RECENT DEVELOPMENTS IN HUMAN LANGUAGE TECHNOLOGY

Werner Winiwarter

Institute for Computer Science and Business Informatics, University of Vienna
Liebiggasse 4/3-4, A-1010 Vienna, Austria
werner.winiwarter@univie.ac.at

Abstract: *Human language technology has finally matured from the academic playground to become a serious candidate as one of the driving innovative forces towards making the vision of the new knowledge-based society come true. This paper gives an overview of the recent developments in this dynamic research field. After pointing out the main research issues, we present the results from several research projects conducted by the author in strong cooperation with several Austrian competence centers. We address the topics of voice interfaces for mobile commerce, ontological and knowledge engineering, multilingual interfaces, natural language information retrieval, and machine learning of natural language.*

1. Introduction

The research in the area of human language technology has reached a level of maturity that makes it a realistic undertaking to use it for solving many of the urgent problems of the coming information age. By gathering valuable experiences from past failures and successes, we are now able to apply this key technology to common tasks of everyday life. Multilinguality, mobile access to the Web via spoken language, and the automatic extraction of knowledge from documents are just a few examples which stress the importance of human language technology.

One of several examples that human language technology is no longer only a topic for academic research, but has already become of interest for industrial development, is the foundation of the company MobileArea. Palm Inc., Mayfield, and Delphi Automotive participate in this enterprise with the common aim to develop products that enable the access to personal data from a PDA and to information from the Internet while steering a vehicle. Driving a car is an optimal application environment for human language technology because spoken language is here the only communication medium that offers extensive information access without affecting the concentration of the driver. In addition, important information for navigation and traffic can be conveyed to the user.

Through the emerging market segment of mobile commerce the mobile phone will become more and more the universal interface to conduct business transactions. At the same time the outward appearance of mobile phones is going to change towards an integration with PDAs. Also in this context, spoken language offers an ideal medium for information search in the Web and the

processing of transactions. In addition, the unrestricted input of handwritten data in sensitive environments, where the use of spoken language is inappropriate, represents an important prerequisite for the optimal utilization of PDAs.

Embedded systems will bring even more far-reaching changes by adding intelligent interfaces to all our daily surroundings. First examples are refrigerators with Internet access or talking shelves in supermarkets. All these developments point at the direction that the traditional data processing front-end computer with keyboard, screen, and mouse will soon be eclipsed by this new ambient intelligence. Human language technology provides the necessary instruments for such a decisive thrust of innovation, which will alter our whole society.

In order to be able to process user input in an efficient and correct way, the natural language interface has to have access to the relevant background knowledge. This includes the compilation of ontologies, which provide a formal representation of concepts and their relationships. A second important condition for a consistent mapping of linguistic surface forms onto internal representations is a comprehensive integration of information from heterogeneous sources.

Another relevant aspect is multilinguality. Currently there are many efforts on their way to establish global digital libraries. To overcome the language barriers by using cross-language information retrieval and machine translation is pivotal to the global usage of these invaluable information resources.

A final essential point is to eliminate the static behavior of many existing human language technology products. Users of such intelligent systems have high expectations in their usability. Missing flexibility in reacting to the user behavior leads quickly to rejection and frustration. Therefore, it is necessary to add adaptive behavior to the systems, i.e. the user behavior is monitored and the system response is adjusted accordingly. For this purpose methods from user modeling and machine learning can be applied. The aim is to derive the user's expectations, previous knowledge, and preferences to guarantee an optimal personalization. At the same time, it is important to consider privacy aspects, i.e. the sensitive handling of user data.

The European Union has recognized the high potential of human language technology for quite some time. The history of EU-funded research on human language technology started with projects within the ESPRIT and EUROTRA action lines. 1991 the Linguistic Research and Engineering (LRE) program was initiated as part of the Third Framework Program. After the transitional program MultiLingual Action Plan (MLAP) the Language Engineering Sector was extended to a budget of over 80 million ECU within the Fourth Framework Program. In the Fifth Framework Program human language technology was part of the Key Action III (Multimedia Contents and Tools) of the IST program with a budget of over 564 million Euro. Finally, also in the new Sixth Framework Program the European Union has again stressed the importance of human language technology by stating as key actions the research on ambient intelligence, mobile commerce, knowledge representation and management systems, multisensorial interfaces capable of understanding and interpreting the natural expression of human beings, and multilinguistic and multicultural systems.

In Austria the research on human language technology has been very active during the last few years. An important innovative force are the competence centers, which have been established recently through an RTD program initiated by the Austrian government to stimulate the long-term cooperation between innovative enterprises and top-quality research in order to contribute to a lasting improvement of the cooperation between science and industry. Several of these competence centers have dedicated their activities to information and communication technologies, which include several promising research projects on human language technology, e.g. the Forschungszentrum Telekommunikation Wien has been working on a speech database for automatic speech recognition in telephony, the E-Commerce Competence Center on adaptive multilingual interfaces, and the Software Competence Center Hagenberg on natural language information retrieval.

A good recent overview of the current state of the art in human language technology in Austria can be found in a special issue of the ÖGAI Journal [20]. It reports on the research work of Austrian researchers working on topics such as conceptual modeling of information systems based on requirements specifications in natural language, multimodal search engines for news messages, embedded adaptive machine translation environments, multilingual terminologies and ontologies for the Semantic Web, and speech and multimodal dialogue systems for telephony applications.

This paper presents results from several recent research projects conducted by the author in strong cooperation with the three competence centers mentioned above. Section 2 discusses the role of VoiceXML in enabling voice interfaces for commercial applications and the potential for its integration with intelligent component technologies. In Section 3 we address linguistic aspects in problem-driven knowledge engineering for the specification of ontologies. Section 4 describes the development of an interface for multilingual natural language access to tourism information, and Section 5 deals with the question of how useful syntactic analysis of queries really is for information retrieval. Finally, Section 6 presents a multilingual natural language interface for e-commerce applications, which makes use of a rule-based machine learning module to guarantee adaptive behavior.

2. The Potential of VoiceXML for Voice-Enabled Mobile Commerce

VoiceXML [25] offers the prospect of a streamlined deployment process of voice interfaces for commercial applications, similar to the ease of developing conventional electronic commerce applications. However, as it is, the capabilities of VoiceXML regarding natural language processing facilities are very limited.

In a joint research project of the Software Competence Center Hagenberg and the Forschungszentrum Telekommunikation Wien we investigated opportunities and constraints for the integration of intelligent component technologies with VoiceXML-based systems [8]. Such components will solve advanced tasks from both natural language analysis and generation.

