Web-Annotations for Humans and Machines

Norbert E. Fuchs[1] and Rolf Schwitter[2]

[1] Department of Informatics & Institute of Computational Linguistics
University of Zurich, Switzerland
fuchs@ifi.unizh.ch
[2] Department of Computing & Centre for Language Technology
Macquarie University, Australia
rolfs@ics.mq.edu.au

Abstract. We propose to manually annotate web pages with computer-processable controlled natural language. These annotations have well-defined formal properties and can be used as query relevant summaries to automatically answer questions expressed in controlled natural language, and as the basis for other forms of automated reasoning. Last, but not least, the annotations can also serve as human-readable summaries of the contents of the web pages. Arguably, annotations written in controlled natural language can bridge the gap between informal and formal notations and leverage true collaboration between humans and machines. This is a position paper that proposes a solution combining existing methods and techniques to achieve a highly relevant practical goal, namely how to effectively access information on the web. However, our solution introduces a "chicken and egg" problem: a critical mass of web annotations will be necessary that people perceive the value of these annotations and start annotating web pages themselves. Only the future will show whether this – basically non-technical – problem can be solved.

Keywords: Annotations, Controlled Natural Languages, Semantic Web, Question Answering, Automated Reasoning

1 Getting Your Questions Answered – Or Perhaps Not

1.1 The Problem

When visiting restaurants in Sydney you notice that many dishes contain capers, and you may ask yourself "Does this Mediterranean plant perhaps grow in Australia?" Asking the service personnel remains inconclusive, and you eventually turn to Google with the query

Are capers grown in Australia?

and get more than 74'000 references. Realising that Google gives you references to all web pages that contain the keywords of your query in any order and any context, you let Google search for the exact phrase

"*Are capers grown in Australia?*"

and receive no answer at all. Recalling that Google orders its results according to their rank you select the top result of the 74'000 references found for your first query. This top result refers to an interview of the Landline[1] program of the Australian Broadcasting Corporation. Perusing the complete interview of 2000 words you find that the text of the interview nowhere explicitly states

Capers are grown in Australia.

which would exactly answer your question. Instead there are text snippets containing variants of and references to the words "capers", "grow" and "Australia" like

- *... a young South Australian couple decided they could grow capers in this country. …*
- *… So they sought out some plants and now boast Australia's first and only commercial caper crop. …*
- *… "No-one grows capers in Australia" ...*
- *... as Australia's first and only commercial growers of capers. …*
- *... because they've never been grown in Australia before …*
- *... that we can grow capers in Australia …*
- *... Australia's first home-grown capers and caper-berries. …*
- *.... they could be grown in Australia ...*

These text snippets do not readily help you to answer your question, are rather confusing, perhaps even contradictory. Absorbing the complete contents of the interview, and applying unspecified world-knowledge, you eventually infer that capers are experimentally grown in South Australia.

A little frustrated, you wished that a search engine would be able to automatically find a satisfying answer to your query without you having to extract the answer from lengthy documents.

So what can be done to automatically find an answer to your question on the web? Note that a solution to this problem is intimately related to another problem. Which answer do you actually expect? Which answer would you accept as satisfactory?

1.2 Question Answering

One approach to answer questions has been investigated by researchers of the language engineering community. To this community question answering (QA) systems are of great interest because they combine information retrieval (IR), natural language processing (NLP), and often machine learning (ML) within the same task. QA systems

- receive natural language queries as input – not keywords,
- process large unstructured document collections – usually not web pages,
- return precise answers as output – not (references to) documents.

Though the fields IR, NLP and ML have seen spectacular progress in recent years, a sobering realisation must be made – there seems to be a ceiling of what can be

[1] www.abc.net.au/landline/content/2006/s1602940.htm

achieved. Here is a representative current result. The best values cited in the "Overview of the TREC 2005 Questions Answering Track"[2] are 71% accuracy, 64% precision and 53% recall for answering mainly simple factoid questions. Though new methods may bring some improvements, we believe that no real breakthrough can be expected, and that eventually automatic methods must be complemented by human intervention to get better results. This echoes our experience that interpreting the Landline interview required additional knowledge not found in the interview itself.

