University of Hagen at CLEF 2003: Natural language access to the GIRT4 data

Johannes Leveling
Applied Computer Science VII
Intelligent Information and Communication Systems
FernUniversität Hagen
58084 Hagen, GERMANY
johannes.leveling@fernuni-hagen.de

Abstract

A natural language interface to databases allows a natural formulation of information needs, requiring little or no previous knowledge about database intrinsics or formal retrieval languages. Aiming at a full understanding of unrestricted natural language queries, the transformation into database queries and search failures caused by a vocabulary mismatch between query terms and database index terms become major problems to solve. This paper investigates methods of constructing query variants and their use in automated retrieval strategies. Performance results for an experimental setup with a GIRT4 database are presented.

1 Introduction

Query processing in a natural language interface (NLI) for databases faces two major problems:

1. The first problem is how to transform a user's natural language query into a database query.
2. The second problem is how to treat vocabulary problems between terms in a user query and terms in a database index. Unrestricted natural language input can contain vague or ambiguous utterances and over- or underspecifications of a user's information need. This causes – from a user's point of view – search failures.

This paper investigates the performance of automated retrieval strategies. The retrieval strategies create query variants from search terms linguistically related to the terms used in the natural language description of an information need, rank them according to semantic similarity and successively retrieve documents up to a fixed result set size. To find search term variants, orthographic and morphological, lexical and syntactical variants are looked up with NLP tools in lexicon resources.

Before presenting the approach and setup for retrieval experiments with GIRT (German Indexing and Retrieval Test database, [8]) in detail, a short overview of the underlying infrastructure of our solution to the first problem is given.

2 The NLI-Z39.50

The NLI-Z39.50 ([10]) is a natural language interface that supports searching in library databases with the standardized Internet protocol Z39.50 [11] for bibliographic data. The NLI-Z39.50 was developed as part of a project funded by the DFG (Deutsche Forschungsgemeinschaft). The Z39.50 (ANSI/NISO Z39.50) protocol is the de-facto standard in information retrieval targeting library databases, government or geospatial information resources and the basis of services for locating collections in digital libraries and for explaining database properties. Figure 1 gives an overview of the data and communication flow in the NLI-Z39.50. The transformation of a natural language query into a database query is divided into subsequent processing stages handled by five separate modules:

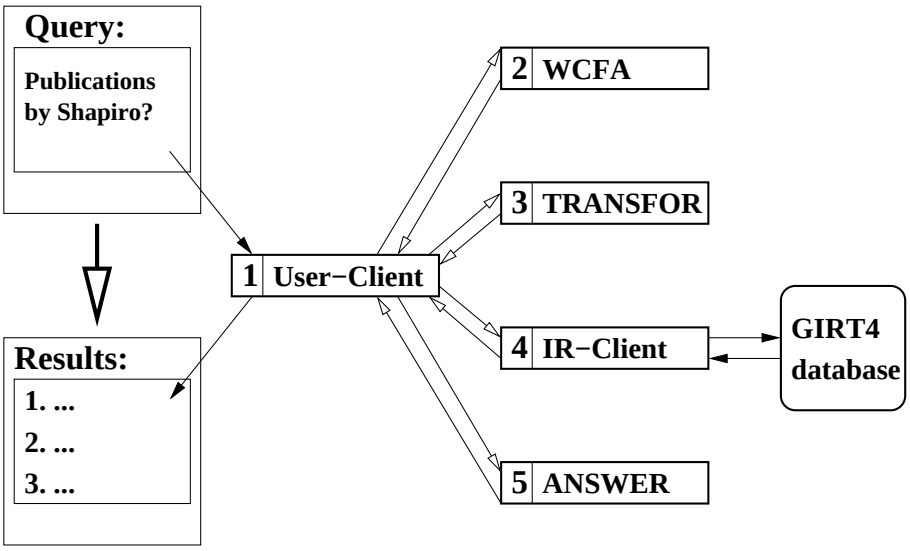

Figure 1: The NLI architecture for the GIRT4 experiments

1 **User-Client:** The User-Client handles input and output and coordinates the processing steps and the communication between the system modules. It is accessible via a standard Internet browser, accepts natural language queries as input and displays the results in a readable form. For example, both the title and description of GIRT topic 81 (*"Ausbildungsabbruch"/ "Vocational training dropout"* and *"Finde Dokumente, die über den Abbruch von Ausbildungsverhältnissen in Betrieben berichten."/ "Find documents on the dropout from vocational training in companies."*) is an acceptable and adequate input for the User-Client.