A VoiceXML voice browser is a software platform running on a network server with the task of enabling access to Web applications for users connected via speech devices. For this task the VoiceXML voice browser supports the following features, which directly influence where

intelligent component technologies might be necessary or beneficial, and how they might be integrated:

- *Automatic speech recognition (ASR):* Existing VoiceXML platforms support speaker-independent speech recognition for telephony speech. The interface to the forms and menus of a VoiceXML dialog is performed via raw text. In VoiceXML 1.0 it is not possible to supply the VoiceXML browser with more fine-grained information, such as n-best lists, confidence scores, or signal-noise ratio, which is a severe obstacle for any attempt to provide advanced environment adaptation.
- *Dual tone multi-frequency (DTMF):* In addition to ASR input, VoiceXML platforms provide support for DTMF input via telephone keypads.
- *Recognition grammars:* To match the ASR result with the active input choices at each dialog step, it is possible to specify grammars for the valid input choices. These recognition grammars directly affect the language model used in an application and may not work well together with natural language processing systems based on very different linguistic models.
- *Mixed-initiative dialog:* To model mixed-initiative dialogs, in which the user has some flexibility in choosing the sequence of inputs and interactions, the standard requires that it must be possible to have more than one input field active at the same time. The way mixed-initiative dialog is modeled in VoiceXML imposes a certain structure of dialog management and restricts the dialog to several classes of speech acts. Natural language understanding techniques may prove beneficial, e.g. for anaphora resolution or to deal with overlapping linguistic spaces of concurrently active recognition grammars.
- *Text-to-speech synthesis (TTS):* VoiceXML platforms must provide TTS to play prompts to the user. This way, the prompts can be specified as simple text in VoiceXML documents, which is convenient for the dynamic generation of such documents. TTS is a prerequisite for natural language generation.
- *Barge-in:* Barge-in enables the system to process speech input even while playing TTS output. Therefore, the user can interrupt system prompts with a response. However, barge-in increases the complexity of speech recognition, resulting in the need for more advanced natural language understanding techniques.
- *ECMAScript support:* VoiceXML documents can include or reference ECMAScript code to perform computations. Since VoiceXML documents are explicitly not intended to perform heavy computations, natural language processing cannot be done effectively in VoiceXML itself. One feasible possibility is to integrate separate natural language components on a dedicated server via ECMAScript as interface.
- *HTTP interface:* The VoiceXML interpreter provides POST and GET for the communication with Web servers to submit user input and to request a VoiceXML document for a new dialog or subdialog.
- *Support for proprietary extensions:* VoiceXML provides a special object-tag which can be used by platform providers to link proprietary extensions of the standard VoiceXML interpreter to applications. This has been used, e.g. to link arbitrary Java code to a VoiceXML document. While representing a gateway to powerful extensions, this also threatens the idea of VoiceXML being a uniform standard. Some providers went even further and invented proprietary tags, which means that such VoiceXML documents cannot be compiled on every platform anymore.

We identified several problems in VoiceXML applications which could be solved by applying more sophisticated natural language analysis techniques:

- Pure menu-driven interfaces may be experienced as unnatural and cumbersome. Natural language interfaces require components for understanding and related linguistic tasks in order to be used efficiently.
- An explicit enumeration or description of the objects in a domain and their interactions by using a computationally simple grammar (e.g. finite-state grammar) is often impossible or infeasible. In such cases natural language understanding techniques can be used to map an utterance to a formal representation that can be processed by the system.
- Users make full use of ellipses and anaphora in realistic dialogs. They cannot be resolved without sophisticated linguistic analysis and the application of semantic knowledge.
- As mentioned before, in a mixed-initiative dialog complex utterances are to be expected, which cannot be handled efficiently by the mechanisms of the VoiceXML language. Again, an efficent solution is to delegate the task to an natural language understanding module.

The design decision whether and how to use natural language understanding techniques is, among other considerations, a question of the system architecture. First, the whole system must allow input that is ambiguous enough to warrant higher-level natural language understanding. In VoiceXML the most important element constraining the input form is the recognition grammar. A grammar rejects any utterances that do not fit its expectations and thus imposes structure on the input. Therefore, there exists a tradeoff between constraints and simplicity vs. freedom and difficult interpretation, which also greatly influences the role natural language understanding can play. We can choose among several possible interfaces to link VoiceXML with natural language understanding modules:

- *Raw speech data:* An extreme approach is to completely bypass the speech recognition mechanisms of VoiceXML. It is questionable if such an approach is feasible in terms of transfer overhead and whether such a design basically makes sense.
- *Trivial grammar:* A similar idea is to use only a trivial grammar, which recognizes any arbitrary string of words. The main shortcoming of this approach is that the grammar no longer restricts the search space for the ASR engine, resulting in deteriorated recognition results.
- *Keyword or phrase spotting:* The Working Draft of VoiceXML 2.0 [26] includes the Speech Recognition Grammar Format (SRGF), which allows to ignore input via a special $GARBAGE rule. With this mechanism keyword spotting grammars can be designed.
- *Complex grammars:* Although the supported complexity of SRGF grammars is limited, SRGF provides several powerful mechanisms such as rule expansion, (limited) recursion, or regular-expression-style constructs. This permits a great deal of flexibility to build models that are close approximations of linguistic constraints. However, so far there exists only little experience with designing such grammars and regarding their feasibility in comparison with conventional grammars from computational linguistics.
- *Natural language understanding as fallback:* A completely different approach is to let VoiceXML handle an utterance and to use natural language support only as a fallback when VoiceXML fails.

Besides natural language analysis, the second main area from human language technology that has a high potential to enhance the functionality of VoiceXML platforms is natural language generation. VoiceXML offers two ways to produce the system prompts for the user: via the playback of prerecorded speech or via TTS [24]. Prompts for TTS are specified as text with an option to add prosodic information.

Such pre-specified prompts are a good choice in applications where the prompts are known before runtime. Some amount of variation can be captured via templates, i.e. prompt specifications with variable slots. However, the more flexibility is needed for an application, the more attractive dynamic natural language generation becomes. Two approaches are conceivable:

- Applications where speech acts are pre-specified as expandable templates. The exact wordings are generated based on the type and amount of information to be fitted into these speech acts.
- Applications where an information-rich lexicon is combined with a text planner and a tactical generator to produce utterances from deep semantic representations.

An important factor for the usefulness of natural language generation for VoiceXML applications is its support of multilinguality. By using a strictly modular component for each language, in principle the modules for the application manager and the dialog manager can be used unchanged across the supported languages. Another important demand for natural language generation is that it should operate on an application-independent level so that it could become part of a generic dialog system platform. However, with respect to the importance of idiomatic expressions and application-specific, elliptic expressions it seems difficult to achieve this goal.