1.3 Automatic Summarisation

An alternative approach has been to automatically summarise documents, and – among other things – to use summaries of documents instead of the documents themselves to answer questions. Summaries can be generated simply by extraction, i.e. by copying relevant information of the document into the summary, or by abstraction, i.e. by paraphrasing and condensing the contents of the document. Though there does not seem to be a consensus on evaluation methods, the results of automatic summarisation are not more encouraging than those of question answering[3], and again we have to realise that human intervention would eventually be required to improve the results. Interestingly, Hovy [1] writes in this context

> *… Since the result [of summarisation] is something new not explicitly contained in the input this stage [of summarisation] requires that the system have access to knowledge separate from the input. …*

Again, we encounter the situation that additional knowledge is required to understand a text.

1.4 Semantic Web

Another approach – aimed directly at web pages – is taken by the semantic web[4] that states as its goals

> *The Semantic Web is about two things. It is about common formats for interchange of data, where on the original Web we only had interchange of documents. Also it is about language for recording how the data relates to real world objects. That allows a person, or a machine, to start off in one database, and then move through an unending set of databases which are connected not by wires but by being about the same thing.*

These goals are to be achieved by languages like RDF[5] and OWL[6]

---

[2] trec.nist.gov/pubs/trec14/papers/QA.OVERVIEW.pdf
[3] acl.ldc.upenn.edu/E/E99/E99-1011.pdf
[4] www.w3.org/2001/sw
[5] www.w3.org/RDF
[6] www.w3.org/2004/OWL

… the World Wide Web Consortium released the Resource Description Framework (RDF) and the OWL Web Ontology Language (OWL) as W3C Recommendations. RDF is used to represent information and to exchange knowledge in the Web. OWL is used to publish and share sets of terms called ontologies, supporting advanced Web search, software agents and knowledge management.

Like Katz and Lin [2] we see two main problems of the semantic web. The first problem is

... to transform existing sources (stored in HTML pages, in legacy databases etc.) into a machine-understandable form (i.e. XML/RDF/OWL) ...

which is hard to do automatically since the transformation involves hurdles similar to those encountered in automatic question answering and automatic summarisation.

The second problem is that this transformation

... is sometimes at odds with a human-based natural language view of the world.

concretely, that languages like RDF and OWL are intended for computers, not for humans.

To solve the second problem natural language front-ends have been proposed. Within the Metalog[7] project of W3C, Marchiori and collaborators developed the language PNL ("Pseudo Natural Language") that they describe as follows

The goal of the Metalog's PNL ("Pseudo Natural Language") is to define a technology that is very close to the people, even if this possibly means sacrificing part of the expressive power of the underlying tower (in other words, to start filling up the upper parts of the P axis). The PNL, as the name says, aims to use a very colloquial form of communication, that is very close to humans: natural language.

and give the example

JOHN and MARY OWN a "red house".

that – capitalising some constituents and putting others in quotes – can hardly be called natural. Incidentally, the example is also ambiguous as to whether John and Mary own together one house, or individually two houses.

Alternative approaches to bridge the gap between the languages of the semantic web and natural language are offered within the Attempto project [3]. There is a bidirectional translation between Attempto Controlled English (ACE) – a subset of standard English equivalent to full first-order logic – and OWL DL that allows users to interface OWL ontologies in ACE without having to know the languages OWL, RDF or XML.

A slightly different approach is proposed by Schwitter and Tilbrook [4] who use a controlled natural language to directly describe knowledge of the semantic web without taking recourse to RDF.

[7] www.w3.org/RDF/Metalog

1.5 Augmenting RDF by Natural Language Annotations

To render RDF friendlier to humans and to facilitate question answering from web pages and data bases, Katz and his collaborators [2] propose to augment RDF with natural language annotations. Using these techniques they have developed the Natural Language Question Answering System START[8] that uses pattern matching to answer natural language questions from a variety of sources.

