

Robust Interpretation of User Requests for Text Retrieval in a Multimodal Environment

Alexandra Klein and **Estela Puig-Waldmüller**

Austrian Research Institute for
Artificial Intelligence
Schottengasse 3
A-1010 Vienna, Austria
{alex, stella}@oefai.at

Harald Trost

Department of Medical Cybernetics and
Artificial Intelligence, University of Vienna
Freyung 6/2
A-1010 Vienna, Austria
harald@ai.univie.ac.at

Abstract

We describe a parser for robust and flexible interpretation of user utterances in a multi-modal system for web search in newspaper databases. Users can speak or type, and they can navigate and follow links using mouse click. Language queries may combine search expressions with browser commands and search space restrictions. In interpreting input queries, the system has to be fault-tolerant to account for spontaneous speech phenomena as well as typing or speech recognition errors which often distort the meaning of the utterance and are difficult to detect and correct. We present a parser integrating shallow parsing techniques with knowledge-based text retrieval to allow for robust processing and coordination of input modes. Parsing consists of two layers: typical meta-expressions like those for search, newspaper types and dates are identified and excluded from the search string to be sent to the search engine. The search terms which are left after preprocessing are then grouped according to co-occurrence statistics which have been derived from a newspaper corpus. These co-occurrence statistics consist of typical noun phrases as they appear in newspaper texts.

1 Introduction

In this paper we describe a parser for robust and flexible interpretation of user utterances in a web-based multi-modal text retrieving system. The parser forms part of a system for web search in Austrian newspaper databases. In this system users can formulate queries or navigation commands using utterances in both spontaneous spoken or written language, and they can navigate and follow links using mouse click. Users are completely free in formulating their utterances and in their use and combination of the input modes. Typed and spoken utterances may combine query expressions with browser commands and search space restrictions. Users may

search for texts with a specific date or in a specific newspaper or in a specific section of a newspaper. They may give complex context descriptions of the texts and they may refer to texts which were found previously. A dialogue manager stores actions and results from previous states and supplements information to construct fully specified formal queries from user requests.

In order to allow for this freedom in user behaviour, flexible processing modules are needed. For every utterance, the parser (and, finally, the dialogue manager) must come up with an adequate interpretation. At the same time, in interpreting the input, it has to be robust and fault-tolerant. It has to cope with typical phenomena of spontaneous speech like hesitation, correction and repetition. There may be typos in written input or - even more difficult to deal with - speech recognition errors from the spoken queries. Such errors often distort the meaning of the utterance and are difficult to detect and correct.

In our interpretation component shallow parsing techniques and knowledge-based text retrieval methods are combined to allow for robust processing and coordination of input modes. We employ a two-layered approach. The first layer serves to separate structure from content, i.e., parts of utterances referring to browser commands and search restrictions (temporal expressions, newspaper types or sections) are analyzed with a combination of keyword spotting and pattern recognition. The underlying assumption is that users will restrict themselves to a rather small vocabulary and a limited range of expressions in expressing this sort of information. This assumption is also confirmed by our Wizard-of-Oz experiments. During this process, stop words (function words and other words typically not contributing to the content of the query) are also removed. The remaining words –which are taken to describe the search content– are then grouped ac-

coding to co-occurrence statistics which have been derived from a newspaper corpus. While text retrieval with the help of linguistic processing has become fairly common, multimodal interaction with textual databases on the web is a fairly recent application of Natural Language Processing. Experience from text retrieval shows that most information is expressed in adjective-noun, noun-preposition-noun, and noun-verb groups (see, e.g. ?)). In our specific domain the third type can be neglected, because verbs typically denote the action –mostly search– which is already extracted in the first layer. Thus, co-occurrence statistics consist of typical noun phrases as they appear in newspaper texts.

2 Empirical evidence and user experiments

In order to assess user behaviour, we carried out Wizard-of-Oz experiments ?). Speech recognition and text retrieval were simulated. In different sessions the users interacted with a number of versions of the system: single input mode versions and versions with combinations of input modes. Their performance in terms of number of interactions as well as task completion time was measured, and their comments regarding the interface and the (simulated) system were collected in a questionnaire. Users were grouped according to previous experience with search engines on the web in general. Our results show that both, beginners and advanced users, preferred multimodal interaction over single input modes, and beginners in particular were able to speed up task completion times significantly with the help of a combination of spoken and written input with mouse clicks ?).

From these experiments, we also obtained a corpus of written and spoken utterances which were considered in the further design of the system. The queries which were posed by the users in spoken language were recorded. The recorded utterances were later read to a speech recognition system. This gave us an impression of the number and type of errors to be expected in dealing with queries in spontaneous speech.

