Personalized Question Answering: A Use Case for Business Analysis

VinhTuan Thai[1], Sean O'Riain[2], Brian Davis[1], and David O'Sullivan[1]

[1] Digital Enterprise Research Institute,
National University of Ireland, Galway, Ireland
{VinhTuan.Thai,Brian.Davis,David.OSullivan}@deri.org
[2] Semantic Infrastructure Research Group,
Hewlett-Packard, Galway, Ireland
sean.oriain@hp.com

Abstract. In this paper, we introduce the Personalized Question Answering framework, which aims at addressing certain limitations of existing domain specific Question Answering systems. Current development efforts are ongoing to apply this framework to a use case within the domain of Business Analysis, highlighting the important role of domain specific semantics. Current research indicates that the inclusion of domain semantics helps to resolve the ambiguity problem and furthermore improves recall for retrieving relevant passages.

Key words: Question Answering, Business Analysis, Semantic Technology, Human Language Technologies, Information Retrieval, Information Extraction

1 Question Answering Overview

Question answering (QA) research originated in the 1960s with the appearance of domain specific QA systems, such as BASEBALL which targeted the American baseball games domain and LUNAR which in turn focused on the lunar rock domain [1]. These early systems were concerned with answering questions posed in natural language, against a structured knowledge base of a specific domain [1]. They are commonly known as natural language front ends to databases. Research within this field remains active with the introduction of new approaches and techniques to enhance QA performance in more real-life, complex settings (e.g. [2–4]).

With the advent of Semantic Web technologies, domain specific knowledge can also be encoded within formal domain ontologies. This in turn has motivated the growth of another branch of QA research; focusing on answering natural language questions against formal domain ontologies. Attempto Controlled English (ACE) [5] and AquaLog [6] are recent research additions to this area. Rather than translating natural language questions into SQL statements, these systems translate them into variants of first-order predicate logical such as Discourse Representation Structure (DRS) in the context of ACE and Query-Triples for

AquaLog respectively. Consequently this permits answer derivation as an outcome of the unification process of both question and knowledge-based logical statements.

The application of domain specific QA extends further than systems that have their knowledge sources completely encoded in relational database or formal ontologies. Many research initiatives have investigated the use of domain ontologies or thesaurus in assisting finding answers for questions against a small volume of collections of unstructured texts contained within terminology-rich documents. A variety of approaches have been proposed, ranging from a pure Vector Space Model based on traditional Information Retrieval research, extending this approach a with domain specific thesaurus in [7], to a template-based approach for medical domain systems [8, 9], or a computational intensive approach in [10], the goal being to convert both the knowledge source and the question into Minimal Logical Form.

Apart from domain specific QA research, the introduction of the QA track at Text Retrieval Conference TREC-8 in 1999 involved researchers focusing on combining tools and techniques from research fields such as Natural Language Processing, Information Retrieval, and Information Extraction in an attempt to solve the QA problem in open-domain setting: the main knowledge source being a predefined large newswire text corpus, with the World Wide Web acting as an auxiliary source of information. The questions being asked consist mainly of: factoid questions, list questions, definition questions and most recently, the relationship task type question [11]. A review of participating systems in TREC QA track is beyond the scope of this paper. Interested readers are referred to [12] for further details. Of crucial importance however is that the existing QA track does not target domain specific knowledge.

QA, in itself, remains an open problem. The research overview has highlighted the diversity of QA systems under development. Each system is designed to address the problem in a particular usage scenario, which imposes certain constraints on available resources and feasible techniques. Nevertheless, there remain usage scenarios for QA systems that require addressing, one of which is Personalized Question Answering. We discuss our motivation for using Personalized Question Answering in Section 2.

The remainder of this paper is structured as follows: Section 3 describes our use case of Personalized QA within the Business Analysis domain. Section 4 presents a proposed framework for Personalized QA; Section 5 concludes this paper, reaffirms our goals and identifies future work.

