Question Answering using Sentence Parsing and Semantic Network Matching

Sven Hartrumpf
Intelligent Information and Communication Systems
University of Hagen (FernUniversität in Hagen)
58084 Hagen, Germany
Sven.Hartrumpf@fernuni-hagen.de

Abstract

The paper describes a question answering system for German called InSicht. All documents in the system are analyzed by a syntactico-semantic parser in order to represent each document sentence by a semantic network (in the MultiNet formalism) or a partial semantic network (if only a parse in chunk mode succeeds). A question sent to InSicht is parsed yielding its semantic network representation and its sentence type. The semantic network is expanded to equivalent or similar semantic networks (query expansion stage) by applying equivalence rules, implicational rules (in backward chaining), and concept variations based on semantic relations in computer lexicons and other knowledge sources. During the search stage, every semantic network generated for the question is matched with semantic networks for document sentences. For efficiency, a concept index server is applied to reduce the number of matches tried. If a match succeeds, an answer string is generated from the matching semantic network in the supporting document by answer generation rules. Among competing answers, one answer is chosen by combining a preference for longer answers and a preference for more frequent answers.

The system is evaluated on the QA@CLEF 2004 test set. A hierarchy of problem classes is proposed and a sample of suboptimally answered questions is annotated with problem classes from this hierarchy. Finally, some conclusions are drawn, main problems are identified, and directions for future work as suggested by these problems are indicated.

1 Introduction

This paper presents the InSicht question answering (QA) system currently implemented for German. Its key characteristics are:

- Deep syntactico-semantic analysis with a parser for questions and documents.
- Independence from other document collections. No other documents, e.g. from the World Wide Web (WWW), are accessed, which helps to avoid unsupported answers. QA that works on WWW documents is sometimes called web-based QA in contrast to textual QA, see for example (Neumann and Xu, 2003).
- Generation of the answer from the semantic representation of the documents that support the answer. Answers are not directly extracted from the documents.

There are few QA systems for German. The system described by Neumann and Xu (2003) differs mainly in its general approach: it relies on shallow, but robust methods, while InSicht builds on deep sentence parsing. In this respect, InSicht resembles the (English) QA system presented by Harabagiu et al.

Table 1: Statistics from Document Preprocessing

subcorpus	articles without duplicates	sentences	words	average sentence length	duplicate articles	
					identical bytes	identical words
FR	122541	2472353	45332424	18.3	22	17152
SDA	140214	1930126	35119427	18.2	333	568
SP	13826	495414	9591113	19.4	0	153
all	276581	4897893	90042964	18.4	355	17873

Table 2: Statistics from Document Parsing

subcorpus	parse results	full parse (%)	chunk parse (%)	no parse (%)
FR	2469689	44.3	21.7	34.0
SDA	1930111	55.8	19.0	25.2
SP	485079	42.7	19.3	38.0
all	4884879	48.7	20.4	30.9

(2001). In contrast to InSicht, this system applies a theorem prover and a large knowledge base to validate candidate answers.

The following Sections 2–7 present InSicht's main components. In Section 8, the system is evaluated on the QA@CLEF 2004 questions. Furthermore, problem classes are defined and attributed to individual questions. The final Section 9 draws conclusions and describes perspectives for future work.

2 Document Processing

The corpus files distributed for QA@CLEF 2004 are split in a first preprocessing step into article files using an SGML parser (*nsgmls*) and a shell script. Then, each article is tokenized, split into sentences, and stored in a separate SGML file conforming to the Corpus Encoding Standard (Ide et al., 1996). The tags for words (w) and sentences (s) are annotated, but it is not attempted to determine paragraph borders because of the mixed encoding quality of the original files.

Duplicate articles are eliminated. Especially in the subcorpus of the *Frankfurter Rundschau* (FR), the percentage of articles with one or more articles showing the same word sequence (ignoring white space and control characters) is astonishingly high (12.3%); for details, see Table 1. Duplicate elimination has several advantages: selecting among candidate answers (see Section 7) becomes more accurate, and debugging during further development of the QA system becomes clearer and faster.