2 **WCFA:** By means of a Word Class Functional Analysis ([7], [4]), the natural language query is transformed into a well-documented knowledge and meaning representation, Multilayered Extended Semantic Networks (abbreviated as MultiNet) [6]. The natural language analysis is supported by HaGenLex [5], a domain-independent set of computer lexicons linked to and supplemented by external sources of lexical and morphological information, in particular CELEX [2] and GermaNet [9]. HaGenLex includes:

- A lexicon with full morpho-syntactic and semantic information of about 20,000 lexemes.
- A shallow lexicon containing word forms with morpho-syntactic information only. This lexicon contains about 50,000 entries.
- Several lexicons with more than 200,000 proper nouns (including product names, company names, country and city names, etc.)

Figure 2 shows the top layer of the multilayered semantic network representation of the description of GIRT topic 81. The core MultiNet consists of concepts (nodes) and semantic relations and functions between them (edges). The relations and functions in the example network are presented with a short description in figure 3. The MultiNet Paradigm defines a fixed set of 93 semantic relations (plus a set of functions) to describe the meaning connections between concepts, including synonymy (SYNO), subordination, i.e. hyponomy and hypernymy (SUB), meronymy and holonymy (PARS), antonymy (ANTO) and relations for change of sorts between lexemes. For example the relation CHEA indicates a change from an event (verb) into an abstract object (noun).

The WCFA provides powerful disambiguation modules, which use semantic and syntactic information to disambiguate lexemes. MultiNet differentiates between homographs, polysemes, and meaning molecules (a regular polyseme with different meaning facets, for example *"Schule"/ "school"*).