> *Unlike information retrieval systems (e.g., search engines), START aims to supply users with "just the right information," instead of merely providing a list of hits. Currently, the system can answer millions of English questions about places (e.g., cities, countries, lakes, coordinates, weather, maps, demographics, political and economic systems), movies (e.g., titles, actors, directors), people (e.g., birth dates, biographies), dictionary definitions, and much, much more.*

While the START system is rather impressive and answers questions from a wide range of sources, we believe that the fixed patterns of the RDF annotations employed by START are too rigid and too restrictive to anticipate the diversity of questions users may ask.

1.6 Annotations in Controlled Natural Language

Realising that attempts to automatically answer questions from legacy texts and web pages have encountered fundamental problems, we propose a radical solution, namely to have humans annotate web pages in a way that facilitates question answering. Concretely, we propose to augment web pages with annotations in a controlled natural language [4]. This proposal offers the following advantages:

- annotations in computer-processable controlled languages permit formal reasoning, specifically question answering by deduction,
- question answering from annotations in controlled natural languages can easily be supported by the necessary linguistic and domain-specific background knowledge,
- annotations in controlled natural languages are readable by anybody, and thus can also serve as a summary of the respective web page,
- annotations in controlled natural languages can be written according to standard guidelines of good summary writing, for instance Wikipedia's guidelines for lead sections[9].

In a similar approach [5] propose to annotate scientific publications with summaries in controlled natural languages, and point out that these annotations can be used for question answering and a number of additional reasoning tasks.

Here is a possible annotation to the above-mentioned Landline interview on capers that contains just a small amount of the factual knowledge of the interview.

A couple grows capers in Australia.

---

[8] start.csail.mit.edu/

[9] en.wikipedia.org/wiki/Wikipedia:Lead_section

This annotation is written in a controlled natural language [6, 7] similar to ACE or PENG[10], and allows us to answer our question by deduction

Are capers grown in Australia?

provided that we take into account the linguistic knowledge relating the active and the passive forms of the transitive verb *grow*.

So we are already done, are we?

In fact, we are not done at all since we skillfully crafted the annotation in a way that allowed us to more or less immediately deduce our question, and we carefully sidestepped a number of serious problems of this approach.

Foremost, there is a "chicken and egg" problem [2] that similarly affects our approach, the semantic web and the START project, namely

> *… people will not spend extra time marking up their data unless they perceive a value for their efforts, and metadata will not be useful until a "critical mass" has been achieved.*

Only the future will show whether this – basically non-technical – problem can be solved, and we will not discuss it further here.

Furthermore, there are technical problems that are specific to our approach. We will address these problems in the remainder of this paper. In section 2 we describe controlled natural languages and in section 3 how these languages can be used to annotate web pages. Section 4 discusses guidelines for writing good annotations. In section 5 we outline how annotations can be added to web pages in a way that benefits both humans and machines. Section 6 is dedicated to question answering on the basis of annotations, to the need for linguistic and other background knowledge, and to the problems that arise when annotations are inconsistent, incomplete, on different levels of detail, using different conceptualisations, or are plainly not understandable. Here we also discuss possible solutions to these problems. Section 7 suggests some alternative uses of annotations. In section 8 we summarise the main ideas and the advantages of our approach.

2 Controlled Natural Languages

Formal languages such as RDF and OWL have exclusively been designed for machines and are hard to write and understand for humans. There is an urgent need for an expressive high-level interface language to the semantic web that allows humans to write annotations in a familiar notation which is unambiguous and offers the same precision as a formal language.

A promising candidate for such a high-level interface language is the use of a controlled natural language. In general, a controlled natural language can be defined as a subset of a natural language with explicit constraints on grammar, lexicon, and style. These constraints usually have the form of writing rules and help to reduce both ambiguity and complexity of a full natural language [6, 7].