3 NL Text or Speech Input: Language Analysis

Users can access articles with spoken or typed utterances which can be formulated by natural language. Such web queries may relate to the way some particular piece of information is presented and connected,

and to what this information refers to. They may also express browser commands or a combination of browser and query commands while referring either to structure (“Search for Noll in the previous newspaper”) or to content (“Search for Noll in the sports’ section”). Within our web queries may relate to the way some particular piece of information is presented and connected (e.g. the browser’s history about the accessed pages), and to what this information refers to (e.g. the section a search string belongs to). To successfully interpret such an utterance one needs to analyze its structure to find out which of these command modes the utterance can be assigned to. This is done in a two-step process. First, each word is looked up in a lexicon and assigned a semantic category. Then, certain rules are applied to strings of these semantic categories. As a result, commands and search restrictions are recognized and the rest of the utterance is passed to search expression interpretation.

3.1 Keyword Spotting and Lexicon Look-Up

We will now describe in more detail how the user’s input is parsed within the Natural Language Interface, and structured into either search patterns - consisting of search strings, sections, dates and timeranges, that are understood by the query engine of the newspaper - or Java browser commands. Structure is analyzed by a flexible bottom-up parser using a rule-based mechanism with simple syntactic patterns.

After reading in the user’s query input, each word of the utterance is looked up in a lexicon and - if found - assigned a corresponding semantic category. This lexicon contains a small list of semantic categories, that we consider important for the interpretation of an utterance in the domain of searching articles and browsing. The lexicon assigns semantic classes for closed categories that are:

- nouns denoting search, newspaper, section, links like search or article.
- nouns expressing a specific section like economy.
- nouns expressing a specific page like home.
- temporal expressions and temporal prepositions like month and ago.
- expressions indicating something new like in a new search.
- adjectives and adverbs indicating direction in time or space, like in previous search or previous week.

- cardinal and ordinal numbers used in conjunction with temporal expressions and link expressions, like in 2 years ago or open the first link.
- adverbs and connectives indicating constraints on search mode, like only and not.
- prepositions indicating whether the request was to browse or to search, cf *zum Sport* (to the sports' section) versus *im Sport Sport* (within sports' section).
- stop words.

All words found within the lexicon, are replaced by their semantic class, search expressions are marked as such, the stop words are deleted.

We distinguish between semantic atoms and semantic classes: atoms by itself do not have a meaning that can be used for searching or browsing commands. They have to be joined following a given set of rules to form a semantic class. To yield such a class rules are applied in -mostly- one to three steps. However, rules are not necessary in any case, a word can be translated as a semantic class right from the beginning also. Our lexicon has about 30 semantic atoms, from which about 40 semantic classes can be formed.

3.2 Pattern Matching and Semantic Classification

Certain patterns of semantic classes which we obtain through lexical look-up, can be assigned a new meaning via rules. So, by composing the individual meanings another more abstract meaning is defined. The result of this process is a list of chunks¹, where stress is laid on the content words. The advantage of concentrating on chunks is - especially within German, a language with a relative free word order, that the order in which chunks occur is much more flexible than the order of words within chunks. This approach might be too shallow for the analysis of deeper semantics, but is sufficient for our needs. So, e.g. "last" and "week" and its semantic mappings 'previous' and 'week' would together yield the new meaning 'date_1w'. To overcome ambiguities and avoid potential rule conflicts, rules with a greater length have a higher priority and are thus preferred, such that *zurück zum Sport* (*back to sports*) would

¹According to (?) a chunk is defined in terms of major heads where a major head is any content word that does not appear between a function word *f* and the content word *f* selects, OR a pronoun selected by a preposition. Further The typical chunk consists of a single content word surrounded by a constellation of function words, matching a fixed template.

win over *zurück* (*back*). If no rules can be applied to a semantic class it will be ignored within the final interpretation. We distinguished between the following semantic classes:

Summing up the process of the structure analysis, the partial analyses are stored, a sequence of partial analyses from the set of rules is chosen, and then combined to yield bigger structures.

3.3 Search String Filter: Extraction of Adjective-Noun Pairs

After having extracted structure analysis the content of the query must be analyzed in more detail, whereas search strings with low co-occurrences are filtered out. Chapter 4 will explain how content analysis is done in our application.

From a corpus of Austrian newspaper texts, adjective-noun and adjective proper name pairs were extracted and counted. These pairs were stored and consulted in query interpretation. Since the texts are tagged manually, the lists of adjectives and nouns/proper names contain a considerable number of errors. Therefore it is necessary to use large amounts of text; it may even be useful to eventually introduce a threshold so that only adjective-noun/proper name pairs which appear more than once or a certain number of times are considered. This of course can not prevent systematic tagging errors.

A robust stemming algorithm maps all adjective-noun/proper name pairs to an approximate 'stem', thus eliminating flexional forms which result in morphological variation which is typical for the German language. For the purpose of creating a repository of co-occurrence pairs, we do not care about proper stemming. Rather, it is our aim to map various inflectional forms onto one base form.

Spelling variations, numbers etc. are smoothed as far as it is possible in automatic processing. For example, ordinal numbers which are labelled as adjectives are reduced to a placeholder for numbers.