2 Personalized Question Answering

Our motivation towards Personalized Question Answering stems from existing shortcomings within current QA systems designed for extracting/retrieving information from unstructured texts. The shortcomings are categorized below:

Authoritative source of information: In an open-domain QA setting, end-users have little control over the source of information from which answers are sought. The reliability of answers is based mostly on the redundancy of data present on the WWW [13]. Similarly, existing domain specific QA systems also limit the source of information to a designated collection of documents. To our knowledge, no QA system is designed in such a way that allows end-users to flexibly specify the source of information from which the answers are to be found. This is of importance with respect to the design of a QA solution. For the majority of existing work, the collection of documents must initially undergo pre-processing. This pre-processing is performed only once, the results being stored later for retrieval. This offline processing strategy makes a computational intensive approach (such as in [14, 10]), feasible because all the necessary processing is already performed offline before any questions can be asked, and therefore reduces significantly the time required to find answers at run time. A QA system that has a dynamic knowledge source will therefore need to take this amount of necessary processing into consideration.

Contextual information: The use of contextual information in QA has not received adequate attention yet. Only the work of Chung et al. [3] highlights the fact that while forming the question, users may omit important facts that are necessary to find the correct answer. User profile information is used in their work to augment the original question with relevant information. The design of QA systems therefore needs to take into account how and to what degree a given question can be expanded to adequately reflect the context in which it is asked.

Writing style of documents: Current domain specific QA systems are usually targeted to scientific domains, with the knowledge source, such as technical, medical and scientific texts [8, 14, 7], written in a straight-forward, declarative manner. This characteristic reduces the ambiguity in these texts. However, this is not always the case with other types of documents, for example business reports. Therefore, QA system should be able to utilize the domain and/or personal knowledge to resolve ambiguity in texts that are written in a rhetorical way.

Targeting to address the above limitations, we propose Personalized Question Answering, which:

- is domain specific, therefore avails of a formal domain ontology
- can cope with dynamic collection of unstructured texts written in rhetorical style
- can handle various question types
- resolves implicit context within questions
- provides an answer-containing chunk of texts rather than the precise answer

Before discussing details of the proposed framework, we first outline in Section 3, a use case for Personalized QA within the domain of Business Analysis.

3 Business Analysis use case

Business Analysis is largely performed as a Business Intelligence[3] (BI) activity with data mining and warehousing providing the information source for monitoring, identification and gathering of information. On-line analytical processing then allows differing data views and report generation possible from which further BI analysis may then be performed. Data excluded from the extract, transform and load phase passes through the process unaltered as unstructured information. Subsequent mining efforts on this information compound the problem by their reliance upon problematic document level technologies such as string-based searching resulting in information being missed.

It is this type of mining activity that enterprises performing customer analysis as a means to identify new business opportunities currently rely upon. The problem becomes more complex when it is considered that business analysts in performing company health checks depend largely upon the consolidated financial information and management statements found in the free text areas of the Form 10-Q. Management statements are those from the companies' CEO and are concerned with the companies' performance. They are viewed as a promotional medium for presentation of corporate image and are important in building credibility and investor confidence. Despite analysts having a clear understanding of the information content that the statements may contain, the searching, identification and extraction of relevant information remains a resource intensive activity.

Current BI technologies remain limited in this type of identification and extraction activities when processing information contained in unstructured texts written in a rhetorical manner. For example, part of performing a company health check involves building an understanding of that company's sales situation mentioned in Form 10-Q[4]. Sales[5] performance is in turn partially dependent upon product and services revenue. An intuitive and time-saving way to gain understanding on these areas is to pose in natural language non-trivial questions such as "*What is the strategy to increase revenues?*", "*Are there are plans to reduce the cost of sales versus revenue?*" and retrieve chunks of text that contain answers to these questions.