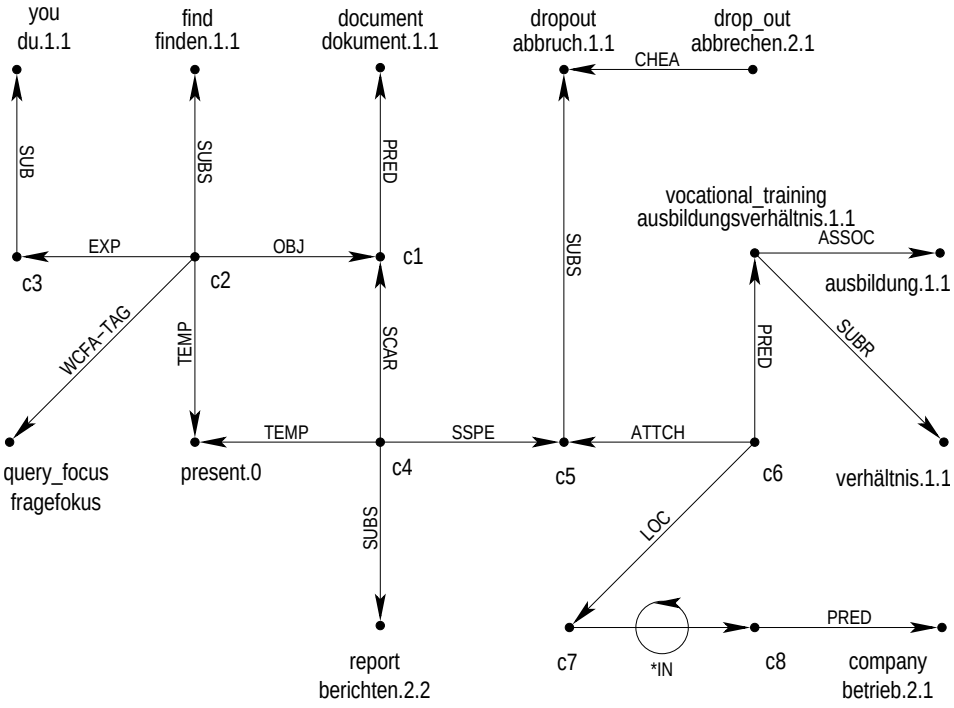

Figure 2: The core MultiNet representation for the query *"Finde Dokumente, die über den Abbruch von Ausbildungsverhältnissen in Betrieben berichten."/ "Find documents on the dropout from vocational training in companies."* (GIRT topic 81)

The semantic network in figure 2 illustrates some inherent features of the WCFA: the disambiguation of a verb (the correct reading represented by the concept node *"berichten.2.2"*), the inclusion of additional lexical information (the edge labelled CHEA between the nodes *"abbrechen.2.1"* and *"abbruch.1.1"*) and the decomposition of a compound noun by means of additional relations and concepts (the edges and nodes for *"ausbildung.1.1"* and *"verhältnis.1.1"*).

3 **TRANSFOR:** A rule system produces a database-independent query representation (DIQR) from the semantic query representation. The transformation process identifies structures in the MultiNet representation and key relations between concepts to return an intermediate query representation containing the core description of a user's information need. The transformation engine and transformation rules are described in greater detail in [10].

A DIQR expression comprises common syntactical components to create database queries:

- Attributes (so-called semantic access points), such as *"author"*, *"publisher"*, or *"title"*.
- Term relations specifying how to search and match query terms. For example, the term relation *"<"* indicates that a matching document must contain a term with a value less than the given search term (for numbers).
- Term types indicating a data type for a term. Typical examples for term types are *"number"*, *"date"*, *"name"*, *"word"*, or *"phrase"*.
- Search terms identifying the concepts searched for. Search terms include multi-word expressions such as adjective noun phrases (such as *"allgemeinbildende Schule"/ "general education school"*).
- Boolean operators for the combination of attributes, term relations, term types, search terms or expressions to construct more complex expressions, for example *"AND"* (conjunction) and *"OR"* (disjunction).

Figure 3: Overview of some important relations defined in the MultiNet paradigm

MultiNet relation	Short description
ASSOC	relation of association
ATTCH	attachment of object to object
CHEA	change of event to abstractum
LOC	relation specifying the location for a situation
PRED	predicative concept specifying a plurality
SCAR	carrier of a state
SSPE	entity specifying a state
SUB	conceptual subordination for objects
SUBR	metarelation for the description of relations
SUBS	conceptual subordination for situations
TEMP	relation specifying a temporal restriction for a situation
*IN	a location-producing function

For example, when a descriptor search is limited to the title attribute, the semantic network from figure 2 would be transformed into the DIQR

```
(AND (media_object =   (word 'dokument.1.1'))
     (title = (AND (OR (word 'abbruch.1.1')
                       (word 'abbrechen.2.1'))
                   (OR (word 'ausbildungsverhältnis.1.1')
                       (AND (word 'ausbildung.1.1')
                            (word 'verhältnis.1.1')))
                   (word 'betrieb.2.1'))))
```

4 **IR-Client:** The IR-Client (Information Retrieval-Client) handles Z39.50 database access, database query processing, and document retrieval. The DIQR is transformed into queries of the Z39.50 protocol. After adapting a query to search features supported by a database it can be submitted to the database. The result set of document representations matching the query is retrieved.

5 **ANSWER:** The ANSWER module generates an answer representation in response to the natural language query (e.g. , it processes the documents which are stored in the database in a bibliographic format, such as USMarc, to return a list of documents in a form readable by the user). The results found can then be presented to the user.

The GIRT4 retrieval experiments were performed on the basis of the existing module infrastructure of the NLI-Z39.50 (the User-Client, WCFA, TRANSFOR, IR-Client and ANSWER module).