VoiceXML supports the definition of prompts via the value of an ECMAScript variable. However, the implementation of a natural language generation component in ECMAScript seems not to be a feasible option. Therefore, some providers have used the proprietary object-tag to plug in a natural language generation component into VoiceXML dialogs to be invoked by synchronous function calls. This restricted integration model has the disadvantage that the other attributes of the active VoiceXML dialog remain static and cannot reflect the computation of the dialog manager, e.g. by providing a new voice input field. Thus, an extended integration model is to generate not only prompts dynamically, but new VoiceXML documents to contain also the subsequent logic. This dynamic generation of VoiceXML documents has also several other advantages, e.g. the support of multimodal interfaces.

To conclude this section, we want to state that we believe in VoiceXML as an important technological development. However, this development is based more on its economic merit than on innovative technical substance. The industrial interests behind the VoiceXML standard have shaped it into its current form. The intelligent component technologies we have investigated do not play a direct role in this set of interests. We have identified a high potential for natural language understanding for VoiceXML applications. The key issue that has to be solved is about grammar formats where one has to find the delicate balance between simplicity and robustness. For integrating natural language generation and VoiceXML there is no serious technical difficulty. However, its need must be motivated, which is mainly the case for multilingual and application-independent settings.

3. Linguistic Aspects in Knowledge Engineering

The eagerly awaited explosive growth in x-commerce (i.e. e-commerce, m-commerce, ubiquitous commerce, voice commerce, content commerce, etc.) is widely considered impossible without further semantic integration. There are remarkable efforts towards the latter as far as the syntactic preconditions are concerned (i.e. XML-based standards). Ontologies are regarded as a means for "real" semantic integration [5], however, current approaches do not consider linguistic insights to an appropriate extent.

Today we can witness three developments concerning the future evolution in the use of the Internet:

- the *technical convergence*, i.e. the ongoing technical standardization,
- a need for *social convergence* between multicultural and multilingual communities, and
- the growth in the area of electronic trading in the Web requiring *economic convergence* or interoperability between commercial agents.

We can view the problem of interoperability from the discipline of semiotics [12]. The Internet is then a huge collection of signs in the semiotic sense to which we assign:

- an entity that represents
- another entity to
- an agent.

If we only look at the signs themselves, we lose sight of the entities they represent and of the agents who interpret them. In such a sign system we have two perspectives:

- the representational perspective on knowledge, i.e. the problem of *reference* and
- the recipient's perspective, i.e. the problem of *usage* and *interpretation*.

Both perspectives are involved in the constitution of meaning and interfere with each other in "real world" problems. In this context, the initiative for a Semantic Web could be understood as an attempt to isolate domains of knowledge, which have communities associated that can agree upon standardized, unambiguous semantics.

In the context of a research project conducted at the E-Commerce Competence Center (EC3) we analyzed the knowledge base developed at the EC3 as the central part of an x-commerce architecture from a linguistic modeling perspective [15]. We addressed problems that have not been focused at within the fields of knowledge and ontological engineering so far.

Figure 1 shows the components of our architecture and the relations between the different tasks, which are interconnected directly or indirectly through the *knowledge base*. The development of the knowledge base includes the construction of a consistent and comprehensive ontology, the representation of business process knowledge, and the modeling of user data.

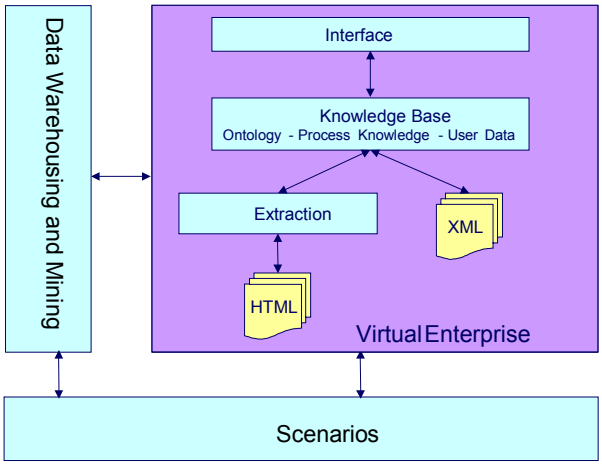

Figure 1: X-commerce architecture

The foundation of the architecture is the *scenarios* component, which applies a mix of techniques from market research, business development, data mining, experimental economics, and simulation to the analysis of new application scenarios. The aim is to better understand the needs of the user, to find business paths including aspects of usage and technology, to detect potential barriers, and to reduce the risk of the introduction of new services. The *data warehousing and mining* component monitors and analyzes usage data as valuable input for all other modules. Together with the scenarios component an environment for a virtual enterprise is established, which offers complex products or services to customers in a customized way.

The *extraction* module deals with the dynamic aspects of the system, i.e. how to add new knowledge to the virtual enterprise. The easiest way to insert information is directly via a standardized format. However, most often knowledge is only available implicitly in form of information embedded in text or hypermedia documents. Therefore, we have to apply information extraction techniques to filter the relevant knowledge. In addition, we also aim at the consistent integration of information coming from multiple sources. Finally, the *interface* module is realized as an multilingual natural language interface (see Section 4 for details).

The ontological knowledge contained in the knowledge base guides many computational processes in the other modules. Therefore, a sound design of the domain ontology considering also linguistic principles was of paramount importance for the overall success of our x-commerce architecture. The current approaches to ontology design do often neglect both the perspectives of usage and reference. They merely focus instead on the relations between the signs, even though there are far more relations to be considered depending on the tasks to be performed by the ontology. We try to analyze problems related to different possible tasks by applying linguistic theories. We concentrate on semantic theory, lexical semantics, rhetorics, and cognitive issues. Additional insights are gained from pragmatics, domain-specific semantic theories, socio-linguistics, and discourse representation.

In the different approaches to semantic theory there are two extreme positions [6]. The *analytic approach* denotes semantic operationalization from within (the language system) without reference to external “semantic sources”, i.e. the semantics of concrete referential relations. It corresponds to a logical deduction based upon semantic features within a “flat space” of sign

relations, resulting in a "standalone" system. The analytic approach can also be called *operational* because semantic features provide the basis of calculus and axioms within formalized systems. In contrast, the *synthetic approach* is based upon empirical evidence taking into account concrete referential relations between signs and the objects they refer to. In this sense, the synthetic approach can also be called *operative*, because of the concreteness of the instantiated referential relations.

