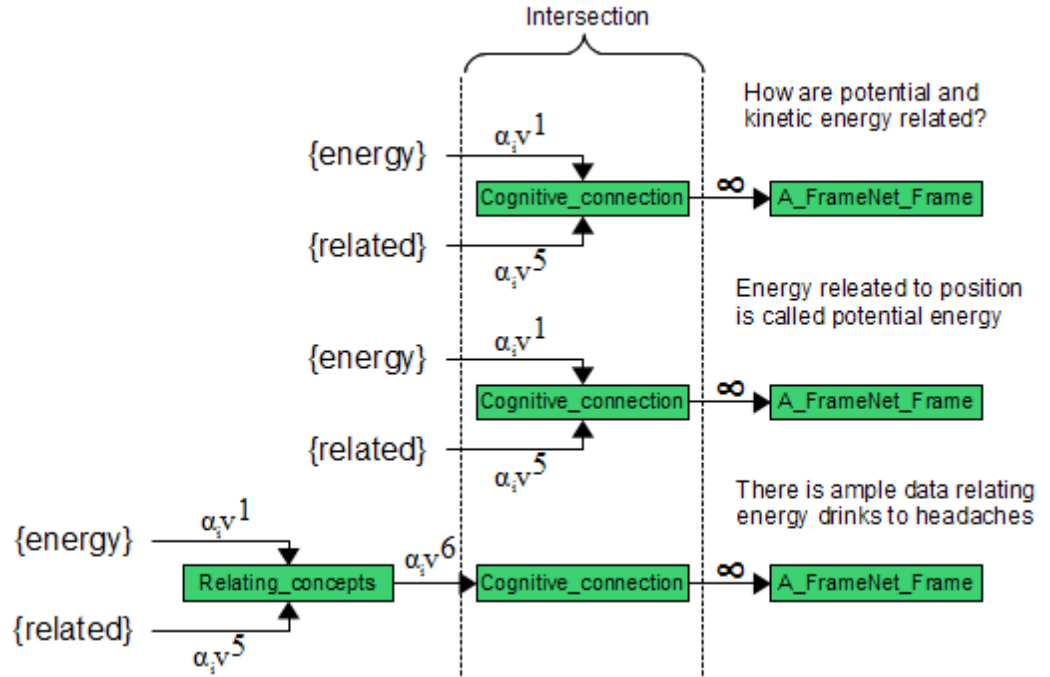
Figure 11. Paths with assigned edge weights from matching words to the topmost super-frame derived through Figure 10.

Notice that each edge weight along a path is exponentially larger than its predecessor with the exception of a special pseudo-frame called [A_FrameNet_Frame] which receives an edge weight of infinity. All frames are ultimately descendants of [A_FrameNet_Frame]. The infinity edge weight is assigned to discourage matching on this universal frame. The query and two candidate answers all share a common frame of [Cognitive_connection]. The heuristic value is computed from the product of the query's edges through the intersecting frame to the candidate's edges toward the matching words (energy/related). For example, the word energy in the preferred candidate answer has path value of $\alpha_i v^1 \, \alpha_i v^1$ compared to $\alpha_i v^1 \, \alpha_i v^6 \, \alpha_i v^1$ of the less desirable answer due to the extra frame of [Relating_concepts]. Since $\alpha_i v^1 \, \alpha_i v^1 < \alpha_i v^1 \, \alpha_i v^6 \, \alpha_i v^1$, the word energy in the answer "energy related to position is called potential energy" is favored over its use in "there is ample data relating energy drinks to headaches." A similar calculation can be done for the matching word *related*. The goal is to find a minimal cost spanning tree between all constructed paths. In the next section of this paper, we test this algorithm on *Microsoft Question-Answering Research Corpus* (Microsoft Research, 2008) and test the PropBank solution with the *Introduction to Programming Using Java* online text (Eck, 2011). A more detailed technical introduction and experimental results of the algorithm will be reported in a separate paper.