3 System Changes for the GIRT Experiments

The two major modifications to provide an experimental setup are the creation of a database in order to access the GIRT4 data via the Z39.50 protocol and enhancements of the IR-Client to support queries using the Z39.50 relevance operator and ranked results sets.

3.1 The Z39.50 GIRT4 Database

A GIRT4 database supporting access via the Z39.50 protocol was created using a free software kit, Zebra [3]. The default indexing strategy was applied to the data, which means that words are indexed as full forms. No additional indexing techniques and no morphological or syntactical preprocessing of the SGML data were employed (i.e. no stopword removal, stemming, or decomposition of compound

nouns). Furthermore, the process for matching queries against documents remained unchanged in the Zebra software.

In contrast to most library databases, a Zebra database supports a relevance query operator and ranked result sets. The standard ranking algorithm implemented in Zebra is based upon a popular term weighting schema. A term weight is computed as the product of a term factor (*tf*) and an inverse document factor (*idf*). Combining *tf* and *idf* for terms in a document or in a query yields a term weight vector, which can be utilized to compute a score (see [12] for an overview of term weighting and ranking methods in information retrieval).

Two technical constraints affect the experimental setup: Firstly, a real world Z39.50 database accepts queries only up to a fixed length (e.g. the number of characters in the Z39.50 query string is limited). Secondly, disjunctions in a query (the Boolean OR operator) are restricted to a small number (about 3 or 4 operators in a query). Queries not meeting these constraints are rejected by the database with an error message. In the NLI-Z39.50, a query preprocessing step creates a set of disjunction-free queries by eliminating OR operators and decreasing the query length, thus reducing the effect of these technical constraints.

3.2 IR-Client

Early test runs with the GIRT topics were performed with a single Boolean query obtained from the topic description. The recall for these test runs was close to zero. In order to find more documents, query construction and processing in the IR-Client were adapted. For the retrieval experiments, the IR-Client was modified to apply the query relevance operator and to deal with ranked result sets.

The DIQR serves as input format for the IR-Client. For example, the query *"Kennst du Bücher von Dörner über Komplexitätsmanagement?"* is represented as:

```
(AND (author = (name 'dörner.0'))
     (title = (OR  (word 'komplexitätsmanagement.1.1')
               (AND (word 'komplexität.1.1')
                    (word 'management.1.1')))))
```

For a library database, disjunctions in this query are eliminated and the set of queries submitted in parallel to the target databases. A fixed number of results can be obtained by retrieving records and removing duplicate results until the result set contains the number of records wanted. In the DIQR, a disjunction already marks alternate search terms (in this case, a compound noun and its parts). Eliminating the disjunction yields two query variants:[1]

```
(AND (author = (name 'dörner.0'))
     (title  = (word 'komplexitätsmanagement.1.1')))
```

and

```
(AND (author = (name 'dörner.0'))
     (title  = (AND (word 'komplexität.1.1')
                    (word 'management.1.1'))))
```

In order to improve recall, a set of query variants is constructed utilizing NLP tools and background knowledge represented in MultiNet. The query variants are ordered by their semantic similarity to the original query, OQ. The top ranked queries are used to either a) construct a single database query by collecting all search terms in the top ranked query variants in a word list and retrieve documents up to a given result set size or b) perform multiple searches in succession, starting with the top ranked query variants and retrieving a document only if it scores better than any document in the current result set.[2] Both methods are investigated with the GIRT retrieval experiments.

[1] A query variant is a reformulation of the original query obtained by eliminating, adding, or substituting one or more query terms with other (linguistically related) terms.

[2] Disregarding technical constraints, a single query could be constructed from multiple query variants by syntactically combining the variants with disjunction operators, e.g. by joining them with an "OR".

3.2.1 Search Term Variants

Search terms variants are generated to increase the chance of matching query terms with database terms (the more terms are matched, the higher the chance for finding a relevant document is). The reverse approach, normalizing index terms and computing concept similarities for WordNet for linguistically motivated indexing is described in [1].