Our Personalized QA framework is based upon semantic technology and when applied to the Business Analysis domain, will offer business analysts the ability to expedite the customer analysis process by having potentially relevant information presented in a timely manner. This can be achieved by having the business analysts associate their knowledge in the form of a formal ontology and a set of

[3] Term introduced by Gartner in the 1980s that refers to the user-centered process of data gathering and analysis for the purpose of developing insights and understanding leading to improved and informed decision making.

[4] Quarterly report filed to the Securities and Exchange Commission (SEC) in the US. It includes un-audited financial statements and provides a view of the company's financial position. Relied upon by Investors and financial professionals when evaluating investment opportunities

[5] Discussion context is the software services domain

domain specific synonyms to the QA system, specify the source document (Form 10-Q), pose their questions, and retrieve chunks of text that contain answers to conduct further analysis. The framework is described in the next section.

4 Personalized Question Answering Framework

The proposed framework for Personalized QA, as shown in Fig. 1, consists of two main modules: Passage Retrieval and Answer Extraction. The Passage Retrieval module performs text processing of documents and analysis of questions on-the-fly to identify passages that are relevant to the input question. This coarse-grained processing reduces the search space for the answer significantly as only relevant passages are fed to Answer Extraction (which is a more computationally intensive module), to perform further fine-grained processing to identify chunks of texts containing the correct answer. The details of these modules are discussed below.