QA Matching Results

We began preliminary testing on the effectiveness of OpenEphyra with PropBank style semantic filtering. We utilized the Microsoft Research QACorpus which was the basis of *Microsoft Encarta 98* digital encyclopedia published from 1993 to 2009. This corpus comprised 1,400 individual questions with each question containing multiple correct answers

for a total of 10,000+ QA pairs. The corpus is further divided into 32,715 individual files with each file containing a number of paragraphs. Overall, the corpus provided 163 megabytes of textual data for experimentation.

We evaluated our prototype by asking 33 information retrieval type questions and recording the hit rate (correct answers), miss rate (incorrect answers), and the positional ranking of the best answer. Table 2 gives the top and bottom 12 queries based on the number of relevant answers returned.

Table 2

Top/Bottom Twelve Questions

QUERY	HIT	MISS	BEST RANK
How is calculus used?	8	2	2
How do insects transmit disease?	8	2	1
How is pollution contributing to global warming?	7	3	1
How does diet improve your health?	7	3	8
what dogs originated in germany?	6	4	1
How is magnetism related to electricity?	4	3	3
↑ TOP 6 ↑			
↓ BOTTOM 6 ↓			
How is religion affected peoples attitudes toward birth control?	1	9	9
How has the space program contributed to everyday life?	0	9	N/A
Why do onions make your eyes water?	0	3	N/A
How is population explosion affecting the earth's resources?	0	10	N/A
How does the warming of the oceans affect weather?	0	4	N/A
How are potential and kinetic energy related?	0	0	No answers

Although the preliminary answers given by OpenEphyra with ASSERT showed promise, the overall precision needs improvement before deployment in an educational setting. Hence, our research is now focused on increasing the hit rate, lowering the miss rate, and improving the positional rankings of the best answers. These observations were the catalyst for our own filtering and scoring algorithms based on FrameNet semantic role labeling and graph theory matching.

To demonstrate we took the query "how are potential and kinetic energy related?" that produced no answers in the existing PropBank pipeline (see Table 2) and subjected it to our proposed FrameNet alternative. In the first step of the pipeline, 1,368 potential candidate answers to the query were selected through a simple bag-of-words approach. The candidates were syntactically parsed with frame elements assigned and inheritance chain calculated as described in the QA Matching Theory and Methodology section of this paper.

After completion of the similarity scoring, 29 answers survived our filtering stage of the pipeline. The top five answers and their dissimilarity scores are given in Table 3. For comparison, the ranking of the bottom five answers are also given; note that lower scores are better (less dissimilar) to the question asked.

Table 3

Top/Bottom Ten Answers to the Question "How are Potential and Kinetic Energy Related?"

RANK	ANSWER	SCORE
1	The law, which states that the sum of kinetic energy, potential energy, and thermal energy in a closed system remains constant, is now generally known as the first law of thermodynamics.	15.95594
2	Not until the action of molecules in a system was better understood by the development of the kinetic theory could this internal energy be related to the sum of the kinetic energies of all the molecules making up the system.	15.95594
3	Electric potential is related to electrical potential energy.	15.95594
4	Boltzmann Constant in physics, fundamental constant, designated k, that relates the average kinetic energy of particles in a gas to the temperature of the gas.	15.95594
5	Energy associated with motion is known as kinetic energy, and energy related to position is called potential energy.	29.77145
	↑ TOP 5 ↑	
	↓ BOTTOM 5 ↓	
1364	production, the excess energy is imparted to the electron pair as kinetic energy.	175.7519
1365	Endothermic reactions are always associated with the spreading, or the dissociation, of molecules.	175.7519
1366	If a gun is fired, the potential energy of the gunpowder is	186.738
1367	In the eye itself, winds are usually light and skies are partly cloudy.	186.738
1368	Such a body has an excess of electrons.	186.738

Our next phase of testing involved establishing how such a system, incorporated into the LMS of an educational organization, could improve learning particularly in the online/distance situations where face-to-face instructor interaction is infrequent or unavailable. In this test, the corpus consisted of the online introductory Java textbook comprised of 326 web pages freely available through a Creative Commons License. Table 4 gives an example of the types of questions a student may ask while taking this course. Such expected questions, given the nature of the course, may be "how do I compile my program?" or "how do I use annotations?".