The WCFA and background knowledge represented as a large MultiNet provide a means to look up search term variants of the following types:

Orthographic variants
Search terms may occur in different orthographic variants in a query or a document. Orthographic variants include terms in new German spelling (such as *"Schiffahrt"* and *"Schifffahrt"*), terms with German "Umlauts" expanded to their two-letter representation (e.g. "Bänke" and "Baenke"), and different hyphenations. The WCFA and a set of orthographic rules generate orthographic variants for a search term. For retrieval experiments, these variants are considered to be equal to the original orthographic search term.

Morphologic variants
Morphology is concerned with the internal structure of words and consists of two subclasses:

- Inflectional morphology (for example plural forms for nouns (*"Stadt"/ "city"* vs. *"Städte"*/ "cities") or comparative and superlative forms for adjectives (*"gut"/ "good"* vs. *"besser"/ "better"*). Inflectional variants are obtained via the WCFA which returns a list of full forms for a given lexeme. From this set of full forms, the forms with a prefix string match in the set are eliminated. For example, the full form lookup for the lexeme *"buch.1.1"/ "book"* returns a list containing the words *"Buch"*, *"Buchs"*, *"Buches"*, *"Buche"*, *"Bücher"*, and *"Büchern"*. The forms *"Buchs"*, *"Buches"*, *"Buche"*, and *"Büchern"* are eliminated as variants, because they match the prefix string *"Buch"* or *"Bücher"*. The remaining term variants, *"Buch"* and *"Bücher"*, are considered to be equal to the original concept

- Derivational morphology (typically affecting the part-of-speech, for example *"Abbruch"/ "dropout"* vs. *"abbrechen"/ "drop_ out"*). Derivational variations can be looked up in HaGenLex and in the MultiNet representation with background knowledge. A typical example for a derivational term variation is shown in figure 2.

Lexical variants
The same meaning can be expressed using different words. Several knowledge and lexicon resources provide lexically and semantically related variants for a concept:

- A mapping of GermaNet to Hagenlex, in MultiNet representation, containing mostly synonymy and subordination relations (such as (*"ansehen.2.3"* SYNO *"betrachten.1.2"*) and (*"ansehen.1.1"* SUB *"wertschätzung.1.1"*)).

- HagenLex entries providing MultiNet relations between lexicalized concepts.

- A semantic network containing background knowledge for proper names (for example state names and their relations to inhabitants, language, and geographical information).

- A MultiNet representation of the GIRT thesaurus. The thesaurus entries were transformed semi-automatically into a large MultiNet. First, the text fields in the thesaurus entries were analyzed via the WCFA, obtaining a set of unconnected, disambiguated lexemes (concept nodes). Thesaurus relations were then transformed into MultiNet relations to connect the concepts (i.e. the *narrower-term* and *broader-term* relation correspond to the MultiNet relation SUB, *related-term* corresponds to ASSOC, and *use-instead* and *use-combination* correspond to SYNO) The combination of MultiNet representations for thesaurus entries forms a large network providing a semantic representation of the GIRT thesaurus.

Syntactic variation
Some syntactic variations are normalized by the WCFA. For example, the MultiNet representation in figure 2 contains a compound and parts of the compound.

A concept score (concept similarity) is assigned by computing the semantic similarity between a concept variation and the corresponding concept in the original DIQR, OQ. The scores are computed by a formula from [1], adapted and extended for MultiNet relations.
Semantic similarity between two concepts x and y for a subset of MultiNet relations:

$$sim(x,y) = \begin{cases} 1 & : \text{if } (x \text{ EQU } y) \text{ or } (y \text{ EQU } x) \text{ exists (equal variants)} \\ 0.9 & : \text{if } (x \text{ SYNO } y) \text{ or } (y \text{ SYNO } x) \text{ exists (synonymy)} \\ 0.7 & : \text{if } (x \text{ SUB } y) \text{ exists (hyponymy)} \\ 0.5 & : \text{if } (y \text{ SUB } x) \text{ exists (hypernymy)} \\ 0.6 & : \text{if } (x \text{ PARS } y) \text{ exists (meronymy)} \\ 0.4 & : \text{if } (y \text{ PARS } x) \text{ exists (holonymy)} \\ 0.3 & : \text{if } (x \text{ ASSOC } y) \text{ or } (y \text{ ASSOC } x) \text{ exists (related concepts)} \\ \ldots & : \ldots \end{cases} \quad (1)$$