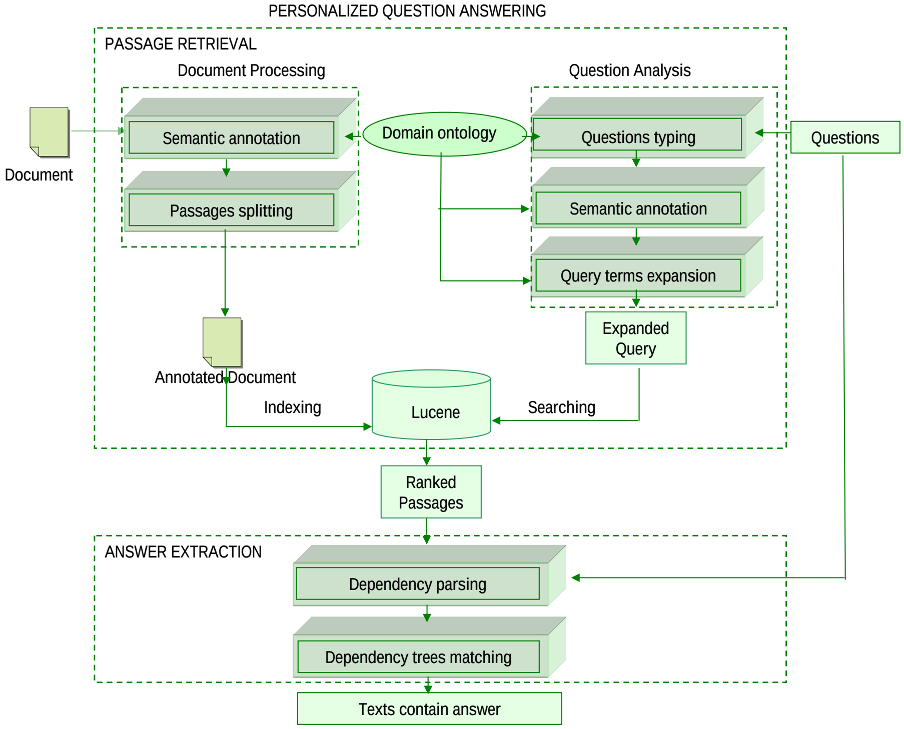


Fig. 1. Personalized Question Answering Framework

4.1 Passage Retrieval

Passage Retrieval serves as an important module in the whole QA process. If this module cannot locate any passage that possibly contains the answer when one actually exists, an answer cannot be found. On the other hand, as noted by Cui et al. [15], too many irrelevant passages returned by this module could hinder the Answer Extraction module in locating the correct answer. It is worth noting that many research works in open-domain QA have studied the Passage Retrieval problem and proposed different density-based algorithms, which are quantitatively evaluated by Tellex et al. [16]. The lack of domain specific knowledge makes these works different from ours significantly because no semantics is taken into consideration; the similarity between the question and the documents is statistically measured based only on the original words. However, this is rather an advantage of domain specific QA systems in terms of the resources available to them, than a limitation of current approaches being used for open-domain QA systems. The work of Zhang et al. [7] takes semantics into consideration while weighting similarity between the question and passages. This is similar to our Passage Retrieval approach described below; however, this work lacks the fine-grained processing performed by our Answer Extraction module to filter out passages that contain the query terms but not in the right syntactic structure to answer the question. The following paragraphs below describe each component of the Passage Retrieval module.

Document Processing: Document Processing involves two text processing tasks: Semantic annotation and Passage splitting.

Although to date there is no formal definition of "Semantic Annotation", this concept is generally referred to as "a specific metadata generation and usage schema, aiming to enable new information access methods and to extend the existing ones" [17]. In other words, the semantic annotation task is performed based on the domain ontology, in order to associate the appearances of domain specific terms or named entities with the respective ontological concepts, therefore anchoring those terms or named entities within contents to their corresponding semantic information. There have been a lot of research efforts within the field of semantic annotation with respect to discerning what to annotate, what additional information users expect to have, whether to embed annotation or not, and how to perform automatic annotation etc. [17]. It is our belief that a semantic annotation strategy should be tailored to the specific task at hand. In the context of Personalized QA, we employ a similar strategy used in the Knowledge and Information Management (KIM) platform [17], to link annotations to concepts in the domain ontology. The General Architecture for Text Engineering - GATE platform [18] provides a type of Processing Resource called a Gazetteer, which performs gazetteer lists lookup, furthermore linking recognized entities to concepts in ontology based on a mapping list manually defined by users. However, instead of embedding an alias of the instance's URI (Uniform Resource Identifier) as in KIM, we directly embed the label of the most specific class associated with the recognized terms or named entities inline with the texts. For example, the following sentence "*CompanyX releases a new operating system.*" is annotated with "*CompanyX BIOntoCompany releases a new operating system BIOntoSoftware*" whereby *BIOntoCompany*, *BIOntoSoftware* are the labels of ontological concepts *http://localhost/temp/BIOnto#Company*, *http://localhost/temp/BIOnto#Software* respectively. Care is taken while naming the labels by prefixing the concept name with the ontology name, which is "*BIOnto*" in our use case, to make them unique. The rationale for this semantic annotation strategy is as follows:

- Preserving the original terms or named entities e.g. "*CompanyX*" ensures that exact keyword-matching still matches, and, avoids generating noise by over-generation if the original term is completely replaced by its annotation. This ensures that such question as "*Which products did CompanyX release?*" does not get as answer a sentence referring to products related to other companies.
- Embedding the class label directly in the text adds semantic information about the recognized terms or named entities. The annotation BIOntoCompany that follows "*CompanyX*" provides an abstraction that helps to answer such question as "*Which products did the competitors release?*". In this case, the term "*competitors*" in the question is also annotated with *BIOntoCompany*, therefore, a relevant answer can be found even though the question does not mention the company name specifically.
- Based on the concept label, the system can derive the URI of the concept in the domain ontology, query the ontology for relevant concepts and use them to expand the set of query terms of the original question.

Once the documents are annotated, they are split into passages based on the paragraph marker identified by GATE. Each passage is now considered as one document on its own and is indexed in the next step. Before indexing is carried out, stop-word removal is applied to each of the documents. Stop-words are words that do not carry any significant meaning, such as "*the*", "*a*", etc. They are used frequently but do not help to distinguish one document from the others and therefore do not help in searching [19]. Removing these insignificant words makes the indexing and searching task more effective. Porter stemming is also applied to convert the morphological variants of words into their roots.