After document preprocessing, the WOCADI (WOrd ClAss based DIsambiguating) parser (Helbig and Hartrumpf, 1997; Hartrumpf, 2003) parses article by article. For each sentence in an article, the syntactico-semantic (deep) parser tries to generate a correct representation as a semantic network of the MultiNet formalism (Helbig, 2001; Helbig and Gnörlich, 2002). To speed up this parsing step, which takes 5–6 months for the whole document collection, parser instances were run in parallel in a Linux cluster of 4–6 standard PCs. Each PC was equipped with one AMD Athlon XP 2000+ or similar CPU. The documents must be parsed only once; questions never require any reprocessing of documents. The subcorpus from the *Schweizerische Depeschenagentur* (SDA) is parsed with a special WOCADI option that triggers the reconstruction of *ß* from *ss*, because WOCADI is not primarily developed for Swiss German.

The parser produced complete semantic networks for 48.7% of all sentences and only partial semantic networks (corresponding to a WOCADI parse in chunk mode) for 20.4%. The percentages for the three subcorpora differ considerably (see Table 2). This reflects the differences in encoding quality of the original SGML files and in language complexity. For example, the SDA subcorpus is parsed best because newswire

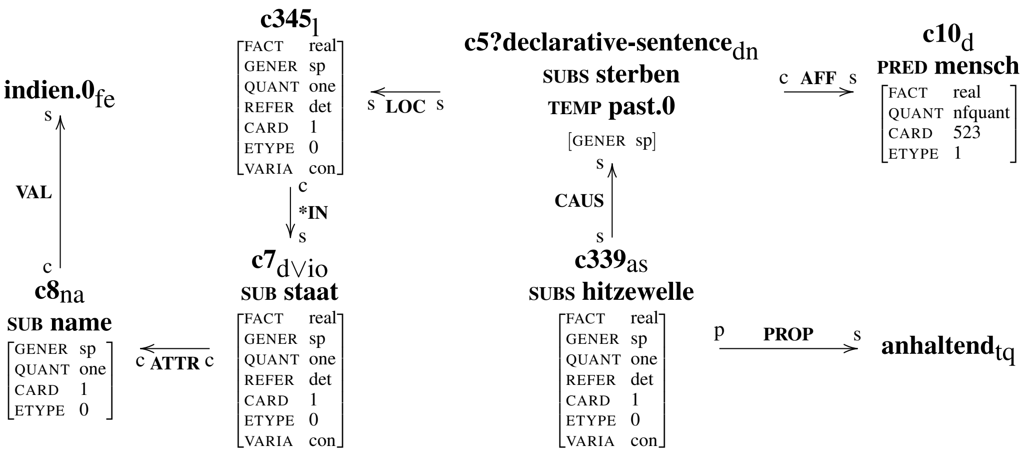

Figure 1: Graphical form of the MultiNet generated by the WOCADI parser for (simplified) document sentence SDA.950618.0048.377: *In Indien starben [. . .] 523 Menschen infolge der [. . .] anhaltenden Hitzewelle.* (*'523 people died in India due to the continuing heat wave.'*)

sentences are typically simpler in structure than newspaper sentences and the original SGML files show fewer encoding errors than the ones for FR and *Der Spiegel* (SP). The numbers in the second column of Table 2 are slightly smaller than the corresponding numbers in the third column of Table 1 because for efficiency reasons the analysis of a text will be stopped if a certain maximal number of semantic network nodes is produced during parsing the sentences of the text. A semantic network for a simplified document sentence is shown in Figure 1. Edges labeled with the relations PRED, SUB, SUBS, and TEMP are *folded* (printed below the name of the start node) if the network topology allows this, e.g. SUB *name* below node name *c8*. As a last step, semantic networks are simplified and normalized as described in Section 5.

3 Question Processing

A question posed by a user (online) or drawn from a test collection (offline; like the 200 questions for QA@CLEF 2004), is parsed by the same parser that produced the semantic networks for the documents. The parser relies only on the question string and ignores for example the annotated question type (in 2004, this could be F for factoid or D for definition). The parsing result is a semantic network from the MultiNet formalism plus additional information relevant for the QA system: the (question) focus (marked in graphical semantic networks by a question mark) and the sentence type (written directly behind the focus mark in graphical semantic networks). The MultiNet for question 164 from QA@CLEF 2004 is shown in graphical form in Figure 2.

For the questions of QA@CLEF 2004, the sentence type is determined with 100% correctness. Only 3 of 10 values for the sentence type attribute occur for these questions, namely *wh-question*, *count-question*, and *definition-question*.