For concepts connected via a path of relations, the semantic similarity is computed as the product of similarities along the path connecting them.

To produce a ranked set of query variants, the semantic similarity (a measure to compare different concepts or queries) between the original query in DIQR (OQ) and a variant is computed as the product of semantic similarities of their concepts.

3.3 The Automated Retrieval Strategy

The general idea for the automated retrieval strategy is to generate and process a set of query variants differing in their search terms. Using the DIQR as a starting point, the following steps are carried out:

1. For each search term (counting multi-word lexemes and adjective noun phrases as a single search term) occuring in the original DIQR, the set of linguistically related concepts is found (orthographic, morphologic, lexical, and syntactical variants). by means of the WCFA and from resources containing background knowledge. The MultiNet relation between two concepts determines their semantic similarity.

2. To generate a set of query variants, search term variants are computed and search terms in the original query are replaced by search term variants. The set of query variants is ranked, ordering queries by their score (the semantic similarity between a query variant and the original query representation).

3a. To construct a single database query, all search terms in the top ranked query variants are collected in a word list to form search terms in an extended query. The documents found are retrieved until the result set exceeds a fixed size.

3b. To perform multiple queries, the top ranked queries are used for retrieval. Documents with a score higher than the minimum score of a document in the result set are retrieved and inserted into the result set.

4. For each document retrieved, the following information is known:

   - The *database score* for the document, determined by the database's ranking schema (the Zebra *tf-idf* score).
   - The *query score* for the current query variant, which is equivalent to the semantic similarity between the original query (OQ) and the current query variant.

   Document scores are computed as the product of database score and query score. If multiple instances of a document are found and retrieved for different query variants, the maximum of both scores is taken. For Boolean queries and databases not supporting the relevance operator, a database score of 1.0 is assumed to introduce a ranked result set for Boolean queries.

The ranked result set consists of the set of documents retrieved, ordered by their document score.

Figure 4: Overview of parameters for retrieval experiments

Run ID	topic fields	background knowledge used?	query type	ranking
Run1:	TD	N	M	QD
Run2:	T	Y	M	QD
Run3:	D	Y	M	QD
Run4:	TD	Y	S	QD
Run5:	TD	Y	M	QD
Baseline:	TD	Y	M (Boolean)	Q

4 GIRT4 Retrieval Experiments

Retrieval results for five runs for the monolingual German GIRT task were submitted to the GIRT committee for relevance assessment. The parameters for these runs and for a baseline experiment, for which results were not submitted (included for comparison), are shown in figure 4.
The experimental parameters varied are:

- The topic fields used as a natural language query: title (T), description (D) or the combination of both (TD).
- Using lexicon and background knowledge, yes (Y) or no (N), as resources for generating term variants.
- Constructing multiple (M) queries or a single query (S) for retrieval.
- Ranking documents by a combination of query score and database score (QD) or by query score alone (Q).

Only search terms with a semantic similarity greater than 0.3 were used. To restrict the total number of query variants, the number of variants for a search term was limited to $1 + \frac{100}{2 \cdot \#\, concepts\ in\ OQ}$. As a default, search terms in a query were right-truncated and searched for in the abstract (TEXT) and title (TITLE) field of the documents. The 250 top ranked query variants were used for query construction and the maximum result set size was limited to 1000 documents. For experiments using the title and the description of a topic, both were analyzed and transformed separately and the resulting query representations were appended.

