Evaluating Ontology Cleaning

Christopher Welty, Ruchi Mahindru, and Jennifer Chu-Carroll

IBM T.J. Watson Research Center
19 Skyline Dr.
Hawthorne, NY 10532
{welty,jencc}@us.ibm.com, ruchi_mahindru@yahoo.com

Abstract

Ontology as a discipline of Computer Science has made many claims about its usefulness, however to date there has been very little evaluation of those claims. We present the results of an experiment using a hybrid search system with a significant knowledge-based component to measure, using precision and recall, the impact of *improving the quality of* an ontology on overall performance. We demonstrate that improving the ontology using OntoClean (Guarino and Welty, 2002), does positively impact performance, and that having knowledge of the search domain is more effective than domain-knowledge-free search techniques such as link analysis.

Content Areas: Ontologies

Introduction

Ontologies have been proposed as a solution to numerous problems in areas such as search, semantic integration, agents, human-computer interaction, etc. The premise is usually that a clear, high-quality ontology can act as an *interlingua* in which mappings between systems can unambiguously be expressed, or in which understanding of a domain can be gained by a system or shared by users (Smith and Welty, 2001).

For the most part, this claim has not been realized in practice, and one reason for this failure is that there was little agreement on just what makes an ontology "clear" or "high quality." One recent development in ontology research has been the specification of a formal methodology for ontological analysis, OntoClean (Guarino and Welty, 2002), that addresses the problem of defining just what "high quality" is for ontologies. Following this definition and approach, a high quality foundational ontology, Dolce, is being developed (Gangemi, et al, 2002). While OntoClean appears to be widely accepted in the scientific community, and OntologyWorks, a small company providing database integration services, has a proprietary analysis tool based on OntoClean (www.ontologyworks.com), there is very little evidence that it can have impact on knowledge-based systems. In fact, there appears to be a significant obstacle in understanding the methodology, and even without this "learning curve", significant manual effort must be expended to employ the methodology to develop actual "clean" ontologies. Furthermore, there has been no clear argument that such an expenditure will pay for itself in the long run. Indeed, "Why does it matter?" has been the most frequent criticism of the OntoClean approach.

We report here on a series of experiments using a hybrid search system with a significant knowledge-based component to test the impact of improving the quality of ontologies on system performance. The use of search as a test system provides a well understood framework for empirical evaluation, and gives an excellent opportunity to address the "Why does it matter?" question.

Preliminary results of this experiment were presented earlier in a workshop (Welty, et al, 2003). In this paper we present the final data for the full experiment, plus a more detailed analysis of the results.

Background

The field of ontology has been sorely lacking in formal empirical analysis in general. This deficiency has been recognized, however, and a few examples are starting to appear. Ablation analysis of ontologies at U.T. Austin (Fan, et al., 2003) is one example in which elements of an ontology used by a search system that relied on recognizing noun compounds were systematically removed to determine which level of specificity were most relevant to the results. There have also been several evaluations of the impact of structured knowledge (loosely construed as ontologies) on IR and search systems in general.

Most closely related to this work is the work of Clark, et al, at Boeing (2000), in which a search system was enhanced by employing an ontology-like artifact (a thesaurus of terms with more meaningful structure than a flat list of keywords).

This work showed that precision and recall performance of a retrieval system can be significantly increased by adding this kind of information. It is important to note that while Clarke, et al, did discuss a process for improving the quality of the ontology; they did not formally evaluate the impact of the *improvement*. Furthermore, Kanisa, Inc. (previously AskJeeves Enterprise Solutions – www.kanisa.com) has based their business on providing domain-specific knowledge-enhanced search, and has been turning a profit since 1Q 2002 (Ulicny, 2003).

Similar evaluations of the impact of ontologies on search-based systems have been done in the question-answering community. Moldovan and Mihalcea (2000) use a significantly enhanced version of WordNet to drastically improve question answering performance, and other groups including Woods, et al (2000), Hovy et al (2001), and Prager, et al (2001), have reported similar results. Again, as with the Boeing work, these groups report positively on the impact of adding an ontology to a search system, but make no attempt to determine whether a good quality ontology would improve performance more. In fact, within the IR and QA communities, WordNet is the most common ontology-like artifact to employ, and previous work has shown that WordNet viewed as an ontology is not particularly of high quality (Oltramari, et al, 2002).