4 Query Expansion

For a semantic network representing a question, equivalent networks are generated by applying equivalence rules (or paraphrase rules) for MultiNet. In contrast to such semantic rules, some QA systems (e.g. the one described by Echihabi et al. (2003)) use reformulation rules working on strings. Surface string operations are the more problematic the freer the word order is. As the word order in German is less constrained than in English, such operations may be more problematic and less effective in German.

For maintenance reasons, many rules are abstracted by so-called rule schemas. For example, three rule schemas connect a state with its inhabitant and with the state adjective, e.g. *Spanien* (*'Spain'*), *Spanier*

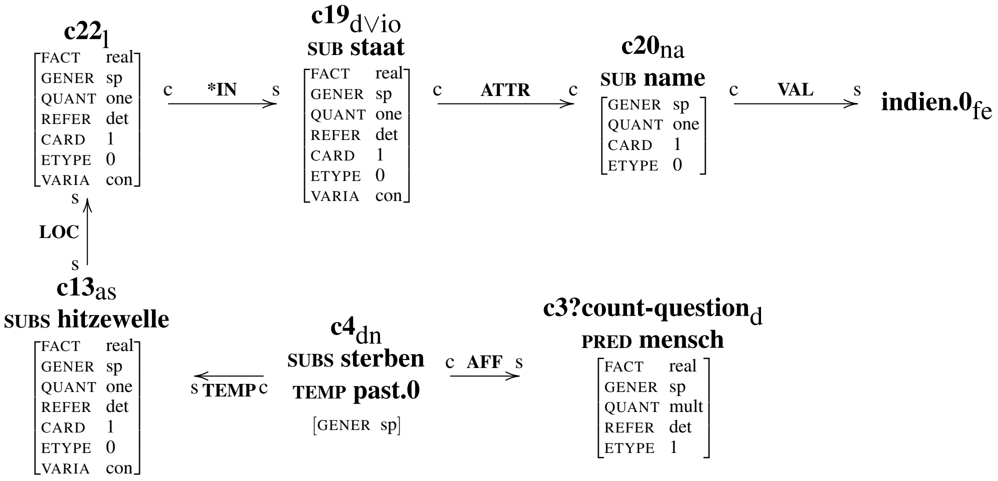

Figure 2: Graphical form of the MultiNet generated by the WOCADI parser for question 164: *Wie viele Menschen starben während der Hitzewelle in Indien?* (*'How many people died during the heat wave in India?'*)

```
((rule
   (
     (subs ?n1 ”ermorden.1.1”)
     (aff ?n1 ?n2)
   →
     (subs ?n3 ”sterben.1.1”)
     (aff ?n3 ?n2)))
   (ktype categ)
   (name ”ermorden.1.1_entailment”))
```

Figure 3: Entailment rule for *ermorden* (*'kill'*) and *sterben* (*'die'*)

(*'Spaniard'*), and *spanisch* (*'spanish'*). In addition, the female and male nouns for the inhabitant are connected in the computer lexicon HaGenLex (**Ha**gen **Ge**rma**n** **Lex**icon; see (Hartrumpf et al., 2003)) by a certain MultiNet relation. Similar rule schemas exist for regions.

In addition to equivalence rules, implicational rules for lexemes are used in backward chaining, e.g. the logical entailment between *ermorden.1.1*[1] (*'kill'*) and *sterben.1.1* (*'die'*); see Figure 3. All rules are applied to find answers that are not explicitly contained in a document but only implied by it. Figure 4 shows one of the 109 semantic networks[2] generated for question 164 from Figure 2 during query expansion. This semantic network was derived by applying two default rules for MultiNet relations (in backward chaining). The first rule transfers the LOC edge from the abstract situation (subordinated to *hitzewelle*) to the situation node (subordinated to *sterben*). The second rule (shown in Figure 5) expresses as a default that a causal relation (CAUS) implies (under certain conditions, indicated by a sort constraint) a temporal overlap (TEMP). Reconsidering the semantic network in Figure 1 for a document sentence, the similarity to the question variant from Figure 4 becomes obvious. This similarity allows a match and the generation of a correct answer (namely just a number: *523*) in the remaining stages of the InSicht system.