Document Indexing: The texts within processed documents are fully indexed using Lucene[6], a freely available Information Retrieval (IR) library. Lucene supports Boolean queries based on the well-known *tf.idf* scoring function in IR research. Interested readers are referred to [19] for more details on the scoring formula being used in Lucene.

Question Analysis: Question Analysis involves three text processing tasks: Question typing, Semantic annotation, and Query terms expansion.

[6] http://lucene.apache.org/

Question typing is a common process used in many QA systems. For instance, in [20], question type taxonomy is created to map the questions into their respective types. This helps to bridge the gap between wordings used in the question and those used in the texts, for example, the system is aware that question starting with "*Where*" asks about places so it is typed as "*Location*". However, since domain specific QA system already has the domain ontology in place, instead of building a separate taxonomy for a question type as in [20], a set of pattern-matching rules is built to map the question type to one of the concepts in the domain ontology. Therefore, for a question such as: "*Which products did CompanyX release?*", the question type is BIOntoProduct. The Wh-word and the pronoun that follow are replaced by the question type; and the question becomes "*BIOntoProduct did CompanyX release?*".

There are, however, some special cases in question typing, for instance, from the sample questions from business analysts in our use case, we observe that for "*Yes/No*" questions such as "*Are there any CompanyX's plans to release new products?*" end-users actually do not expect to receive "*Yes*" or "*No*" as an answer but instead the proof that the entities/events of interest exist if they do. Therefore, a set of pattern-matching rules is in place to reform this type of questions to the form of "*What*" question, for the above example it is reformed to "*What are CompanyX's plans to release new products?*" and then the question typing process is carried out as mentioned above. There are also cases whereby the questions cannot be typed to one of the domain concepts. In these cases, question words are removed and the remaining words are treated as a set of query terms.

Once the question is typed, it is also annotated, but in a different manner from the semantic annotation performed on documents. Care is taken so that specific named entities are not annotated with their ontological concepts' label to avoid noise, e.g. attaching a label *BIOntoCompany* after the word "*CompanyX*" in the question will match any terms annotated with *BIOntoCompany* in the document.

Before splitting the questions into query terms and submitting to IR engine, Query terms expansion is performed based on the domain ontology and a set of domain specific synonyms. Initial analysis of the sample questions from the business analyst in the use case indicates two phenomena:

- When the question is typed into a concept in the ontology and that concept has sub-concepts, the question needs expanding with all the sub-concepts in the ontology. Assuming that concept *http://localhost/temp/BIOnto#Product* has sub-concepts *http://localhost/temp/BIOnto#Software* and *http://localhost/temp/BIOnto#Hardware*, the first example question in this section needs to include those two sub-concepts as query terms. This ensures that those entities or terms annotated as *BIOntoSoftware* or *BIOntoHardware* can also be matched during the searching stage.
- End-users tend to use synonyms of verbs specifically to express the same meaning. For example, "*reduce*" and "*lower*" are used interchangeably. There-

fore, synonym lookup is performed against the available synonym set to include them in the set of query terms sent to the search engine.

Performing query terms expansion based on the domain ontology and synonym sets in effect addresses the issue of ambiguity caused by rhetorical writing style used in the source document.

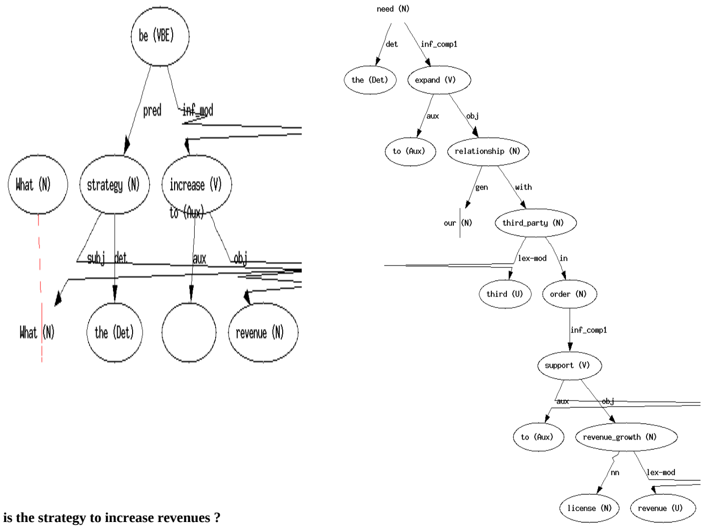
Searching: Lucene is used to search for indexed documents containing the expanded query terms. Boolean query type is used, with AND operator between original terms and OR operator used between expanded terms. Ranked relevant passages returned from the search are fed into the next module, Answer Extraction, to filter out sentences containing query terms whose syntactic structures do not match that of the original question.