System Overview

The RISQUE system is an evolution of the system reported in (Chu-Carroll, et al, 2002). This system provides a natural-language front end to a conventional search engine, but uses clues in the natural language question, a knowledge-base of industry terms, and knowledge of the web site structure (see below) to construct an *advanced search query* using the full expressiveness of the search engine query language. This search is limited to a corporate web site, in this case our knowledge is of the ibm.com buy and support pages for the ThinkPad™ and NetVista™ product lines.

The main components of RISQUE include a parser, the terminology knowledge base, rules for question types, a hub page finder, a query formulation component, and the search engine.

The parser is a slot-grammar parser (McCord, 1980) that must be seeded with multi-word industry specific terms, so that e.g. “disk drive” will be parsed as a compound noun, rather than a head noun with a pre-modifier. These multi-word terms come from the knowledge-base. From the grammatical structure of the question, we extract the primary verb phrase and the noun phrases. The verb phrase information is the main evidence used to fire rules for recognizing question types, which themselves depend on the web site structure. The ibm.com web site, like many enterprise sites, is divided into a section for support and a section for sales. This gives RISQUE its two most basic question types, “buy” and “support”.

The hub-page finder is a system of declarative rules that takes the noun phrases from the question and determines whether they correspond to products listed in the terminology, and if so finds the most appropriate “hub page” or “comparison page” for that product. For example, many questions about IBM Thinkpads can be answered with information on the “ThinkPad home page” on the IBM site. Directing users to these key pages is often the quickest path to an answer. Hub and comparison pages are described in the next section. The rules are broken into two parts, one set is derived directly from the knowledge base and includes *moreImportantThan* relationships and the taxonomy; the second includes rules expressing the relationships between linguistically-derived information and the hub pages. For example, if the question contains a superlative, as in “What is the fastest Thinkpad?”, the rules indicate that the Thinkpad comparison page should be returned.

The system iteratively generates complex queries using knowledge of web site structure. The query may include URL restrictions, such as “only consider pages with 'support' in the URL for support questions”, or “exclude pages with research.ibm.com in the URL for buy questions”. The query will also make use of boolean connectives, disjunction to support synonym expansion, and conjunction of noun phrase terms. If this query does not return enough hits, the query formulation component will relax the query according to a number of heuristics, such as dropping the least important noun phrase. Knowledge of which terms are more important than others, based on manual analysis of the web site, is included in the knowledge-base. RISQUE iteratively formulates search queries and accumulates results until a pre-determined number of result pages are obtained.

The RISQUE system was tested and trained with a set of questions made up by team members. For the initial experiment described in (Welty, et al, 2003), the system was evaluated with questions made up by a domain expert from outside the group. Finally, the system was evaluated with roughly 200 questions collected from students and professors at CCNY.

Role of Ontology

The central terminology of RISQUE is an ontology-like knowledge-base of industry terms arranged in a taxonomy according to specificity. In addition to the taxonomy, the knowledge base includes important information used by the system:

Hub page: Most terms at the top level of the taxonomy have a corresponding "hub page" – a page that gives a general description of the things in that category. For example, there is a hub page for IBM ThinkPad, and also a hub page for IBM T-Series ThinkPad. The ibm.com website, along with most e-commerce websites, are designed to pack a lot of information in these particular pages, with links to as much information as the designers can imagine might be relevant to someone seeking support or seeking to purchase. The taxonomy is used to associate terms with *the most specific* hub page that is relevant. For example, if we know that a "ThinkPad A21" is a "ThinkPad A-Series Model" which itself is a "Thinkpad", then if only the latter two terms have hub pages, the hub page for "Thinkpad A21" would be the "Thinkpad A-Series" hub page – the most specific hypernym that has an associated hub page.

Comparison Page: Many e-commerce web sites including IBM's provide the ability to compare two or more similar products. Our knowledge-base stores information on how to find or generate comparison pages for products. These pages will be displayed for questions like, "What is the fastest A-Series ThinkPad?" Similar to hub pages, the taxonomy is used to associate terms with the *most specific common* comparison page.