Within the architecture of our knowledge base, both approaches play an important role and correspond with distinct components. The problems of reference and usage are closely related to the synthetic approach: concrete reference and specific usage result in (systems of) signs that have to be interpreted in terms of and mapped to a core (language or sign) system. Therefore, we propose a two-layered architecture:

- a *core language*, that is internally operationalized, and
- a *translation layer*, that deals with the problems of usage and concrete reference.

As a motivational example we take the requirement for our virtual enterprise from the application domain of tourism to find acceptable alternatives for the query term "hotel". SeI and SeII show 2 possible natural language user queries:

SeI: I am looking for a hotel near to the center
SeII: I am looking for a hotel with an Art-Nouveau façade

For the term "hotel" we can find three related groups of concepts in our ontology:

GrI: hotel, motel, guesthouse, pension
GrII: room, suite
GrIII: bed, double-bed

Now, if we look at SeI then members of all three groups are plausible constituents. The problem that we encounter is that we cannot deduce the correctness of members from GrII or GrIII as semantically acceptable alternatives to <hotel> in SeI from a purely classifying ontological system: neither are <bed> or <room> subtypes of accommodation, nor is there a hierarchical relation between <bed> and <room>. Apparently, the semantic acceptability of a constituent depends on the *cotext* (the textual environment) and the *context* (the situational pragmatics). This co(n)text lets us find an adequate interpretation for the metonymical relations between the members of the three groups. In contrast, only GrIII can be regarded as semantically acceptable within the co(n)text of SeII.

Therefore, a translation of the uttered and overt representation into the actually meant object (resolving of the reference relation) seems to be a precondition for any semantic validation. We wanted to demonstrate the need for empirical-referential semantics, since in the research fields of Semantic Web or ontologies there are still few efforts towards this direction. Apart of and after the synthetic translation, we still need an analytic component to check for semantic acceptability itself. Thus, we need both the analytic and synthetic approach. They cannot be isolated and thus

be seen as dichotomic, neither as far as the semantics of natural language is concerned nor in relation to the architecture of our knowledge base.

The linguistic concept of *markedness/unmarkedness* can make further contributions to the task of semantic interpretation, i.e. asking for the user's intentions. Coming back to the previous example, the question is now which of the three groups is the most natural or statistically probable alternative for SeI. The concept of markedness/unmarkedness relates first of all to the extralingual context, i.e. to the pragmatics of the situation. Let us assume the following three assignments with the otherwise abstract scenario ScI for SeI:

AssI: A person with backpack might be asking rather for a <bed>.
AssII: A business man we would expect to look for a <hotel> in the first place.
AssIII: For the abstract scenario – without a person associated – <room> seems to be the most natural constituent, since <room> is essentially the entity that is to be rented. Neither do we rent only a bed nor the entire hotel.

Now we have the interesting situation that there are three "most natural" or unmarked constituents for ScI depending on whether we take into account the (non-textual) context or not. The mere interpretation of the cotext supports only inferences on what can be identified as semantically acceptable alternative constituents. Including the non-textual context enables two additional directions of inference:

IdI: The system knows who is the actor in the user-role and can thus infer the "most natural" alternative constituent based on the user characteristics. Then, this can be contrasted to the used constituent (the actual input). Any difference, "consciously uttered markedness" so to speak, supports then higher degrees of adaptivity of the user interface.
IdII: If the system has no information about the user, it may infer some of his characteristics depending on the used constituent (the actual input), since the most natural constituent is also the one with the (statistically) highest probability.

Thus, for both cases the concept of markedness/unmarkedness leads to a better understanding of the user's intention. Furthermore, the concept of naturalness is based on statistical probability, which corresponds to the traces of the user's interactions with the system.

In this section we tried to show that linguistics and its subdisciplines provide a promising and mighty framework for theoretical reflections about how to address common and also more specific issues in modeling and constructing knowledge bases. In future research on this topic we target at apt models for the user, the situation, and specific touristic knowledge domains with a focus on a general solution for the context-sensitivity of knowledge and on translation issues (user language, multilinguality, schema mapping, and information extraction). Regarding the notational aspects we will focus on how to annotate the knowledge structures by means of current notational standards such as RDF-Schema, DAML+OIL, or Topic Maps. Finally, concerning the implementation, we focus on the problem of combining metadata and their instantiation at runtime (hybrid databases, serialization, etc.) as well as on research about what kind of more complex inferences are feasible to support within our x-commerce architecture.

4. Multilingual Access to Tourism Information

Natural language interfaces are a continuing research topic in computer science since the very first days of this discipline (for a good survey of the field see [1]). Natural language is especially appealing as an interface for database queries because the user is able to express his information request without the need to learn a formal query language. Human language technology is also a potential key to the success of applications in e-commerce. In particular, the provision of multilingual access to information resources is crucial, even more so in such a multilingual environment as Europe. As part of the x-commerce architecture developed at the E-Commerce Competence Center (see Section 3) we have designed and implemented an interface prototype called AD.M.I.N. for multilingual natural language access to tourism data [3].

At present, the user may express his information requests for accommodation in English or German; the language of the query is automatically detected by the system based on an n-gram text classification approach [4]. This makes it fairly simple to provide for an extension to additional languages. We used sentence parts out of news stories for various topics as training set. The n-grams (n=1...5) in this training set were analyzed, sorted according to their frequency, and stored as frequency profile for a language. To identify the language of a query, the n-gram profile of the query is computed first. Then, for each n-gram occurring in the query, the difference between the rank of the n-gram in the query profile and the rank in a language profile is calculated. The sum of these differences gives the distance between the query and the language profile. The language with the smallest difference is selected, however, if this distance is still greater than a certain threshold, the user is asked to rephrase his query.

After the successful identification of the language of a query the query string is transferred to the query interpretation module. The basic idea of its design is to extract the requested concepts of the application domain from the natural language query string. Based on these concepts the final SQL query is composed of "SQL-fragments", i.e. SQL statements that are available for a wide range of different query patterns. The computation within the query interpretation module is performed in several steps:

- The *NumConverter* recognizes numerals, e.g. "eleven", and converts them to digits, e.g. "11".
- The *QueryCleaner* performs the tokenization of the string and discards all terms which cannot be found either in the domain ontology or in the database.
- The *QueryRewriter* replaces each word with its preferred term as indicated in the ontology.
- The *Tagger* tags the remaining query words to add semantic information, which is later used in particular to determine which modifiers can be expected in the neighborhood of the individual words.
- The *SQLQueryGenerator* builds the SQL query by selecting, filling, and combining the necessary SQL-fragments based on the identified query concepts and their modifers.