[10] www.ics.mq.edu.au/~rolfs/peng/

The probably most successful controlled natural language is ASD Simplified Technical English [8] that has been designed to improve the readability of aircraft maintenance documentation for non-native readers. However, readability of the language is only one important characteristic which needs to be combined with machine-processability to make the language useful in the context of question answering.

There are some relatively new controlled natural languages such as Attempto Controlled English [9], Common Logic Controlled English [10], and Processable English [11] that combine and balance readability and processability in such a way that makes it easier for humans and machines to work in cooperation. These highly expressive controlled natural languages are equivalent to large – in fact undecidable – fragments of first-order predicate logic, and have already been used as specification and knowledge representation languages in various application domains.

With some instruction, or supported by an intelligent authoring tool [12], even non-specialists can use these machine-oriented controlled natural languages to write annotations in a familiar notation without the need to formally encode their knowledge, and without a steep learning curve.

3 Controlled Natural Languages for Web Annotations

Traditionally, RDF-based languages and technologies have been promoted to semi-automatically generate annotations for web pages with machine-processable information. These annotations are usually not very expressive, and – once generated – are difficult to read and modify by humans.

For example, in the Friend of a Friend (FOAF) project[11], individual web pages are linked to machine-readable RDF documents which describe people, the links between them and the things these people create and are interested in. FOAF makes it easy to share and transfer information and to automatically extend, merge and reuse this information online. In the Annotea project[12], RDF-based annotation schemata are used for describing annotations as metadata and XPointer for locating the annotations in the annotated document. The annotations are stored locally, or in one or more annotation servers. When a document is browsed, a client such as Amaya[13] queries each of these servers, requesting the annotations related to that document.

RDF has been criticized as the formal underpinning for the semantic web [13, 14]. In particular the current RDF-based architecture for the semantic web has severe problems when more expressive rule languages are incorporated. An alternative approach is to use first-order logic as the semantic underpinning [13]. First-order logic is well-established and numerous state-of-the-art tools exist for processing first-order axiomatisations. There are various subsets of first-order logic that offer different tradeoffs with respect to expressivity, complexity and computability. Moreover, the direct mapping of subsets of first-order logic languages – for example between Horn logic and description logic – provides immediate semantic interoperability [15].

[11] www.foaf-project.org/
[12] www.w3.org/2001/Annotea/
[13] www.w3.org/Amaya/Amaya.html

The controlled natural language we promote for annotating web pages is first-order equivalent, but we have shown that subsets thereof can be translated automatically into OWL DL [16, 17]. However, exclusively relying on description logic would considerably reduce the expressive power of the controlled natural language [5].

4 How to Compose Web Annotations

To compose meaningful annotations for web pages let us have a brief look at what we can learn from the field of news writing and from existing guidelines for well-designed web pages.

4.1 Inverted Pyramid Style

Information in news reports is usually presented in an inverted pyramid style which begins with the conclusion, expressed as a single sentence. The subsequent paragraphs will then convey the most important and interesting information, leaving details and background information to further paragraphs in an order of diminishing importance. This format has the advantage that a reader can leave a report at any time without missing the most important facts. It also allows less important information to be more easily removed to fit a fixed size in a print medium.

Web usability experts recommend the inverted pyramid style for presenting textual information on web pages[14]. Putting the most important information into a lead section at the beginning of a web page better supports scanning of web pages by the human eye, and additionally minimizes the need for scrolling. Usability studies show that 79% of users scan a new web page and only 16% read it word-by-word[15].