Synonyms: Synonyms account for simple variations on spelling, acronyms, abbreviations, etc., as well as traditional synonyms. This information is used to find the term being referenced in a question, as well as in query expansion. The use of synonyms in query expansion made the notion of an *expansion* (see below) more important.

moreImportantThan: e-Commerce websites have an organization that is important to capture in interpreting questions. For example, IBM's web site is organized such that add-on accessory pages list which models they are compatible with, but computer pages do not list which accessories are compatible with them. This knowledge is explicit and intentional for the website maintainers, but is not necessarily obvious to a customer browsing the site for the first time. Thus, when an accessory and a computer are mentioned in the same question, such as, "What CD drive goes with my ThinkPad T23?" we consider the CD drive to be the more important term in the question. The more important term in the question will have its hub page returned in a higher position, and the less important term may, in some circumstances, not have its hub page appear at all. In addition, the less important term will be dropped first during query relaxation. The *moreImportantThan* relation is considered to be transitive, and is also inherited down the taxonomy. Thus we only represent in the knowledge-base that accessories are *moreImportantThan* computers, and from this we infer that "CD drive" is *moreImportantThan* "ThinkPad T23."

Expansions: An interesting situation that we had to account for in dealing with questions generated by non-experts was that often people are confused about what industry terms mean. For example, many people think "SCSI" is a kind of disk drive, when in fact is it a type of communications bus. These types of errors do not appear in the web pages, thus making SCSI a simple synonym of "disk drive" would not be productive – synonyms are used in query expansion and therefore searches for disk drives would turn up communication bus technology pages. To solve this, the *expansion* relation between terms is treated as an asymmetric synonym. When "SCSI" appears in a query, it will be considered a synonym of disk-drive, however when disk-drive appears in a query, it will not be considered a synonym of "SCSI".

Clean-up Process

The original RISQUE system terminology, Quilt, was developed by domain experts with no experience with or knowledge of ontology engineering methods, and contained on the order of 3K synsets and 4.6K terms. We improved this terminology in a number of ways:

1. Developed a "backbone taxonomy" of terms
2. Analyzed terms and their position in the hierarchy
3. Organized terms more logically
4. Ensured every term was grounded in the top level
5. Ensured terminology was logically consistent

We used three tools in performing this cleanup: The OntoClean methodology was used in analysis, and helped with #1-3; an ontology editor was used to view the taxonomy, this was critical in #2-4; a reasoner was used to ensure consistency and coverage of the

moreImportantThan relation. The analysis and cleanup took on the order of one person-week, and resulted 3K synsets and 10.8K terms.

The main cleanup effort was establishing a *backbone taxonomy* (Guarino and Welty, 2001) of roughly 30 terms. The backbone taxonomy terms represent the terms with the highest organizational utility. They cover the entire domain of discourse: every entity must instantiate a backbone class. The backbone proved a critical resource in organizing the terms, as our requirement was that all other terms "ground up" through hypernym links to one and only one backbone term – the backbone terms were mutually disjoint.

That isn't to say there weren't terms with multiple hypernyms, but no term had more than one ancestor hypernym in the backbone. This was a critical organizational policy that helped keep the taxonomy clean. Analyzing 3,000 terms for their formal meta-properties proved daunting, however it was easy to find terms grounded in more than one backbone term. These terms were analyzed more carefully and in most cases were found to exhibit one of the common modeling pitfalls outlined in (Guarino and Welty, 2002). The most common of these was the confusion between *part-of* and *subclass*. For example, a purpose-built keyboard for a small hand-held computer is not a more specific term than the computer. It is an accessory for that computer.

The initial Quilt taxonomy contained a large number of inconsistent, meaningless, redundant and disconnected terms. These terms were found using an ontology editor and reasoner, and either eliminated or connected properly to the backbone. It is important to note that the cleaned taxonomy did not have any new synsets, other than a few in the relatively small backbone taxonomy. We did, however, find considerable inconsistent usage of synonym patterns. For example, the "Thinkpad T21" might have "T21" as a synonym, however the "Thinkpad T23" might have "TP 23" as a synonym. These abbreviation patterns were normalized with a program, accounting for the large increase in the number of terms. Surprisingly, this increase did not end up impacting the evaluation significantly, as the questions tended to use the full names of computer models when present.

Experimental Setup

Although the main goal of RISQUE was to show an improvement over traditional web search, the particular experiment described in this paper was to isolate the process of improving the quality of the terminology using ontology-based analysis tools. The RISQUE system architecture treats the terminology as a pluggable module, which allowed us to isolate this particular change while holding all other aspects of the system constant. We then concentrated on how to compare a poorly structured terminology with a cleaned one. After the cleanup was complete, we performed four evaluations as follows:

- *baseline:* basic search over the IBM web pages using a traditional search engine
- *quilt:* The full RISQUE system with the original Quilt terminology
- *clean:* The full RISQUE system with the cleaned terminology
- *google:* Basic search using Google restricted to the ibm.com web pages