4.2 Answer Extraction

In this module, the question is matched with a given candidate answer, which is a sentence derived from passages selected by the Passage Retrieval module. A good review of previous works on Answer Extraction is provided in [1]. Typically, once the sentence has been found by some coarse-grained processing, a set of constraints is applied to check if the candidate sentence is actually the answer. A drawback of the majority of answer extraction techniques is that those techniques such as typical word overlap or term density ranking fail to capture grammatical roles and dependencies within candidate answer-sentences, such as logical subjects and objects [15, 1]. For instance, when presented with the question "*Which company acquired Compaq?*", one would expect to retrieve "*HP*" as an answer to the question. However, typical term density ranking systems would have difficulty with distinguishing the sentence "*HP acquired Compaq*" from "*Compaq acquired HP*". It is concluded that neglecting relations between words can result in a "major source of false positives within the IR" [16].

Answer Extraction modules within systems that involve processing beyond the lexical level, typically conduct pre-annotation of grammatical constraints/ relations of matching questions and candidate sentences [1]. The set of constraints can either be regarded as being "absolute" or as a set of "preferences" [1]. However, the degree of constraint plays an important role in determining the robustness of the system. It is recommended that grammatical constraints must "be treated as preferences and not as being mandatory" [1]. Previous work, such as PiQASso [21] system, shows that strict relations matching suffers substantially from poor recall. Cui et al. [15] propose a solution to the above "strict matching problem" by employing fuzzy or approximate matching of dependency relation using MiniPar[22]. To make this paper self-contained, we provide an overview of the MiniPar Dependency Parser and the dependency tree generated by Minipar.

MiniPar Dependency Parser MiniPar[22] is a fast robust dependency parser which generates dependency trees for words within a given sentence. In a dependency tree, each word or chunked phrase is represented by a node. Each node is linked: one node corresponding to the governor and the other daughter node corresponding to the modifier. The label associated between each link is regarded as a dependency relation between two nodes i.e. *subj, obj, gen* etc. Fig. 2 is generated by MiniPar and illustrates the output dependency parse trees of a sample question-Q1 and sample candidate answer-S1 taken from an extract of Form 10-Q.



Q1 : What is the strategy to increase revenues ?

S1 " the need to expand our relationships with third parties in order to support license revenue growth"

Fig. 2. Dependency trees for Question Q1 and Answer candidate S1

Approximate/Fuzzy relation matching The work of Cui et al. [15] addresses the setback of strict matching between dependency relations. In this work, the authors extract relation paths between two given nodes based on previous work in [23]. A variation of a Statistical Machine Translation model [24] is applied to calculate the probability given candidate sentence and question terms resulting in a match given a combination of relation paths. Mapping scores between relation paths are learned based on two statistical techniques: (1) Mutual Information in order to learn pair-wise relation mappings between questions and candidate answers and (2) Expectation maximization as an interactive training process [24]. Question-answer pairs are extracted from the TREC 8 and 9

QA tasks in order to provide training data. Quantitative evaluation shows that their approach achieves significant retrieval performance when implemented on a range of current QA systems, achieving a MMR 50-138% and over 95% for the top one passage. It is therefore concluded that the approximate dependency relation matching method can boost precision in identifying the answer sentence.