Similar to Table 2, the results show how many of the suggested answers are relevant to the question and the positional ranking of the best answer amongst the returned candidates. The queries were executed under the modified PropBank pipeline with mixed results with respect to hit, miss, and best rank. For example, when asked "how do I compile my program?," 7 out of the 10 returned answers were considered satisfactory for the question (70% precision) but the best answer of the set was found (ranked) in the tenth position.

Nonetheless, a satisfactory (correct) answer was given somewhere in the suggested top-10. With the ability of the course instructor to annotate and/or modify the list, the student's question should be adequately answered. We believe, as in our previous tests with the Microsoft Encarta corpus, that the alternate FrameNet pipeline will yield improvements. Further testing is ongoing.

Table 4

Commonly Asked Questions for an Introductory Course on Java Programming

Relevancy	Question -- [Rank] Best Answer
7 / 10	*Q1: How do I compile my program?* [10] That section contains information about how to compile and run Java programs, and you might want to take a look at it before trying to write and use your own programs.
8 / 9	*Q2: How do I use recursion?* [8] Recursive algorithms work by reducing a complex problem into smaller problems which can solved either directly or by applying the same algorithm "recursively." RGB
6 / 10	*Q3: How can I use graphics?* [2] If OSC is of type BufferedImage, then the method OSC.getGraphics() returns an object of type Graphics that can be used for drawing on the image....An off-screen canvas is a region of the computer's memory that can be used as a drawing surface...
1 / 5	*Q4: How do I use annotations?* [1] Some annotations are used to make the programmer's intent more explicit....For example, @Override is a standard annotation that can be used to annotate method definitions....Instead of writing that code by hand, a programmer could use an annotation to mark the variables that are part of the state that is to be saved....An annotation used in this way is an aid to writing correct programs, since the programmer can be warned about a potential error in advance, instead of having to hunt it down later as a bug

Research Motivation

The motivation for this research is inspired by offline communication and forum posts that instructors frequently encounter with their students.

Hello Professor:

I need help with an assignment question:

"Analyze the UML model as described in Figure 4 of the textbook and give a solution."

I could not find this figure in the textbook, nor the assigned reading. Where is it?

Figure 12. An example of a student's forum post.

Figure 12 contains some key observations. First, the student's query is very specific. Although any popular search engine will provide links to resources on the topic of UML, the student's question is considerably more focused in asking for a seemingly missing figure related to a textbook reading assigned to this course. In order to accommodate this degree of specificity, deeper natural processing techniques such as semantic role labeling are necessary. Computational complexity, as shown in Table 1, would suggest this technology would normally be prohibitive. However, our proposed offline asynchronous implementation is tractable and consequently can provide an educational benefit within this context. A second observation is although the question is directed to the course instructor ("Hello Professor"), it has nonetheless been posted to the course forum. The reason is two-fold: to ask his peers in the event someone may know the answer or for the benefit of his peers in the circumstance that others may have the same question. Our prototype system specifically addresses this by giving the instructor the option to submit his/her comments along with the annotated auto-responses directly to the forum.

Our objective is to produce a QA system of sufficient accuracy of benefit to assist students and teachers. Our preliminary tests results have shown that the existing pipeline of the OpenEphyra framework is insufficient for this task. Consequently, our theoretical work focuses on a modified pipeline (Figure 9) that utilizes FrameNet SRL and weighted spanning trees (Figure 10/11). Systematically, we have introduced features in our implementation to further aid in the prototype's accuracy. The instructor has the option to not allow the system to respond to the student with its answers until the instructor has had an opportunity to review the system's suggestions and provide his/her corrections (see Figure 2 sequence no. 7 disabled). This allows the proposed system to operate either in unsupervised, semi-supervised, or fully supervised mode at the discretion of the instructor.