The evaluation was performed on 200 natural language questions about IBM products collected mainly from students and faculty at CCNY. Participants were asked to imagine they wanted to purchase something from IBM, or get support for an existing product. The experiments were run against the live ibm.com website, over which we had no control. As a result, we ran the evaluations in parallel, with each question running through all four systems at the same time, in order to prevent changes in the web site from impacting performance of one system in isolation. The google and baseline queries were formulated manually from a conjunction of all the words in the noun phrases from the natural language question. The answer to a question was considered correct if one of the pages in the top ten returned by the search contained an answer to the question – i.e. the answer is on the returned page or a single click (and some reading) away. For comparison questions, e.g. "What is the fastest desktop?", or "What is the lightest ThinkPad?" the answers were considered correct if the comparison page selected by RISQUE contained the relevant data for each type of computer, e.g. the processor speed of each desktop model, or the weight of each ThinkPad model.

Results and Analysis

Our results are shown in the table below. Each experiment lists the number correct (of 200) and the percent.

	# correct	% correct	% improvement
Base	65	32%	
Quilt	105	52%	63%
Clean	121	61%	over baseline: 91% over Quilt: 17%

Google	43	22%	-31%

First of all, our results confirm the overwhelming evidence to date that ontologies can significantly improve search results. In the original version of RISQUE, which utilizes the pre-cleaned-up version of the ontology, Quilt, we achieved a 63% relative improvement over our baseline system. This improvement is obtained as a combination of RISQUE's iterative query formulation process, the hubpage finder, and RISQUE's query expansion feature, all of which utilize the set of known industry terms, and their synonyms/expansions provided by the ontology.

Most notably, our results show a clear improvement of the search results when using the higher quality "cleaned" terminology, which shows a 17% relative improvement over the original terminology and, as utilized by RISQUE's processing components, nearly doubles the performance of the baseline system. While the improved terminology contained more actual words, this expansion did not in itself account for the increase in precision – as noted above, questions tended to use the full names of products, rather than abbreviations. This expansion of terms would likely have more of an impact on actual usage of the system, but was not relevant to our evaluation of ontology cleaning.

Many of the correct answers in our system come from hub and comparison pages, so the fact that terms are more consistently connected through the taxonomy with these pages in the cleaned terminology was the major reason the cleanup improved precision. Another important factor was the proper derivation of the *moreImportantThan* relation between terms, which was incorrect or ambiguous in a number of cases in the original search because of missing links in the taxonomy.

The heavy reliance of our system on "hub pages" for correct answers would seem to indicate that link analysis, or a similar technique that ranks highly connected pages over less connected ones, would improve search considerably given the large number of incoming and outgoing links on these pages. If effective, such a technique would clearly be preferable over a knowledge-base, since it requires significantly less manual effort to maintain. This led us to perform an experiment using the Google™ search engine restricted to the ibm.com website. We were very surprised to find that this experiment was the worst performer of all.

The important difference between the knowledge-based search and one based on link analysis is knowledge of the structure of the website, as reflected e.g. in the *moreImportantThan* relation. As discussed above, one of the things captured in the relation is the fact that information about compatibility is located on accessory pages, not the computer pages. Thus the highly-connected ThinkPad hub pages receive high scores from Google™ for questions like, "What modem goes with my ThinkPad t30?", but they do not contain an answer to the question. In fact, the correct page contains so few references to Thinkpad t30 that it does not show up in the top ten, and furthermore for any query that contains the word "thinkpad" google search will return the Thinkpad hub page, as it contains that term numerous times and is very highly connected. This page is typically not the answer for most questions, and our internal hub page finder *knows* this. We believe that explicit knowledge, though expensive to maintain, will always win out over heuristics like link analysis.

These results are consistent with the results previously reported in (Welty, et al, 2003), however the limitations of that experiment, in which questions were taken from a domain expert, were overcome by using students who did not know anything about the experiment being performed or the system being evaluated. The overall precision of our searches decreased in this experiment over the previous, indicating that the difference in subjects did impact the system, however the relative improvements across the three principle evaluations remained roughly the same.

The fourth evaluation, for google, dropped significantly in relative precision; in the previous experiment the google evaluation showed only a 3% relative decrease in precision from the base. In this experiment, google showed a 31% relative decrease in precision over the baseline. The main cause of this decrease was that there were significantly more questions about accessories and comparisons, for which (as described above) link analysis and keyword ranking were poor heuristics.