Applications for Personalized Question Answering It is our intention to adapt the above approach to the specific domain of Business Analysis and to integrate it as part of Answer Extraction module for the framework described in Fig. 1. We can utilize questions collected from business analysts and furthermore corpora of Form 10-Q to extract dependency relations using the MiniPar parser in order to generate sample questions-answer pairs for statistical modeling similar to [15]. We believe that this approach in combination with large samples of restricted domain training data will yield high precision while still maintaining high recall.

5 Conclusion and Future work

In this paper we introduce the idea of ”Personalized Question Answering” and propose a framework for its realization. A usage scenario within the Business Analysis domain is also presented. Investigative analysis has highlighted that domain knowledge in the form of formal ontology plays an important role in shaping the approach and design of the aforementioned framework. This is particularly true for semantic annotation and query expansion whereby semantics are needed to address the issue of ambiguity caused by rhetorical writing style used in the source document. Current research indicates that: (1) the inclusion of domain semantics leads to better recall in passage retrieval; (2) in a domain-specific QA system, certain types of questions may require specific analysis (e.g. ”*Yes/No*” questions in this business analysis domain); (3) the use of approximate dependency matching between questions-candidate answer pairs may yield higher precision for answer extraction without impacting on recall.

An application prototype applying Personalized Question Answering framework into Business Analysis use case is being implemented. Once the fully functional prototype is available, a quantitative evaluation scheme will be implemented to gauge the effectiveness of the system within the Business Analysis context. As the system is domain specific, the TREC QA track training data is not suitable for benchmarking, since it is targeted exclusively towards open-domain systems. The evaluation scheme will therefore involve the manual creation of test corpus of business reports, from which given a collection of test questions, Business Analysts will manually extract corresponding question answer pairs. This derived set of question/answer pairs will be used to benchmark the performance of our Personalised Question Answering System.

Future work will also involve prototype functionality enhancement to cater for complex questions whose answers are not explicitly stated, and those that contain implicit contexts. Additional functionality will focus on a caching mechanism for the Question Analysis component to address performance measures for domain frequently asked questions. Last but not least, it is our goal to integrate the QA system with the Analyst Workbench [25] to provide business analysts with an integrated environment to perform business intelligence activity in an effective and timely manner.

Acknowledgments. We would like to thank John Collins, Business Development & Business Engineering Manager, HP Galway and Mike Turley, CEO of DERI, for discussions on business analysis problem. We also thank the anonymous reviewers for their constructive comments. This work is supported by Science Foundation Ireland(SFI) under the DERI-Lion project (SFI/02/CE1/1131).

References

1. Hirschman, L., Gaizauskas, R.: Natural language question answering: the view from here. Nat. Lang. Eng. **7** (2001) 275–300
2. Berger, H., Dittenbach, M., Merkl, D.: An adaptive information retrieval system based on associative networks. In: APCCM '04: Proceedings of the first Asian-Pacific conference on Conceptual modelling, Darlinghurst, Australia, Australia, Australian Computer Society, Inc. (2004) 27–36
3. Chung, H., Song, Y.I., Han, K.S., Yoon, D.S., Lee, J.Y., Rim, H.C., Kim, S.H.: A practical qa system in restricted domains. In Aliod, D.M., Vicedo, J.L., eds.: ACL 2004: Question Answering in Restricted Domains, Barcelona, Spain, Association for Computational Linguistics (2004) 39–45
4. Sneiders, E.: Automated question answering using question templates that cover the conceptual model of the database. In: NLDB '02: Proceedings of the 6th International Conference on Applications of Natural Language to Information Systems-Revised Papers, London, UK, Springer-Verlag (2002) 235–239
5. Bernstein, A., Kaufmann, E., Fuchs, N.E., von Bonin, J.: Talking to the semantic web a controlled english query interface for ontologies. In: 14th Workshop on Information Technology and Systems. (2004) 212–217
6. Lopez, V., Pasin, M., Motta, E.: Aqualog: An ontology-portable question answering system for the semantic web. In: ESWC. (2005) 546–562
7. Zhang, Z., Sylva, L.D., Davidson, C., Lizarralde, G., Nie, J.Y.: Domain-specific qa for construction sector. In: Proceedings of SIGIR 04 Workshop: Information Retrieval for Question Answering. (2004)
8. Demner-Fushman, D., Lin, J.: Knowledge extraction for clinical question answering: Preliminary results. In: Proceedings of the AAAI-05 Workshop on Question Answering in Restricted Domains, Pittsburgh, Pennsylvania. (2005)
9. Niu, Y., Hirst, G.: Analysis of semantic classes in medical text for question answering. In: Proceedings of 42st Annual Meeting of the Association for Computational Linguistics, Workshop on Question Answering in Restricted Domains. (2004) 54–61
10. Molla, D., Schwitter, R., Rinaldi, F., Dowdall, J., Hess, M.: Extrans: Extracting answers from technical texts. IEEE Intelligent Systems **18** (2003) 12–17
11. TREC: Text retrieval conference trec http://trec.nist.gov (2005)