We have also incorporated a re-enforcement learning step (Figure 2 sequence no. 12) for perpetual accuracy improvements facilitated through the adjusted edge weight vector (α) of the model (Figure 11). Furthermore, the constrained domain of the course curriculum further aids in accuracy by limiting the search to only content available within the online offering.

Finally, it is worth noting that students are seeking assistance from search engines frequently, arguably more so in an online distance education situation than in the traditional face-to-face paradigm. Consequently, a QA system that is focused on the course material and moderated by the instructor offers advantages over the less restrictive, and sometimes inaccurate, sources of online information such as search engines and Wikipedia.

Related Works

A recent survey by Shuib et al. (2010) on 129 postgraduate computer science students found that students were having considerable difficulty in finding information appropriate to their learning style using the search tools available. It is also clear that not all available searchable content is created equal when measured against its educational value. In a case

study involving the use of a digital library in a middle school, Abbas et al. (2002) observed the usefulness of search systems varied based on the type of classes and differing student achievement levels. They conclude that an educationally useful search engine is more than seeking on-topic documents of interest but also is an organizational and collaboration tool that teaches iteration and refinement processes often leading to more than a single 'correct' answer.

Marshall et al. (2006) examined information retrieval in education using a digital library environment known as GetSmart. This system successfully augmented traditional search with concept mapping. Of 60 university students surveyed taking an undergraduate computing course, 86% reported that the marriage of concept mapping with search was "very valuable" or "somewhat helpful" in their queries.

Curlango-Rosas et al. (2011) developed an intelligent agent specifically for the retrieval of learning objects. A learning object (LO) was defined as "any entity, digital or non-digital, that may be used for learning, education or training [IEEE 2002]." Obviously, an LO is of special interest to educators over other forms of web content. Their "Learning Object Search Tool Enhancer" (LOBSTER) demonstrated that 96% of piloting teachers found suitable quality LOs compared to only an 80% success rate when using Google.

Martin and Leon (2012) proposed a digital library search for teachers that leveraged semantic and natural language processing. Their system made extensive use of case-based reasoning technology to construct a searchable ontology they named OntoFAMA. In the survey 50 engineering students were asked to rank the relevancy of suggested LOs by both OntoFAMA and Google; 85.4% of retrieved LOs from OntoFAMA were considered of acceptable or better quality compared to only 78.5% when using Google (a measure of precision). Only 14.4% of OntoFAMA's suggestions were considered poor quality compared to Google's 21.3% (a measure of recall).

A supervised learning approach to searching was investigated by Prates and Siqueira (2011). They used information extraction to create a training set that forms a baseline to the appropriateness of a retrieved document in a specific educational context. A teacher selects segments from available sources deemed as representative of the content he/she finds suitable. A student's query is expanded by using additional relevant terms as learned through the baseline before submission to a web search engine. Empirical tests showed that queries expanded in this manner gave better precision than their original nonexpanded counterparts.

In education a large body of information remains underutilized due to lack of an effective information retrieval system. Mittal, Gupta, Kumar, and Kashyap (2005) recognized this and they devised a QA system to search for information stored in PowerPoint slides, FAQs, and e-books. Feng et al. (2006) investigated a QA like discussion-bot, using a remote agent to provide answers to students' discussion board questions. Their results highlighted the importance of QA in online and distance education.

Feng, Shaw, Kim, and Hovy (2006) examined an intelligent robot that intercepted posts

to an educational forum and volunteered its own answers. This was similar to our own QA system, but lacked deeper semantic processing techniques. This gave quick answers to questions but the quality of the responses suffered. Cong, Wang, Lin, Song, and Sun (2008) used QA in their research to retrieve answers from online forums. Their technique took into consideration whether the answer was by a known domain expert and weighed his/her response accordingly.