Conclusion

We described a system for knowledge-based natural language search called RISQUE, and focused on the knowledge-based components and their role in the system. We performed a controlled experiment to compare the precision of the search system with an unprincipled ontology to the same system with a principled ontology. Our results showed an 17% relative improvement in precision (from 52% to 61%) with no other changes in the system other than

applying the OntoClean methodology to analyzing the ontology and cleaning it.

These results provide some evidence that improving the quality of an ontology does improve the performance of an ontology-based search. It stands to reason that any system that has a significant ontology component would benefit from improving the ontology portion. It is our further claim that as systems with an ontology component begin to employ more sophisticated reasoning, improvements in ontology quality will have even more impact. This is the focus of our current investigations.

In light of the results reported in the ablation studies described in (Fan, et al, 2003), in which it was shown that the higher, more abstract, levels of an ontology have significantly more impact on system performance than lower ones, we can address the common complaint about OntoClean, that it takes too long to perform detailed meta-property analysis on large ontologies. Accepting the results of these two empirical evaluations as general indicators, we conclude that focusing this detailed analysis on the top level would have the highest cost-benefit, and could easily justify the expense.

Acknowledgements

Other contributors to the Risque system were John Prager, Yael Ravin, Christian Lenz Cesar, David Ferrucci, and Jerry Cwiklik. Our thanks to Mike Moran and Alex Holt for supporting the project. This work was supported in part by the Advanced Research and Development Activity (ARDA)'s Novel Intelligence for Massive Data (NIMD) Program under contract number MDA904-01-C-0988.

References

Chu-Carroll, J., John Prager, Yael Ravin and Christian Cesar. 2002. A Hybrid Approach to Natural Language Web Search. *Proceedings of EMNLP-2002*.

Clark, P., J. Thompson, Heather Holmback, and Lisbeth Duncan. 2000. Exploiting a Thesaurus-Based Semantic Net for Knowledge-Based Search. *Proceedings of IAAI-2000*. Pp. 988-995. Austin:AAAI Press.

Fan, J., K. Barker and B. Porter. 2003. The Knowledge Required to Interpret Noun Compounds. *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*. Acapulco:AAAI Press.

Gangemi A., N. Guarino, C. Masolo, A. Oltramari, L. Schneider. 2002. Sweetening Ontologies with DOLCE. *Proceedings of EKAW 2002*. Siguenza:Springer.

Guarino, N. and C. Welty. 2000. A Formal Ontology of Properties. In, Dieng, R., and Corby, O., eds, *Proceedings of EKAW-2000: The 12th International Conference on Knowledge Engineering and Knowledge Management.* Spring-Verlag LNCS Vol. 1937:97-112.

Guarino, Nicola and Chris Welty. 2002. Evaluating Ontological Decisions with OntoClean. *Communications of the ACM.* **45**(2):61-65. New York: ACM Press.

Hovy, E, L. Gerber, U. Hermjakob, C.-Y. Lin, and D. Ravichandran. 2001. Toward Semantics-Based Answer Pinpointing. In *Proceedings of the DARPA Human Language Technology Conference (HLT)*. San Diego, CA.

McCord, Michael C. 1980. Slot grammars. *American journal of Computational Linguistics*, **6**(1):255-286, January 1980.

Moldovan, D. and Rada Mihalcea. 2000. Using WordNet and Lexical Operators to Improve Internet Searches. *IEEE Internet Computing*. **4**(1): 34-43.

Oltramari, A., Aldo Gangemi, Nicola Guarino, and Claudio Masolo. Restructuring WordNet's Top-Level: The OntoClean approach. *Proceedings of LREC2002 (OntoLex workshop)*. Las Palmas, Spain

Prager, J., Jennifer Chu-Carroll, and Krzysztof Czuba. Use of WordNet Hypernyms for Answering What-Is Questions. *Proceedings of TREC 2001*.

Smith, B. and C. Welty. 2001. Ontology: Towards a new synthesis. In *Formal Ontology in Information Systems*. Ongunquit:ACM Press.

Ulicny, B. Putting your customers' answers to work. 2003. Keynote address, *AAAI Spring Symposium on New Directions in Question Answering*.

Welty, C., R. Mahindru, and J. Chu-Carroll. 2003. Evaluating Ontological Analysis. In *Proceedings of the ISWC-03 Workshop on Semantic Integration.* Sanibel Island.

Woods, W., Stephen Green, Paul Martin, and Ann Houston. Halfway to Question Answering. Proceedings of *TREC 2000*.