4.2 The Lead Section

The lead section is the most important structural element of a well-designed web page and should convey the conclusion in a succinct form, usually in not more than 20-25 words. The importance of this lead section is also eminent in the manual of style of Wikipedia. This manual recommends that a Wikipedia article should be introduced by a lead section before the first headline and should summarize the most important information[16]:

> *The **lead section** should briefly summarize the most important points covered in an article in such a way that it could stand on its own as a concise version of the article. It is even more important here than for the rest of the article that the text be accessible, and consideration should be given to creating interest in reading the whole article.*

[14] www.useit.com/alertbox/9606.html
[15] www.useit.com/alertbox/9710a.html
[16] en.wikipedia.org/wiki/Wikipedia:Lead_section

It is obvious that such a lead section needs to be easy to read and write by humans and that machine-processability would add enormous benefits for various reasoning tasks such as question answering, consistency checking and information fusion. Representing this information in an RDF-based formal language is not very helpful, since this language is probably not expressive enough, and its syntax is a slap in the face of any human author (specialists or non-specialists alike). At first glance, writing a lead section looks like a challenging optimization problem, but a machine-oriented controlled natural language can bridge the gap here and the field of news writing can give us some valuable guidelines how to do this in a clever and informative way.

4.3 The Five W's (plus H)

The lead section not only encompasses specific constraints on sentence structure but also promotes a particular way in which the content is presented. The basic idea is that the lead section should attempt to answer all the fundamental questions about a peculiar event and this can be memorised as: *who* did *what when where* and *why*, and occasionally also *how*.

Let us illustrate how the most important information of the Landline web page[17] can be represented in a lead section using these five W's (plus H) as a guiding principle. For this purpose, we will use the following linguistic schema for sentences

subject + predicate + object + {modifiers}

where the subject answers the question about *who* is involved in a specific situation, the predicate states a particular event or state, the object answers a *what* question, and optional modifiers answer a *when*, *where*, *why* or *how* question.

Of course, users can freely compose lead sections following this schema. Alternatively, composing a lead section can be supported by an intelligent authoring tool that displays predictive information while the lead section is being written (cf. [4]).

Here is the step-wise construction of a possible lead section of the Landline web page with the help of a predictive authoring tool.

In our case, the transitive verb *cultivates* as predicate requires both a subject and an object.

[subject: who]
A couple ... **[predicate]**
A couple cultivates ... **[object: what]**
A couple cultivates capers ... **[modifiers: how | where | when | why]**

The *how*, *where*, *when*, and *why* can all be expressed as a sequence of modifiers terminated by a period.

A couple cultivates capers experimentally ... **[modifiers: where | when | why]**
A couple cultivates capers experimentally in South Australia ... **[modifiers …]**
…
A couple cultivates capers experimentally in South Australia since 1999 for economical benefit.

[17] www.abc.net.au/landline/content/2006/s1602940.htm

Please note that the information expressed in each constituent can directly be queried by questions in controlled natural language. For example:

Who cultivates capers? → *a couple*
Where does a couple cultivate capers? → *in South Australia*

but to answer our original question

Are capers grown in Australia?

we need additional linguistic background knowledge in the form of a lexical derivation rule (*if somebody cultivates something then somebody grows something*), the linguistic knowledge relating the active and passive forms of the transitive verb *grow*, and domain specific knowledge that specifies that *South Australia* is part of *Australia*.

As we will see in section 6.2, in general much more background information will be needed that has to be provided by external knowledge sources or explicitly by statements in controlled natural language.

5 How to Attach Web Annotations to a Web Page

Under the proposed model annotations function as lead sections of web pages. Therefore they need to be directly embedded into a web page by the author. Internally, the lead section is marked up as a paragraph and labeled with the help of an XHTML language attribute ("lang") together with an experimental language tag[18] ("x-cnl"). A search engine supporting this tag could then recognise that a paragraph is written in controlled natural language. In our case the result looks as follows:

* A couple cultivates capers experimentally in South Australia since 1999 for economical benefit . *

In this example the *lang* attribute's value *cnl* stands for an experimental language tag and indicates that the following snippet is written in controlled natural language. Figure 1 illustrates how the lead section can add value to a web page for both humans and machines.

Figure 1: Landline Article with Lead Section in Controlled Natural Language

[18] www.w3.org/TR/html4/struct/dirlang.html

The annotation being part of the web page, it will be indexed by search engines, and will also be available for any ranking that the search engine performs.