As an example of query processing we take the English user query: "Show me all the farms close to Imst where pets are allowed". For this query the following words are identified as being relevant for the application domain: "farms", "close to", "Imst", and "pets". The word "farms" is

classified as a particular type of accommodation, i.e. *"typ Bauernhof"*, in our ontology. The word "pet" is identified as an additional restriction on the type of accommodation, i.e. only those accommodations where the flag for "accompanying pets are allowed *(einrichtung haustiere)*" is set in their descriptions. *"Imst"* is retrieved as the name of a city in Tyrol, Austria and, finally, "close to" is recognized as an restriction *("nahe")* on the location of the accommodation. The result of the query is shown in Fig. 2. The language-independent representation requires the retrieved objects to be of type "typ Bauernhof", near ("nahe") to the city of "Imst" and allowing pets ("einrichtung haustiere").

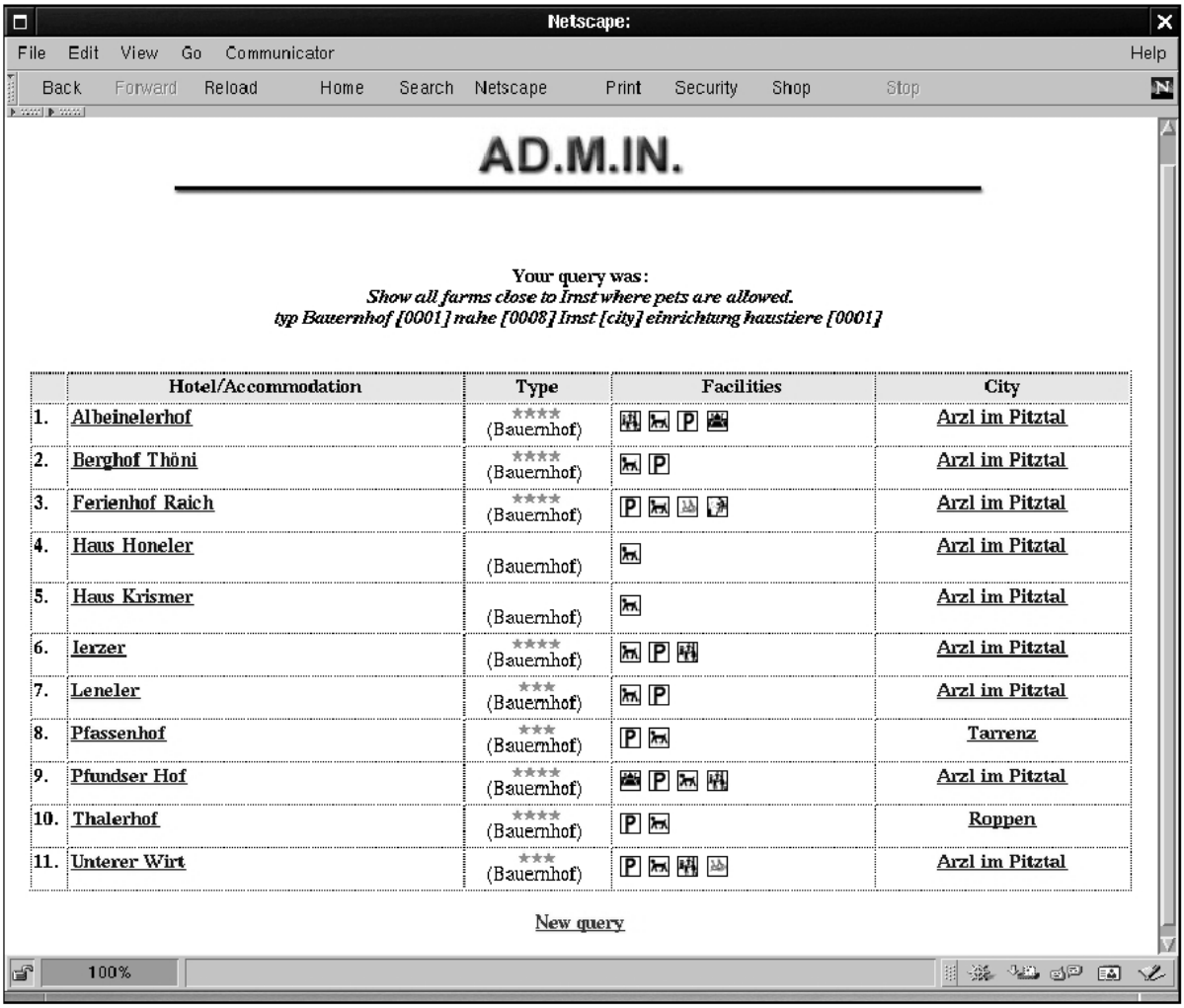

Figure 2: Example of query result

In this section we have described a multilingual natural language database interface for tourism information. Its main features are language identification by using n-grams, query analysis based on keyword matching and the application of human language technology, the use of a domain ontology, and a Web-based user interface for query formulation and the presentation of query results. Future work will focus on enabling an intelligent dialog between the user and the system, increasing the robustness regarding wrong or missing information, and personalizing the communication according to the needs of the individual users.

5. Syntactic Analysis of Queries in Information Retrieval

Up to now, the results of applying sophisticated human language technology to information retrieval have been mostly disappointing [11, 13, 16]. In a research project at the Software Competence Center Hagenberg we investigated in detail the role of syntactic analysis in information retrieval and tried to find answers to the question why it works better for some queries and worse for others [7].

It is a weakness of the vector space model of information retrieval [2] that the vector representation throws all words of a query together into one structure and does not allow the modeling of any relationships between subsets of words. Intuitively, it is obvious that the usage and the importance of a word depend on the context in which it is used. In the algorithm that we developed we exploit the syntactic context of words, hoping that part of the semantic context is thereby indirectly captured as well.

Our algorithm consists of three main components (see Fig. 3). The first module parses the query and analyzes its syntactic structure, the second module processes and weighs the words, and the third module aggregates the two intermediate results to a final score ranking.