Marom and Zukerman (2009) experimented on corpus-based techniques for the automation of help-desk responses, using request-response e-mail dialogues between customers and employees of a large corporation. Help-desk e-mail correspondence contains a high level or repetition and redundancy, and responses to customers contain varying degrees of overlap. This repetition and overlap is also commonplace in an educational environment where multiple students may have the same question. Hence, our prototype allows the instructor to post the annotated answers to the course forum.

Surdeanu, Ciaramita, and Zaragoza (2011) created a re-ranking model for non-factoid answer ranking using question-answer pairs from Yahoo! The results of the experiment show that semantic roles can improve ad hoc retrieval on a large set of non-factoid open-domain questions. Their paper provided compelling evidence that complex linguistic features such as semantic roles have a significant impact on large-scale information retrieval tasks and consequently can be a major benefit in online education.

Shen and Lapta (2007) introduced a general framework for answer extraction which exploited the semantic role annotations in FrameNet. The results were promising and show that a graph-based answer extraction model can effectively incorporate FrameNet style role semantic information. To the best of our knowledge, our proposed asynchronous QA system has not been implemented.

Conclusion

The Internet has provided educational opportunities in ways that were unavailable just 10 years ago. Online and distance learning has extended these opportunities past the traditional campus-based classes into students' homes, workplaces, and smart phones. Learning is now self-paced with the instructor acting as a facilitator and mentor. However, this anytime/anywhere access has presented a challenge in answering students' questions particularly in a medium where time zones are irrelevant. Our research contribution aims to develop an asynchronous QA system to fit student needs and support student and teacher participation in the learning process. Instructor feedback can validate the student's choice or provide instruction in choosing a suitable answer. Our system can assist with this validation. We also hope this research leads to improved communications between student and teacher and to lessen the frustrations a student may encounter in a distance learning environment where real-time access to the instructor may be absent. Our future work involves continued improvement to the question answering accuracy using semantic role labeling as well as enhancements to the Moodle user-interface to the extent that our working prototype

can be evaluated in an actual online course. Future papers will report on the progress of the QA accuracy and include evaluations from teachers and students who participated in the Moodle pilot.

Acknowledgement

References

Abbas, J., Norris, C., & Soloway, E. (2002). Middle school children's use of the ARTEMIS digital library. In *Proceedings of the 2nd ACM/IEEE-CS Joint Conference on Digital libraries* (JCDL '02) (pp. 98-105). New York, USA: ACM.

Baker, C. F., Fillmore, C. J., & Lowe, J. B. (1998). The Berkeley FrameNet Project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1* (pp. 86-90). Association for Computational Linguistics.

Beldarrain, Y. (2006). Distance education trends: Integrating new technologies to foster student interaction and collaboration. *Distance Education, 27*(2), 139-153.

Cong, G., Wang, L., Lin, C.-Y., Song, Y.-I., & Sun, Y. (2008). Finding question-answer pairs from online forums. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 467-474). ACM.

Curlango-Rosas C., Ponce, G., & Lopez-Morteo, G. (2011). A specialized search assistant for learning objects. *ACM Trans. Web, 5*, 4, Article 21.

Das, D., Schneider, N., Chen, D., & Smith, N. (2010). *SEMAFOR 1.0: A probabilistic frame-semantic parser*. Language Technologies Institute, School of Computer Science, Carnegie Mellon University.

Denvir, C., Balmer, N. J., & Pleasence, P. (2011). Surfing the web - recreation or resource? Exploring how young people in the UK use the Internet as an advice portal for problems with a legal dimension. *Interacting with Computers, 23*(1), 96-104.

Eck, D. J. (2011). *Introduction to programming using Java* (6th ed.) Retrieved from http://math.hws.edu/javanotes/

Erk, K., & Pado, S. (2006). Shalmaneser - a flexible toolbox for semantic role assignment. In *Proceedings of LREC 2006* (pp. 527-532).