Besides rules, InSicht applies other means to generate equivalent (or similar) semantic networks: Each concept in a semantic network can be replaced by concepts that are synonyms, hyponyms, etc. Such concept variations are based on lexico-semantic relations in HaGenLex. As HaGenLex contains a mapping from lexemes to GermaNet concept IDs (Osswald, 2004), synonymy and subordination relations from

[1] A lemma followed by a numerical homograph identifier and a numerical polyseme identifier forms a so-called concept identifier (or concept ID) in HaGenLex. In this paper, the numerical suffix of concept IDs is often omitted to improve readability.

[2] This number does not include any concept variations.

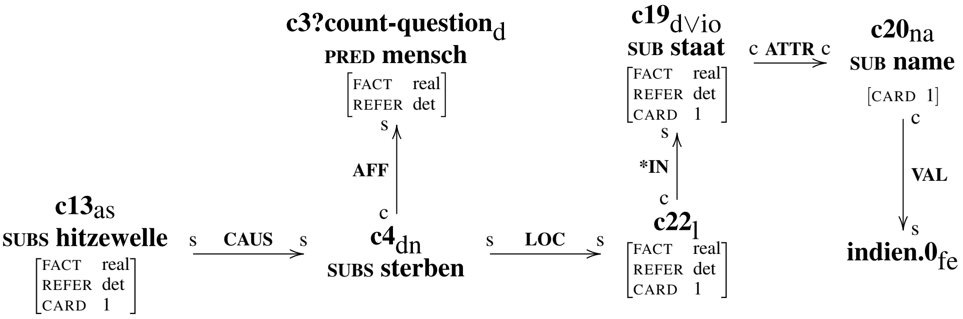

Figure 4: One result from query expansion for question 164 from Figure 2

```
((rule
  (
    (caus ?n1 ?n2)
    (sort ?n1 as)
  →
    (temp ?n2 ?n1)))
  (ktype proto)
  (name ”caus_temp”))
```

Figure 5: Example rule (applied in backward chaining during query expansion)

GermaNet were used in a separate experiment in addition to the lexico-semantic relations from HaGenLex. For the questions from the test set, this extension led to no changes in the answers given. On average, query expansion using rules led to 6.5 additional semantic networks for a question from QA@CLEF 2004. If one counts the combination with concept variations, around 215 semantic networks are used per question.

The use of inference rules during a query expansion stage is just a pragmatic decision. In an ideal system without memory constraints, rules could come into play later: the semantic representation of all documents would be loaded as a huge knowledge base (where one had to cope with inconsistencies) and rules would be used by a theorem prover to test whether the question (or some derived form) can be deduced from the knowledge base. The main reasons to avoid such a system are the huge amount of facts coming from the document collection and the problem of inconsistencies.

5 Search

To search for an answer by semantic network matching, the semantic network for the question is split in two parts: the *queried network* (roughly corresponding to the representation of the phrase headed by the interrogative pronoun or determiner) and the *match network* (the semantic network without the queried network). The matcher calls a concept ID index server for all concepts in the match network to speed up the search. Efficient matching of the match network is achieved by simplifying networks as described in the next paragraph (for question networks and document networks in the same way) so that a subset test with a large set of query expansions (generated as described in Section 4) becomes feasible. Average answer time is several seconds on a standard PC. A variant of this matching approach has been tried in the monolingual GIRT task (see one of the five runs reported by Leveling and Hartrumpf (2004)), currently with retrieval results that are not sufficient yet.

Semantic networks are simplified and normalized to achieve acceptable answer times. The following simplifications are applied: First, inner nodes of a semantic network that correspond to instances (for example *c4* and all nodes named *cN* in Figure 4) are combined (collapsed) with their concept nodes (typically connected by a SUB, SUBS, PRED, or PREDS relation) to allow a canonical order of network edges. Sometimes this operation necessitates additional query expansions. (These semantic networks are basically

(*in ”c1*in” ”c1staat.1.1”)
(aff ”c1sterben.1.1” ”c1mensch.1.1”)
(attr ”c1staat.1.1” ”c1name.1.1”)
(caus ”c1hitzewelle.1.1” ”c1sterben.1.1”)
(loc ”c1sterben.1.1” ”c1*in”)
(prop ”c1hitzewelle.1.1” ”anhaltend.1.1”)
(temp ”c1sterben.1.1” ”past.0”)
(val ”c1name.1.1” ”indien.0”)

Figure 6: Simplified and normalized semantic network for the MultiNet of Figure 1. For better readability, features of nodes are omitted.

variations of possible instance node names.) Second, semantic details from some layers in MultiNet are omitted, e.g. the features ETYPE and VARIA of nodes and the knowledge types of edges (Helbig, 2001). After such simplifications, a lexicographically sorted list of MultiNet edges can be seen as a canonical form, which allows efficient matching. The simplified and normalized semantic network corresponding to the MultiNet in Figure 1 is shown in Figure 6.