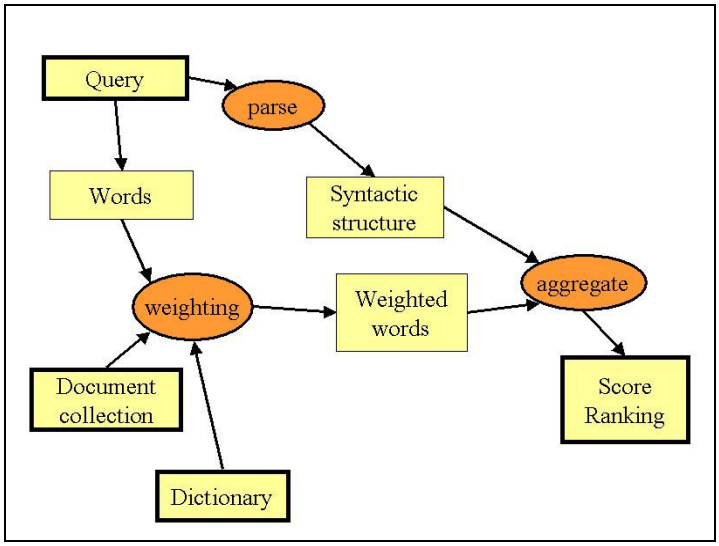

Figure 3: Natural language information retrieval algorithm

The first task of the parsing component is to build a graph representation of the query. In our implementation we use the Link Grammar Parser (LGP) [10, 14] for this task. The query is first divided into sentences, and then, for each sentence, the LGP produces a set of possible parses. Each parse can be represented as a constituent tree; the trees are combined into a parse lattice such that identical phrases are shared among trees to avoid the computational effort of processing each parse separately. From this representation we compute a measure of connectedness *c* for each pair of words in the query:

$$c(w_1, w_2) := \min_a d(N_{w_1}, a) + d(N_{w_2}, a) \quad (1)$$

N_w is the tree node corresponding to word w, d measures the distance between two nodes, and a is a common ancestor of the nodes. As distance measure we choose the sum of the distances to the closest common ancestor of the two words in the parse tree. The quantity (1) indicates how strongly two words are bound together by the syntactic structure of the query; this value is later used by the aggregation component.

The task of the weighing module is to provide candidates for filling the terminal nodes of the syntactic representation. So far, we only use the original set of words contained in the query. In future we plan to extend this set by adding synonyms for each word. The function s calculates a value for the importance of a word:

$$s(w) := idf(w) \tag{2}$$

This means, currently, we only use the conventional IDF rating method. However, in future we plan to incorporate other resources into the weighing process, such as collection-independent frequency tables, or possibly even resources that allow to modify a weight based on qualitative restrictions like context. Whenever two candidates are both present in a document, the difference of their positions is calculated; in the case of several occurrences as the minimum difference. This difference is transformed into the *positional score factor* psf_{doc} by an exponential decaying function:

$$psf_{doc}(w_1, w_2) := e^{-c_{psf} d_{\min}(w_1, w_2)} \tag{3}$$

The term $d_{\min}(w_1,w_2)$ is the positional difference of the closest occurrence of w_1 and w_2, measured in words. The reasoning behind (3) is to approximate syntactic connectedness in the document by spatial closeness; the transformation function should be continuous, monotonically decreasing, and experimentally tuned such that it agrees reasonably with the typical spatial distance of words which are in fact syntactically connected. An analogous transformation function is applied to the measure of connectedness c computed by the parsing module to obtain the *structure score factor ssf*.

The *basic word score bws* is computed from the two individual word scores by using the geometric mean:

$$bws(w_1, w_2) := \sqrt{s(w_1)s(w_2)} \tag{4}$$

Finally, the aggregation module calculates an *overall score* os_{doc} from the positional score factor, the structure score factor, and the basic word score:

$$os_{doc}(w_1) := \max_{w_2 \neq w_1} bws(w_1, w_2) ssf(w_1, w_2) psf_{doc}(w_1, w_2) \tag{5}$$

The final score of a document ds_{doc} of a document *doc* is the sum of the overall scores for the document of all query words.

For our experimental evaluation we used the well-known CACM test collection (3204 documents, 64 queries). We compared our algorithm to a baseline of simple term weighing (basic vector space model without modifications like query expansion, etc.). The current version of the system does not perform significantly different from the baseline: 34.8% precision for the baseline, 34.6% for our system. However, the interesting point is to analyze the performance differences for different types of queries. Our system outperforms the baseline system for some query types whereas it is in turn outperformed for others (see Fig. 4).

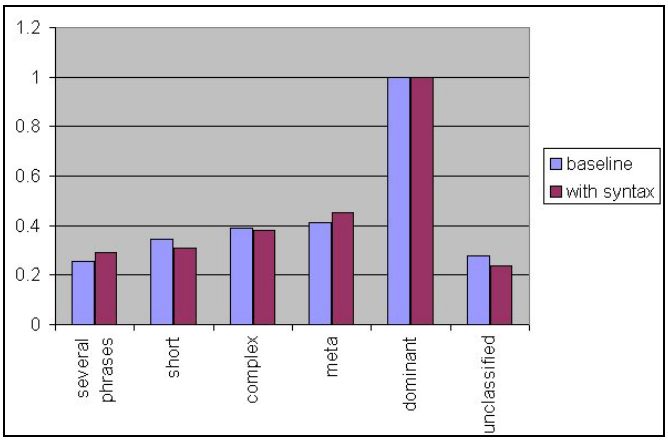

Figure 4: Average precision by query category

In the following, we present a detailed analysis in which we attempt to categorize the queries and to reveal reasons why some queries are better or worse suited for being handled by our approach:

- *Very short queries:* Our approach rewards co-occurring words which are in the same region of the parse lattice. In the case of very short queries, however, there is only one region. In other words, there is no conceptual difference between our approach and the simple vector space model. Since the exact parameterization of the latter has been optimized over the years, its superior performance is no real surprise.
- *Complex or ungrammatical queries:* As the other extreme, such queries make it difficult for the parser to produce useful output. In addition, the complex structures generate larger lattices where semantically related items have a larger distance from each other than in an equivalent simpler formulation.
- *Queries with meta-level content:* Some queries contain phrases or whole sentences that do not describe a topic, but the search process itself or the user's preferences, e.g. "I'm interested in" or "find all descriptions of". Whereas our approach outperforms the baseline for this category, there are some problems with semantically empty functional constructs like "the use of" that hurt our approach more than the vector space model. The reason is that we consider such tight syntactic couplings important and assign disproportionately large bonuses to any occurrences thereof.
- *Queries with several phrases:* Such queries seem to be well suited for our approach. The effect is the greater the more phrases there are in the query, and the more probable it is for words to co-occur randomly. The latter point is especially important for very frequent words like "time" or "system". They occur too often to be useful as keywords, but are significant as part of a phrase like "time sharing" or "operating system".

- *Queries with a dominant term:* Some queries contain a very specific term that selects exactly the relevant documents and therefore overshadows the effects of all other terms. For such cases there is no difference between the baseline and our system.