6 Deductions from Web Annotations

Question answering on the basis of annotations is done by a two-step process that in general needs linguistic and domain-specific background knowledge, and has to cope with the problems arising from inconsistent, incomplete, or differently conceptualised annotations.

Though the proposed annotations have a simple structure, background knowledge is complex, and in general involves quantification, negation, and disjunction. Thus question answering cannot be reduced to mere pattern matching, but requires first-order theorem proving.

6.1 A Two-Step Process to Answer Questions

Assuming that web pages are annotated by a lead section in controlled natural language, we suggest a two-step process to concisely answer questions in controlled natural language. This two-step process again reflects our decision to split the work between humans and machines according to their abilities, and thus complements our proposal to have web pages manually annotated.

In a first step, the question expressed in controlled natural language is automatically split into keywords that are then submitted together with the XHTML language attribute *lang="x-cnl"* to a ranking search engine that supports the language attribute. Since annotations are part of the respective web pages, the search engine will only return web pages containing the tag "x-cnl". Furthermore, the returned web pages are ranked with respect to the keywords of the question.

In the second step, we select the N top-ranked web pages and then try to automatically deduce the answer to our question separately from each of the N annotations. Deduction is done by converting the question Q and the annotations A of the selected web pages into their logical representations, Q' respectively A' and submitting $A' \cup \neg Q'$ to a theorem prover – possibly extending A' by formalised background knowledge (cf. section 6.2). Though we assume each annotation to be logically consistent, we cannot expect the set of annotations to be consistent. We also cannot expect that each of the N annotations will answer our question. If we get more than one answer, we present all answers to the user without trying to consolidate them, and leave their interpretation to the user. If available, we also provide information on the trustworthiness of the source. Note that page ranking already provides an implicit level of trustworthiness.

To support the outlined two-step process we propose a query tool that hides the computational details from the users and that contains a predictive editor to formulate questions in controlled natural language (cf. [4] for details).

6.2 Background Knowledge

No system can answer real world questions and make inferences without additional knowledge, i.e. knowledge that is not contained in the input. This applies also to our case: annotations written in controlled natural language require additional linguistic and domain specific background knowledge to serve as a complete knowledge base. However, an attractive feature of our approach is that much of the linguistic knowledge and all of the domain knowledge can be expressed in controlled natural language and is thus accessible for both man and machine.

Linguistic background knowledge is already needed in the first step of our approach when we split a question into keywords. If the annotation is

A couple cultivates capers experimentally in South Australia since 1999 for economical benefit.

and the question is

Are capers grown in Australia?

we cannot expect to get the question answered. However, we increase the probability to find adequate answers if we do a query expansion by allowing for synonyms of the content words of the question. For instance, linguistic resources like WordNet[19] provide for the verb *grow* the synonyms *cultivate*, *develop*, *increase*, *mature*, *originate*, *change* that we can add as alternatives to the keyword *grow* when we submit the keywords to the search engine. This will allow us to retrieve the above annotation as the basis for question answering.

More linguistic – and also domain-specific – background knowledge is required for the second, deductive, step of our approach. Assuming that the word *grown* of the question has been replaced by *cultivated* then we need linguistic knowledge to relate the active *somebody cultivates* and the passive *something is cultivated*. This relation can be expressed in controlled natural language, for example

If somebody cultivates something then something is cultivated by somebody.

Domain-specific knowledge needed for the second, deductive, step can conveniently be expressed in controlled natural language, for instance the geographic fact

South Australia is a part of Australia.

Now the question can be positively answered on the basis of the annotation. Other questions, like

Who grows capers in Australia?

What grows in South Australia?

Where do capers grow?

Since when are capers cultivated in South Australia?

Why are capers cultivated in South Australia?

---

[19] wordnet.princeton.edu/perl/webwn

can similarly be answered provided the required background knowledge is made available.

Where does the background knowledge come from, where is it stored, and how is it applied?