5 Results and Analysis

Experiments with the GIRT task for which results were submitted all rely on the use of the Z39.50 relevance operator. Because this operator is not supported by library databases, an additional experiment was conducted, which did not employ the relevance operator (the Baseline experiment).
The following observations can be made from the retrieval results:

- The Baseline experiment was performed with multiple Boolean queries for each topic and evaluated with the *trec_eval* program. For the 25 GIRT4 queries, 2117 documents are assessed as relevant. A total of 883 documents were found and retrieved in the Baseline experiment, of which 200 documents are relevant (average precision for all retrieved documents: 0.0809). For twelve queries the result set was empty.

  The Baseline experiment (data retrieval) shows that recall for Boolean queries is low (compared to the information retrieval experiments). Compared with the first experiments with near-zero recall, multiple query variants offer a considerable improvement in retrieval performance. For the NLI-Z39.50, this retrieval strategy is applicable without modification, because it does not require the (unsupported) relevance operator. For data retrieval in very large databases, a Boolean retrieval

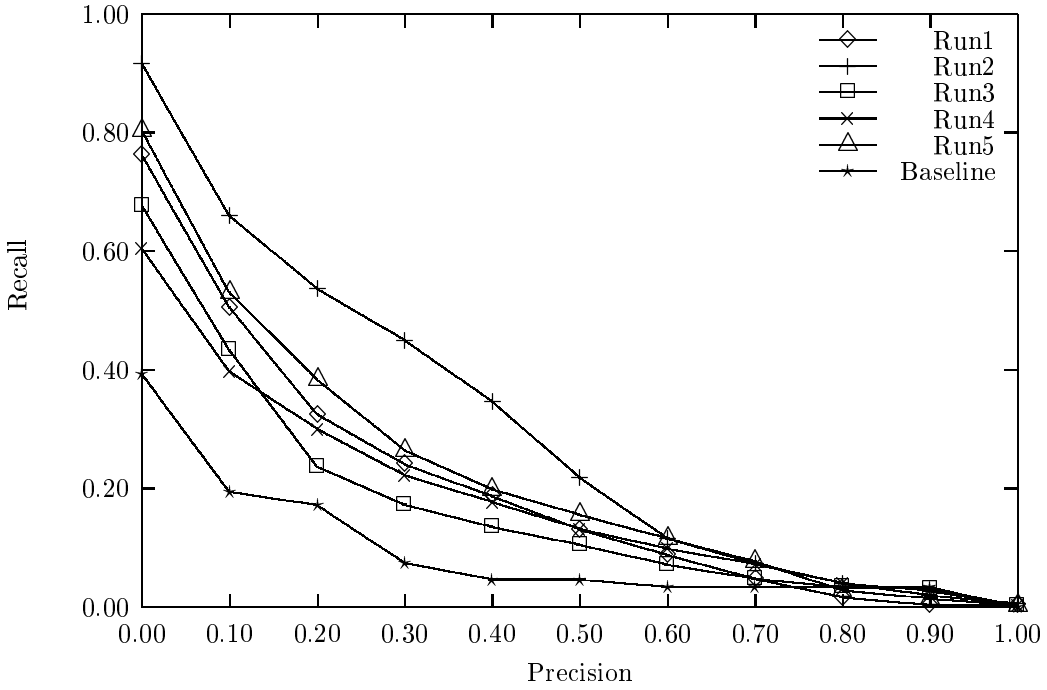

Figure 5: Recall-Precision Graph for GIRT4 (German-German) Retrieval Results

model may still be adequate, because a database of several million documents usually contains documents matching exactly with a given query. Information retrieval tasks rely on the concept of relevance, or – in this case – the Z39.50 relevance operator.