In this section we have presented an algorithm for natural language information retrieval. We compared the performance with simple term weighing and realized that it really depends on the type of query whether our approach can be regarded as beneficial or not. Therefore, the final goal to be achieved by our future work is a hybrid algorithm that selectively applies syntactic analysis to certain classes of queries while relying on standard statistical techniques otherwise.

6. Applying Machine Learning to Interface Development

The expansion of the Internet as the de facto global information infrastructure has not only greatly simplified the access to existing information sources but has also motivated the creation of numerous new sources. With this dramatic growth of the Web and the diversity of information it offers it becomes increasingly difficult to use navigation-oriented browsers or keyword searching techniques to find, extract, and aggregate relevant information. More often, users are confronted with a large, heterogeneous, and constantly evolving network.

Database interfaces usually offer a model of interaction that is very different by accepting a declarative statement of a query as input and providing the results for that query in a fully-fledged way. Different techniques of building indexes, extending HTML language, adopting meta representation mechanisms, encapsulating resources into objects, and establishing host query servers are advocated to integrate database systems into the Web. Still, the Web is not a database in the sense that it does not provide the user with this uniform interface that releases him from concerns about how data is stored and about the syntactic and semantic differences among Web sites.

As an example, many companies are now making their product databases available online with a multitude of user interfaces; some provide key word search, others have their databases nicely categorized, while again others have more advanced searching capabilities. However, a global search for online products and the comparative analysis of their features and attributes is usually still impeded by the semantic differences among these databases. Over the last few years, some approaches have emerged that specifically try to deal with the problems of semantic heterogeneity of product specifications and to provide the users with a user-friendly interface to browse through different vendors' product specifications and to easily retrieve product information from all over the world.

In the context of a research project at the Software Competence Center Hagenberg we addressed this problem by developing a multilingual natural language interface architecture for e-commerce applications [21]. As we have already mentioned in Section 4 natural language interfaces have a high potential in such environments. However, one big stumbling block on the road to the realization of successful applications is the large cost that has to be invested in the acquisition of the necessary linguistic knowledge. Therefore, we propose to overcome this obstacle by automatically learning the required knowledge using machine learning (see also [17]).

In our architecture we restrict the linguistic analysis of the user input to the lexical level. We replace an elaborate semantic analysis module by a machine learning classifier, which assigns the input to the correct query type. In previous experiments with German training data and German, English, and Japanese test data we could show that the learned knowledge successfully abstracts from language-specific phenomena at the surface level [22]. Regarding the choice of the appropriate machine learning algorithm we performed an extensive comparative evaluation of different supervised learning paradigms [23]. As a result we selected rule-based learning because it performed competitively and offers the big advantage that the learned rules can be easily evaluated, implemented, and presented to the user in a clear and understandable form. This allows a transparent knowledge representation, which can be used to explain decisions of the system to the user.

The natural language input is analyzed by the following modules:

- First, the language of the input is detected.
- The input is transferred to the morphological and lexical analyzer for that language. This module performs the tokenization of the input and transforms it into a *deep form list (DFL)*, which indicates for each token its surface form, category, and semantic deep form.
- Unknown values contained in the input are processed separately by the *unknown value list (UVL)* analyzer to check whether they represent identifiers of existing entities in the database. In such a case we tag the unknown value in the resulting UVL with the entity type, otherwise we indicate the data type.
- DFL and UVL are the input for the machine learning classifier. We map the entries in the two lists to binary features and obtain a ranked list of query types according to the learned classification rules.
- As last step the classifications are used to generate the database queries.

The rule-based machine learning algorithm learns a set of rules from the instances in the training set. A rule is defined as a conjunction of *literals*. If a rule is satisfied, it assigns a class to a new case. For the case of binary features the literals correspond to feature tests with positive or negative *sign*. This means that they verify whether a new case possesses a certain feature (for positive tests) or not (for negative tests). The algorithm we developed is based on FOIL [9], however, we introduce a new weighing scheme, which proved to be superior for the application in natural language interfaces, i.e. for learning problems with a large number of classes and binary features [18].

We learn for each class a set of rules by applying a separate-and-conquer strategy. The instances for a certain class represent the *target relation*. The algorithm iteratively learns a rule and removes those instances from the target relation that are covered by the rule. This is repeated until no instances are left in the target relation. A rule is grown by repeated specialization, adding literals until the rule does no longer cover any instances of other classes.

In other words, we try to find rules that possess some *positive bindings*, i.e. instances that belong to the target relation, but no *negative bindings* for instances of other classes. Therefore, the reason for adding a literal is to increase the relative proportion of positive bindings.

For the selection of the next literal we use the following weighing function:

$$W_{f,s,C} = b_f^+ \cdot \left(b^- - b_f^-\right) \cdot w_{f,s,C} \tag{6}$$

In this formula, *s* indicates the sign of the feature test. The number of positive (negative) bindings after adding the literal for the test of feature *f* is written as $b_f^+\left(b_f^-\right)$. Finally, b^- indicates the number of negative bindings before adding the literal so that $b^- - b_f^-$ calculates the reduction of negative bindings achieved by adding the literal. The weights $w_{f,s,C}$ are calculated as class-dependent weights for class *C* by making use of feature weights w_f:

$$w_{f,s,C} = \begin{cases} w_f \cdot \left|D_{C,f}\right| & \text{if } s \text{ is positive} \\ w_f \cdot \left[1 - \left|D_{C,f}\right|\right] & \text{otherwise} \end{cases} \tag{7}$$

$\left|D_{C,f}\right|$ denotes the proportion of instances that possess the feature *f* and belong to class *C* to the total number of training cases for *C*. Finally, w_f is calculated as:

$$w_f = \frac{1}{c}\sum_{j=1}^{c}\left[1 - 4\left|D_{f,j}\right| \cdot \left(1 - \left|D_{f,j}\right|\right)\right] \tag{8}$$

The term under the summation symbol represents the selectivity of feature *f* for class *j*. It equals 1 if either all or none of the instances possess this feature, because this makes it a very discriminating characteristic. The opposite extreme is that $\left|D_{f,j}\right|$ equals 50% because then the feature possesses no information for the prediction of the class and the term under the summation symbol becomes 0.

For a case study with 100 classes (query types) and a data set of 1000 queries containing 317 features we achieved 91.8% correctly classified queries for 10-fold cross validation (2.35% standard deviation). This was a statistically significant improvement over FOIL, which only reached 85.2% (3.46% standard deviation).