Feng, D., Shaw, E., Kim, J., & Hovy, E. (2006). An intelligent discussion-bot for answering student queries in threaded discussions. In *Proceedings of the 11th International Conference on Intelligent User Interfaces* (pp. 171-177). ACM.

Marom, Y., & Zukerman, I. (2009). An empirical study of corpus-based response automation methods for an e-mail-based help-desk domain. *Computational Linguistics, 35*(4), 597-635.

Marshall, B., Chen, H., Shen R., & Fox, E. (2006). Moving digital libraries into the student learning space: The GetSmart experience. *J. Educ. Resour. Comput. 6,* 1, Article 2.

Martin, A., & Leon, C. (2012, April). An intelligent e-learning scenario for knowledge retrieval. *Global Engineering Education Conference (EDUCON), 2012 IEEE* (pp.1-6).

Microsoft Research (2008). *Microsoft research - question-answering corpus*. Retrieved April 24, 2012 from http://research.microsoft.com/en-us/downloads/88c0021c-328a-4148-a158-a42d7331c6cf

Miller, G. A. (1995). WordNet: A lexical database for English. *Communications of the ACM, 38*(11), 39-41.

Mittal, A., Gupta, S., Kumar, P., & Kashyap, S. (2005). A fully automatic question-answering system for intelligent search in e-learning documents. *International Journal on E-Learning, 4*(1), 149-166.

Palmer, M., Gildea, D., & Kingsbury, P. (2005). The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics,* 31(1), 71-106.

Pradhan, S., Ward, W., Hacioglu, K., & Martin, J. H. (2004). Shallow semantic parsing using support vector machines. In *Human Language Technology Conference/ North American Chapter of the Association for Computational Linguistics (HLT/ NAACL).*

Prates, J., & Siqueira, S. (2011). Using educational resources to improve the efficiency of web searches for additional learning material. In *Proceedings of the 2011 IEEE 11th International Conference on Advanced Learning Technologies (ICALT '11)* (pp. 563-567). Washington, DC, USA: IEEE Computer Society.

Punyakanok, V., Roth, D., & Yih, W. (2008). The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics, 34*(2), 257-287.

Pynoo, B., Tondeur, J., van Braak, J., Duyck, W., Sijnave, B., & Duyck, P. (2012). Teachers' acceptance and use of an educational portal. *Computers & Education, 58*(4), 1308-1317.

Reigeluth, C. M. (1999). What is instructional design theory? In C. M. Reigeluth (Ed.), *Instructional design theories and models: A new paradigm of instructional theory* (Vol. 2, pp. 5–29). Mahwah, NJ: Lawrence Erlbaum Associates.

Schlaefer, N., Gieselmann, P., & Sautter, G. (2006). The Ephyra QA system at TREC 2006. *Proceedings of the 15th Text Retrieval Conference.*

Shen, D., & Lapata, M. (2007). Using semantic roles to improve question answering. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)* (pp. 12-21). Association for Computational Linguistics.

Shuib, M., Liyana, N., Noorhidawati, A., Ismail, B., & Hafiz, M. (2010, Dec.). The use of information retrieval tools: A study of computer science postgraduate students. *Science and Social Research (CSSR), 2010 International Conference* (pp.379-384).

Surdeanu, M., Ciaramita, M. & Zaragoza, H. (2011). Learning to rank answers to non-factoid questions from web collections. *Computational Linguistics, 37*, 351-383.

Wen, D., Cuzzola, J., Brown, L., & Kinshuk (2012). Exploiting semantic roles for asynchronous question answering in an educational setting. In L. Kosseim & D. Inkpen (Eds.), *Advances in artificial intelligence - Canadian AI 2012, LNAI 7310* (pp. 374-379). Germany: Springer-Verlag.

Athabasca University 🔺