- The setup performs considerably better with shorter queries (Run2, title queries) than with longer queries (Run3, description queries). This is probably due to the fact that for shorter queries, more search term variants are employed to create the pool of query variants and the queries contain more terms different to the original query terms (in a set of at most 250 query variants).
- The strategy with multiple queries seems to perform better than the strategy of constructing a single query (Run5 vs. Run4). This is evidence that multiple queries (and multiple scores for a document) allow a finer granularity for ranking documents.
- Using additional lexical information and background knowledge shows a small improvement in retrieval performance (Run5 vs. Run1). For Run1, decomposition of compound nouns was still activated but other means to create search term variants were deactivated. The results imply that sufficient background knowledge in MultiNet representation is not yet available to show the expected significant increase in retrieval performance.

Summarizing, the automated retrieval strategy combined with additional lexical information and background knowledge still offers a high potential to increase retrieval performance. Additional methods like pseudo-relevance feedback or other means to produce query variants (such as omitting or adding search terms) have not yet been explored in combination with this approach.

6 Conclusions and Future Work

The experiments described in this paper show automated retrieval strategies providing significant increase in both recall and precision, compared with Boolean queries for data retrieval. The experimental setup was constrained by technical limitations of library databases to permit a practical application of the same

setup. Aiming at multiple heterogeneous databases the indexing or matching processes of a database cannot be changed as easily and cheaply as changing the interface software or the retrieval strategy.

Future research involves creating a database interface for which queries **and** documents are analyzed and transformed into semantic representations using NLP techniques. Finding matching documents then takes place on the level of meaning and knowledge representation. Some of the ideas presented here will be integrated into and implemented for the successor project of the NLI-Z39.50.

References

[1] Avi Arampatzis, Theo P. van der Weide, Patrick van Bommel, and Cornelius H. A. Koster. Linguistically Motivated Information Retrieval. In Allen Kent, editor, *Encyclopedia of Library and Information Science*, volume 69. Marcel Decker Inc., New York, 2000.

[2] R. Harald Baayen, Richard Piepenbrock, and Leon Gulikers. *The CELEX Lexical Database. Release 2 (CD-ROM).* Linguistic Data Consortium, University of Pennsylvania, Philadelphia, Pennsylvania, 1995.

[3] Sebastian Hammer, Adam Dickmeiss, Heikki Levanto, and Mike Taylor. *Zebra – User's Guide and Reference.* Index Data Aps, Ryesgade 3, Kopenhagen, 1995–2003.

[4] Sven Hartrumpf. *Hybrid Disambiguation in Natural Language Analysis.* PhD thesis, Fern-Universität Hagen, Fachbereich Informatik, Hagen, Germany, 2002.

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

[6] Hermann Helbig. *Die semantische Struktur natürlicher Sprache: Wissensrepräsentation mit MultiNet.* Springer, Berlin, 2001.

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

[8] Michael Kluck and Fredric C. Gey. The domain-specific task of CLEF — specific evaluation strategies in cross-language information retrieval. *LNCS*, 2069:48–56, 2001.

[9] Claudia Kunze and Andreas Wagner. Anwendungsperspektiven des GermaNet, eines lexikalisch-semantischen Netzes für das Deutsche. In Ingrid Lemberg, Bernhard Schröder, and Angelika Storrer, editors, *Chancen und Perspektiven computergestützter Lexikographie*, volume 107 of *Lexicographica Series Maior*, pages 229–246. Niemeyer, Tübingen, 2001.

[10] Johannes Leveling and Hermann Helbig. A robust natural language interface for access to bibliographic databases. In Nagib Callaos, Maurice Margenstern, and Belkis Sanchez, editors, *Proceedings of the 6th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2002)*, volume XI, pages 133–138, Orlando, Florida, 2002. International Institute of Informatics and Systemics (IIIS).

[11] Z39.50 Maintenance Agency (Library of Congress). *Information Retrieval (Z39.50): Application Service Definition and Protocol Specification for Open Systems Interconnection, ANSI/NISO Z39.50-1995 (version 3).* NISO Press, Bethesda, MD, 1995.

[12] Justin Zobel and Alistair Moffat. Exploring the similarity space. *SIGIR Forum*, 32(1):18–34, 1998.