For the same case study we had developed a sophisticated semantic analysis component by hand as part of previous research work [19]. For the development of the underlying rule base we had spent several person-months. The reason for this extensive effort was due to the complexity of the classification task because the query types were often very similar and difficult to distinguish even for human experts. All this effort could be replaced by our machine learning component, which successfully learned a compact representation of the problem space with sufficient accuracy. We could show that machine learning really represents a promising alternative to the notorious “knowledge acquisition bottleneck” in natural language interface development. Furthermore, the learned knowledge focuses on the meaning of the query and abstracts from language-specific details at the surface level. Therefore, this kind of approach is especially beneficial for the use in multilingual environments.

7. Conclusion

In this paper we have given an overview of important recent developments in human language technology. We know that this selection of research efforts represents only a small fraction of the large amount of various research issues. However, it is our hope that the paper is still able to give the reader an idea of the huge impact human language technology can have on all our lives. In our research work we have tried to find answers to some of the most relevant research problems. However, we are aware of the fact that at the same time we may have raised even more additional challenges for the future.

Acknowledgements

I would like to take this opportunity to thank Prof. A Min Tjoa for his great support and encouragement throughout many years. I still remember the time when I started as PhD student of Prof. Tjoa in 1990. I wanted to dedicate my future research on human language technology. However, Prof. Tjoa was not so delighted by this idea and wanted me to consider another research topic. The reason was that human language technology was in a long phase of stagnation during this time. He warned me that it will be a long and stony way if I wanted to pursuit this research topic. He should be proved correct. However, my mind was set and after Prof. Tjoa realized that, he put all his efforts in lightening my burden and helping me along the way. In many fruitful discussions I had to defend and rethink my research work. Even after successfully finishing my PhD thesis Prof. Tjoa has always been a constant source of inspiration. I thank him deeply for his guidance, assistance, and friendship.

References

[1] ANDROUTSOPOULOS, I., G. D. RITCHIE, and P. THANISCH, Natural Language Interfaces to Databases – an Introduction, Journal of Natural Language Engineering 1(1), 1995.

[2] BAEZA-YATES, R. and B. RIBEIRO-NETO, Modern Information Retrieval, Addison-Wesley, Boston, 1999.

[3] BERGER, H. et al, Providing Multilingual Natural Language Access to Tourism Information, in: Proc. of the 3rd International Conference on Information Integration and Web-based Applications & Services, 2001.

[4] CAVNAR, W. B. and J. M. TRENKLE, N-gram-based Text Categorization, in: Proc. of the 3rd International Symposium on Document Analysis and Information Retrieval, 1994.

[5] FENSEL, D., Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce, Springer-Verlag, Berlin, 2001.

[6] LEINFELLNER, E., Semantische Netze und Textzusammenhang, Verlag Peter Lang, Frankfurt/M., 1992.

[7] MITTENDORFER, M. and W. WINIWARTER, Exploiting Syntactic Analysis of Queries for Information Retrieval, Data & Knowledge Engineering 42(3), 2002.

[8] MITTENDORFER, M., G. NIKLFELD, and W. WINIWARTER, Making the VoiceWeb Smarter – Integrating Intelligent Component Technologies and VoiceXML, in: Proc. of the 2nd International Conference on Web Information Systems Engineering, 2002.

[9] QUINLAN, J. R. and R. M. CAMERON-JONES, Induction of Logic Programs: FOIL and Related Systems, New Generation Computing 13, 1995.

[10] SLEATOR, D. and D. TEMPERLEY, Parsing English with a Link Grammar, in: Proc. of the 3rd International Workshop on Parsing Technologies, 1993.

[11] SMEATON, A., Using NLP or NLP Resources for Information Retrieval Tasks, in: T. Strzalkowski (ed.), Natural Language Information Retrieval, Kluwer, Dordrecht, 1999.

[12] SOWA, J. F., Ontology, Metadata and Semiotics, in: B. Ganter and G. W. Mineau (eds.), Conceptual Structures: Logical, Linguistic, and Computational Issues, Lecture Notes in Artificial Intelligence 1867, Springer-Verlag, Berlin, 2000.

[13] STRZALKOWSKI, T. et al, Evaluating Natural Language Processing Techniques in Information Retrieval, in: T. Strzalkowski (ed.), Natural Language Information Retrieval, Kluwer, Dordrecht, 1999.

[14] TEMPERLEY, D., D. SLEATOR, and J. LAFFERTY, Carnegie Mellon Link Grammar Web Site, http://www.link.cs.cmu.edu/link/.

[15] URRO, R. and W. WINIWARTER, Specifying Ontologies: Linguistic Aspects in Problem-driven Knowledge Engineering, in: Proc. of the 2nd International Conference on Web Information Systems Engineering, 2002.

[16] VOORHEES, E. M., Natural Language Processing and Information Retrieval, in: M. T. Pazienza (ed.), Information Extraction, Lecture Notes in Artificial Intelligence 1714, Springer-Verlag, Berlin, 1999.

[17] WERMTER S., E. RILOFF, and G. SCHELER (eds.), Connectionist, Statistical, and Symbolic Approaches to Learning for Natural Language Processing, Lecture Notes in Artificial Intelligence 1040, Springer-Verlag, Berlin, 1996.

[18] WINIWARTER, W., ILE – an Integrated Learning Environment for Knowledge Discovery from Databases, in: Proc. of the Australasian Database Conference, 1998.

[19] WINIWARTER, W., The Integrated Deductive Approach to Natural Language Interfaces, PhD thesis, University of Vienna, 1994.

[20] WINIWARTER, W. (ed.), Sprachtechnologie, ÖGAI Journal 20(1), 2001.

[21] WINIWARTER, W. and I. K. IBRAHIM, A Multilingual Natural Language Interface for E-Commerce Applications, Proc. of the 13th International Conference on Applications of Prolog, 2000.

[22] WINIWARTER, W. and Y. KAMBAYASHI, A Comparative Study of the Application of Different Learning Techniques to Natural Language Interfaces, in: Proc. of the Workshop on Computational Natural Language Learning, 1997.

[23] WINIWARTER, W. and Y. KAMBAYASHI, A Machine Learning Workbench in a DOOD Framework, in: Proc. of the 8th International Conference on Database and Expert Systems Applications, 1997.

[24] W3C, Multimodal Requirements for Voice Markup Languages, W3C Working Draft 10 July 2000, http://www.w3.org/TR/2000/WD-multimodal-reqs-20000710/, 2000.

[25] W3C, Voice eXtensible Markup Language (VoiceXML) version 1.0, W3C Note 05 May 2000, http://www.w3.org/TR/2000/NOTE-voicexml-20000505/, 2000.

[26] W3C, Voice Extensible Markup Language (VoiceXML) version 2.0, W3C Working Draft 24 April 2002, http://www.w3.org/TR/2002/WD-voicexml20-20020424/, 2002.