6 Answer Generation

Generation rules take the (simplified) semantic network of the question (the queried network part), the sentence type of the question, and the matching semantic network from the document as input and generate a German phrase (typically a noun phrase) as a candidate answer. The generation rules are kept simple because the integration of a separately developed generation module is planned so that InSicht's current answer generation is only a temporary solution. Despite the limitations of the current answer generation, it proved advantageous to work with small coverage rules because they filter what a good answer can be. For example, no rule generates a pronoun; so uninformative pronouns cannot occur in the answer. If the expected answer becomes more complex, this filtering advantage will shrink.

An answer extraction strategy working on surface strings in documents is avoided because in languages showing more inflectional variation than say English, simple extraction from surface strings can lead to an answer that describes the correct entity, but in an incorrect syntactic case. Such an answer should be judged as inexact or even wrong.

7 Answer Selection

The preceding steps typically result in many pairs of generated answer string and supporting document ID[3] for a given question. To select the best answer, a preference for longer answers and a preference for more frequent answers are combined. Answer length is measured by the number of characters and words. In case of several supporting documents, the document whose ID comes alphabetically first is picked. This strategy is simple and open to improvements but works surprisingly well so far.

To automatically detect cases where question processing (or some later stage) made a mistake that led to a very general matching and finally to far too many competing candidate answers, a maximum for different answer strings is defined (depending on question type). If it is exceeded, the system retreats to an empty answer (*NIL*) with a reduced confidence score.

8 Evaluation on the QA@CLEF 2004 Test Set

By annotating each question leading to a suboptimal answer[4] with a problem class, the system components which need improvements most urgently can be identified. After fixing a general programming error, InSicht achieved 80 correct answers in an unofficial re-run (official run: 67) and 7 inexact answers for 197[5]

[3] As each answer is generated from a semantic network corresponding to one document sentence, the system also knows the ID (a byte offset) of the supporting sentence in this document.

[4] A suboptimal answer is one not marked as correct (R) by the assessors.

[5] Three questions have been excluded from the evaluation by the co-ordinators of the German QA task after my report of spelling errors; see problem class *q.ungrammatical* in Table 3.

Table 3: Hierarchy of problem classes and problem class frequencies (percentages sum to 100.2 due to rounding)

name	description	% for QA@CLEF 2004
problem		
q.error	error on question side	
q.parse_error	question parse is not complete and correct	
q.no_parse	parse fails	0.0
q.chunk_parse	only chunk parse result	0.0
q.incorrect_parse	parser generates full parse result, but it contains an error	13.3
q.ungrammatical	question is ungrammatical	2.7
d.error	error on document side	
d.parse_error	document sentence parse is not complete and correct	
d.no_parse	parse fails	33.2
d.chunk_parse	only chunk parse result	2.0
d.incorrect_parse	parser generates full parse result, but it contains an error	7.8
d.ungrammatical	document sentence is ungrammatical	2.0
q-d.error	error in connecting question and document	
q-d.failed_generation	no answer string can be generated for a found answer	2.0
q-d.matching_error	match between semantic networks is incorrect	5.9
q-d.missing_cotext	answer is spread across several sentences	5.9
q-d.missing_inferences	inferential knowledge is missing	25.4

scored questions, which leaves 110 questions (where the system gave an incorrect empty answer) to be annotated. The hierarchy of problem classes shown in Table 3 was defined before annotation started. As this annotation is time-consuming, only a sample of 43 questions has been classified so far. Therefore only the percentages for problem class *q.error* and its subclasses are exact, the other percentages are estimates from the sample.

For a question, a problem subclass (preferably a most specific subclass) for *q.error*, *d.error*, and *q-d.error* could be annotated in theory. But the chosen approach is more pragmatic: If a problem is found in an early processing stage, one should stop looking at later stages, no matter whether one could investigate them despite the early problem, one could speculate about them, or just guess.