Linguistic knowledge can be extracted from linguistic resources such as WordNet, expressed in controlled natural language, and directly be converted to the logical representation of the controlled natural language. Domain knowledge can be composed by the user in controlled natural language, or (semi-) automatically extracted from existing ontologies or knowledge bases such as Cyc[20], and then converted into controlled natural language.

Since writers of annotations cannot anticipate the variety of questions asked, it seems natural to associate the background knowledge with the question, concretely to incorporate it in a suitable representation into the query tool. Alternatively, it may turn out to be more convenient to split the background knowledge into a user-independent part that is associated with the annotation and stored on some server, and a user-specific part that is associated with the question.

6.3 Missing and Inconsistent Answers

We cannot expect that each retrieved annotation will answer our question since annotations can violate the principles of good writing presented in section 4. One should rather assume that some annotations are incomplete, conceptualised differently to the question, or expressed in a way that no satisfying answer can be deduced.

Another issue is inconsistency. Though each annotation is expected to be consistent, the set of retrieved annotations is not necessarily consistent, and thus answers to our question can be inconsistent. Some researchers [18] have suggested to replace standard first-order logic by paraconsistent logic. Though this might be applicable in some cases, we believe that the enormous range of information available on the web simply does not allow for a coherent solution[21]. Instead, we leave it to the user to interpret the validity and the trustworthiness of the answers.

7 Other Uses of Web Annotations

Since web annotations in computer-processable controlled natural language have a logical foundation they can be used for many other purposes involving deduction, for instance comparing annotations of different web pages or checking annotations for compliance with respect to ontologies and knowledge bases.

If required by an application, annotations written in controlled natural language can be exported in RuleML[22], or in non-XML notations.

---

[20] www.cyc.com/

[21] www.w3.org/DesignIssues/Inconsistent.html

[22] www.ruleml.org

The annotations can also be exported as news feeds, for instance in our capers example, to inform Australian exporters of fruit and vegetable of an opportunity to expand their business with a new product.

Last, but not least an annotation in controlled natural language is a human-readable summary of the respective web page, and fulfills similar functions to the lead section of Wikipedia articles.

8 Conclusions

We propose to manually augment web pages with annotations in a controlled natural language. Our approach offers the following advantages:

- annotations in computer-processable controlled languages permit formal reasoning, specifically question answering by deduction,
- question answering from annotations in controlled natural languages can easily be supported by the necessary linguistic and domain-specific background knowledge,
- annotations in controlled natural languages are readable by anybody, and thus can also serve as a summary of the respective web page,
- annotations in controlled natural languages can be written – preferably with the support of an authoring tool – according to standard guidelines of good summary writing, for instance Wikipedia's guidelines for lead sections.

Arguably, annotations written in controlled natural language can bridge the gap between informal and formal notations and leverage true collaboration between humans and machines. However, our solution introduces a "chicken and egg" problem: a critical mass of web annotations will be necessary that people perceive the value of these annotations and start annotating web pages themselves. Only the future will show whether this – basically non-technical – problem can be solved.

Acknowledgements

The reported work was performed while N. E. Fuchs was visiting the Centre for Language Technology at Macquarie University in Sydney, Australia. The authors are grateful to Macquarie University's Visiting Scholars Scheme 2006 that provided the major part of the necessary funding. The authors would also like to thank their colleagues Mark Dras, Kaarel Kaljurand, Tobias Kuhn, Diego Mollá, Luiz Augusto Pizzato and three anonymous ESWC'07 reviewers for helpful hints and comments.