12. Andrenucci, A., Sneiders, E.: Automated question answering: Review of the main approaches. In: ICITA '05: Proceedings of the Third International Conference on Information Technology and Applications (ICITA'05) Volume 2, Washington, DC, USA, IEEE Computer Society (2005) 514–519
13. Dumais, S., Banko, M., Brill, E., Lin, J., Ng, A.: Web question answering: Is more always better? In: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2002), Tampere, Finland. (2002)
14. Diekema, A.R., Yilmazel, O., Chen, J., Harwell, S., Liddy, E.D., He, L.: What do you mean? finding answers to complex questions. In: Proceedings of the AAAI Spring Symposium: New Directions in Question Answering. Palo Alto, California. (2003)
15. Cui, H., Sun, R., Li, K., Kan, M.Y., Chua, T.S.: Question answering passage retrieval using dependency relations. In: SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, ACM Press (2005) 400–407
16. Tellex, S., Katz, B., Lin, J., Fernandes, A., Marton, G.: Quantitative evaluation of passage retrieval algorithms for question answering. In: SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, New York, NY, USA, ACM Press (2003) 41–47
17. Kiryakov, A., Popov, B., Terziev, I., Manov, D., Ognyanoff, D.: Semantic annotation, indexing, and retrieval. Journal of Web Semantics **2** (2005) 39
18. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: Gate: A framework and graphical development environment for robust nlp tools and applications. In: Proceedings of the 40th Annual Meeting of the ACL. (2002)
19. Hatcher, E., Gospodnetic, O.: Lucene in Action. Manning Publications Co. (2005)
20. Hovy, H., Gerber, L., Hermjakob, U., Lin, C., Ravichandran, D.: Towards semantic-based answer pinpointing (2001)
21. Attardi, G., Cisternino, A., Formica, F., Simi, M., Tommasi, A.: Piqasso: Pisa question answering system. In: Text REtrieval Conference. (2001)
22. Lin, D.: Dependency-based evaluation of minipar. In: Proc. Of Workshop on the Evaluation of Parsing Systems, Granada, Spain. (1998)
23. Gao, J., Nie, J.Y., Wu, G., Cao, G.: Dependence language model for information retrieval. In: SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, ACM Press (2004) 170–177
24. Brown, P.F., Pietra, S.D., Pietra, V.J.D., Mercer, R.L.: The mathematic of statistical machine translation: Parameter estimation. Computational Linguistics **19** (1994) 263–311
25. O'Riain, S., Spyns, P.: Enhancing business analysis function with semantics. In Meersma, R., Tari, Z., eds.: On the Move to Meaningful Internet Systems 2006: CoopIS, DOA, GADA and ODBASE; Confederated International Conferences CoopIS, DOA, GADA and ODBASE 2006 Proceedings. LNCS 4275, Springer (2006) 818–835