Seeing the high numbers for the problem class *d.parse_error* and its subclasses one could suspect that a parse error for the relevant document sentence[6] excludes a correct answer in general. Fortunately this is not the case. For example, question 081 was answered correctly by using the semantic network for sentence SDA.940610.0174.84 although the semantic network contained some errors; but the semantic network part relevant for the answer was correct.

9 Conclusions and Perspectives

InSicht achieves high precision: non-empty answers (i.e. not *NIL* answers) are rarely wrong (for the QA@CLEF 2004 questions only one; in the unofficial re-run not a single one). Furthermore, the deep level of representation based on semantic networks opens the way for intelligent processes like paraphrasing on the semantic level and inferences.

The experience with the current system showed the following five problems; after naming the problem, a solution for future work is suggested:

1. Missing inferential knowledge: encode and semi-automatically acquire entailments etc.

[6] If several document sentences are relevant, InSicht (as other QA systems) can often profit from this redundancy.

2. Limited parser coverage: extend the lexicons and improve the robustness and grammatical knowledge of the parser.

3. Ignoring partial semantic networks (produced by the parser in chunk mode): devise methods to utilize partial semantic networks for finding answers.

4. Answers spread across several sentences are not found: apply the text mode of the parser (involving intersentential coreference resolution, see (Hartrumpf, 2001)).

5. Long processing for documents: optimize the parser and develop on-demand processing strategies.

References

Echihabi, Abdessamad; Douglas W. Oard; Daniel Marcu; and Ulf Hermjakob (2003). Cross-language question answering at the USC Information Sciences Institute. In *Results of the CLEF 2003 Cross-Language System Evaluation Campaign, Working Notes for the CLEF 2003 Workshop* (edited by Peters, Carol), pp. 331–337. Trondheim, Norway.

Harabagiu, Sanda; Dan Moldovan; Marius Paşca; Rada Mihalcea; Mihai Surdeanu; Răzvan Bunescu; Roxana Gîrju; Vasile Rus; and Paul Morărescu (2001). The role of lexico-semantic feedback in open-domain textual question-answering. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL-2001)*, pp. 274–281. Toulouse, France.

Hartrumpf, Sven (2001). Coreference resolution with syntactico-semantic rules and corpus statistics. In *Proceedings of the Fifth Computational Natural Language Learning Workshop (CoNLL-2001)*, pp. 137–144. Toulouse, France. URL `http://www.aclweb.org/anthology/W01-0717`.

Hartrumpf, Sven (2003). *Hybrid Disambiguation in Natural Language Analysis*. Osnabrück, Germany: Der Andere Verlag.

Hartrumpf, Sven; Hermann Helbig; and Rainer Osswald (2003). The semantically based computer lexicon HaGenLex – Structure and technological environment. *Traitement automatique des langues*, 44(2):81–105.

Helbig, Hermann (2001). *Die semantische Struktur natürlicher Sprache: Wissensrepräsentation mit MultiNet*. Berlin: Springer.

Helbig, Hermann and Carsten Gnörlich (2002). Multilayered extended semantic networks as a language for meaning representation in NLP systems. In *Computational Linguistics and Intelligent Text Processing (CICLing 2002)* (edited by Gelbukh, Alexander), volume 2276 of *LNCS*, pp. 69–85. Berlin: Springer.

Helbig, Hermann and Sven Hartrumpf (1997). Word class functions for syntactic-semantic analysis. In *Proceedings of the 2nd International Conference on Recent Advances in Natural Language Processing (RANLP'97)*, pp. 312–317. Tzigov Chark, Bulgaria.

Ide, Nancy; Greg Priest-Dorman; and Jean Véronis (1996). *Corpus Encoding Standard*. URL `http://www.cs.vassar.edu/CES/`.

Leveling, Johannes and Sven Hartrumpf (2004). University of Hagen at CLEF 2004: Indexing and translating concepts for the GIRT task. In *Results of the CLEF 2004 Cross-Language System Evaluation Campaign, Working Notes for the CLEF 2004 Workshop* (edited by Peters, Carol). Bath, England.

Neumann, Günter and Feiyu Xu (2003). Mining answers in German web pages. In *Proceedings of the International Conference on Web Intelligence (WI-2003)*. Halifax, Canada.

Osswald, Rainer (2004). Die Verwendung von GermaNet zur Pflege und Erweiterung des Computerlexikons HaGenLex. *LDV Forum*, 19(1):43–51.