References

1. Hovy, E.: Text Summarization, in: R. Mitkov (ed.), The Oxford Handbook of Computational Linguistics, Oxford University Press (2003)

2. Katz, B., Lin, J.: Annotating the Semantic Web Using Natural Language, in: Proceedings of the 2nd Workshop on NLP and XML at COLING 2002, Taipei, Taiwan (2002)
3. Attempto Project; www.ifi.unizh.ch/attempto and attempto.ifi.unizh.ch
4. Schwitter, R., Tilbrook, M.: Annotating Websites with Machine-processable Information in Controlled Natural Language, in: M. A. Orgun and T. Meyer (eds.), Advances in Ontologies 2006, Proc. of AOW 2006, Hobart, Australia, Australian Computer Society, Conferences in Research and Practice in Information Technology, Vol. 72, (2006), 75-84
5. Kuhn, T., Royer, L., Fuchs, N. E., Schroeder, M.: Improving Text Mining with Controlled Natural Language: A Case Study for Protein Interactions, in: U. Leser, B. Eckman, and F. Naumann, editors, Proceedings of the 3rd International Workshop on Data Integration in the Life Sciences 2006 (DILS'06), Lecture Notes in Bioinformatics, Springer (2006)
6. Huijsen, W. O.: Controlled Language - An Introduction, in: Proceedings of CLAW 1998, Pittsburgh (1998) 1-15
7. O'Brien, S.: Controlling Controlled English – An Analysis of Several Controlled Language Rule Sets, in: Proceedings of EAMT-CLAW 03, Controlled Language Translation, Dublin City University (2003), 105-114
8. Simplified Technical English. Specification ASD-STE100, A Guide or the Preparation of Aircraft Maintenance Documentation in the International Aerospace Maintenance Language, Issue 3, January (2005)
9. Fuchs, N. E., Kaljurand, K., Schneider, G.: Attempto Controlled English Meets the Challenges of Knowledge Representation, Reasoning, Interoperability and User Interfaces, in: FLAIRS'2006, 2006.
10. Sowa, J. F.: Common Logic Controlled English Draft, 24 February 2004, available at: http://www.jfsowa.com/clce/specs.htm, (2004)
11. Schwitter, R.: English as a Formal Specification Language, in: Proceedings of the Thirteenth International Workshop on Database and Expert Systems Applications (DEXA 2002), W04: Third International Workshop on Natural Language and Information Systems - NLIS, 2-6 September 2002, Aix-en-Provence, France, (2002) 228-232
12. Schwitter, R., Ljungberg, A., Hood, D.: ECOLE – A Look-ahead Editor for a Controlled Language, in: Controlled Translation, Proceedings of EAMT-CLAW03, Joint Conference combining the 8th International Workshop of the European Association for Machine Translation and the 4th Controlled Language Application Workshop, May 15-17, Dublin City University, Ireland, (2003) 141-150
13. Horrocks, I., Patel-Schneider, P. F.: Three Theses of Representation in the Semantic Web, in: Proc. of WWW 2003, (2003) 39-47
14. Patel-Schneider, P. F.: A Revised Architecture for Semantic Web Reasoning, in: Proceedings of Third Workshop on Principles and Practice of Semantic Web Reasoning (PPSWR'05), Dagstuhl, Germany (11-16th September 2005), Organization: REWERSE, LNCS 3703, (2005) 32-36
15. Grosof, B. N., Horrocks, I., Volz, R., Decker, S.: Description Logic Programs: Combining Logic Programs with Description Logic, in: Proceedings of the 12th International Conference on the World Wide Web (2003) 48-57
16. Schwitter, R., Tilbrook, M.: Let's Talk in Description Logic via Controlled Natural Language, in: Proc. of the Third International Workshop on Logic and Engineering of Natural Language Semantics (LENLS2006) in Conjunction with the 20th Annual Conference of the Japanese Society for Artificial Intelligence, Tokyo, Japan, June 5-6, (2006) 193-207
17. Kaljurand, K., Fuchs, N. E.: Bidirectional mapping between OWL DL and Attempto Controlled English, in: Fourth Workshop on Principles and Practice of Semantic Web Reasoning, Budva, Montenegro, (2006)
18. Schaffert, S., Bry, F., Besnard, P., Decker, H., Decker, S., Enguix, C., Herzig A.: Position Paper: Paraconsistent Reasoning for the Semantic Web. Technical Report PMS-FB-2005-42. Institut für Informatik der Ludwig-Maximilians-Universität München, (2005)