Whenever a word is encountered in processing which can be considered an adjective, it is kept. Whenever the following word may be a noun or a proper name, it is checked whether the adjective-noun/proper name combination is contained in the repository of adjective-noun/proper name combinations which has previously been extracted from a corpus. If the adjective-noun/proper name combination is found, it is passed on to the search engine as a query. Whenever the combination has not occurred in the corpus, only the noun or proper name

is considered a key word.

Again, inflectional variations as well as different spellings etc. are mapped onto base forms as far as possible. The same stemming algorithm is used which was employed in creating the repository of adjective-noun/proper name pairs. The robust (and rough) stemming and categorization algorithms produce a certain amount of mistakes in the lists of pairs as well as in the mapping process, but taking into account larger text corpora evens out these problems the more text is processed.

Our approach distinguishes noun phrases which have a record of co-occurrence from noun phrases which may be spontaneous expressions or modifications or even errors created by users. For example, the phrase *europäische Staaten* would be retained while *beteiligte Staaten* would be reduced to the noun. Some adjective used in search expressions serve to qualify the global search expression rather than the noun or proper name in question. For example, a search for *yesterday's speech* would only yield articles about this particular speech.

4 Merging Chunks: Translation into one unambiguous Command

After pattern matching and rule application we get the meaning of the user's utterance as the sum of the lengths of the component analyses. The best sequence of analysis is then merged and delivered to the control module, that is able to include its knowledge about previous searches. The language analysis module translates its parsing results into one unambiguous command line. This command consists of a fixed set of parameters, which are given a certain value: DIRECTION, SECTION, SEARCHSTRING, TIME, ZEITUNG, CONTEXT, PAGE, SUBMIT and LINKNUMBER. The outcome - or left-hand side - of a rule-based simplification can be divided into three command types (some examples are given below):

- **Search Command:** parameters DIRECTION, SECTION, SEARCHSTRING, TIME, ZEITUNG, CONTEXT, PAGE may be filled, SUBMIT must be filled. LINKNUMBER must be empty.
 - Normal Search: e.g. "Search for Camilleri"
 - New Search
 - Search using the Search History: e.g. "Search for Camilleri in the previous section"
- **Navigation Commands:** parameters DIRECTION, SECTION, SEARCHSTRING, TIME,

ZEITUNG, CONTEXT, PAGE may be filled, SUBMIT may not be filled. LINKNUMBER must be empty.

- Normal Browsing using the Accessed Page History: e.g. "Go to the next page"
- Browse using the Search History: e.g. "Go to the last ressort", "Go back the search containing Montalbano"
- **Opening Link Command:** the parameter LINKNUMBER must be filled, the other actions must be empty.
 - Open Link: e.g. "Open the first article"

The action name browsing refers to the timeline and the point of reference of a browsing but also of a search command. E.g. take an utterance, where someone wants to search for a topic but within a context, that was defined in the previous search. For our application we would first have to locate the user's point of reference and then execute her search command. If there is no given reference, we assume by default that a new time point is created in our time line.

One such command could look like this: the utterance "I am looking for something about Highsmith within the previous section" would be mapped to "DIRECTION previous, SECTION T, SEARCHSTRING Steuerhinterziehung, TIME nil, ZEITUNG nil, CONTEXT c, PAGE nil, SUBMIT nil, LINKNUMBER nil". The controller that receives the command line has to process the line following some rules.

5 Controller: Including Interactive Memory Information

The results of these rule-based transformations have to be handed over to the control module. These include either commands understood by the Java browser (*go back, forward, new search*) or search patterns involving the search string(s), the date or time range of the issued article, and the preferred section. The control module compares these results with the searching and browsing history. The browser's history includes information about all accessed pages, whether the user has accessed them by spoken, typed or mouse-click commands. Those commands, that either are incomplete or impossible in the given context, are ignored and rejected. A powerful interaction control is necessary in order to recognize the user's intent by comparing it to what the system knows about the addressed entities and

their relation to each other as well as to the data which are accessible at the specific moment in the interaction.

6 Result: Translating into Http Request or Browser Command

After the command has been passed by the control module, it is either executed by the Java browser or translated into a GET method through an Http request to the newspaper's archive database. The resulting articles are displayed in the Java browser, another search can be started by the user.

7 Conclusion

We have presented an interpretation component for natural language user input in a web-based multimodal text retrieval system. By applying well-known and simple methods from shallow parsing and knowledge-based text retrieval and integrating them in a novel way we have succeeded in creating a robust, flexible and efficient parser for our application.

Crucial for our application is the distinction between those parts of utterances relating to structure and those relating to content. This is achieved by taking advantage of the fact that only a limited vocabulary and set of expressions are used for the former. This allows us to employ simple rule-based techniques for their interpretation. The identification of the content on the other hand is done with the help of a co-occurrence repository, at the moment consisting of adjective-noun/proper name pairs. In the future we will have to investigate whether search results can be improved by inserting other combinations, like noun-preposition-noun triples.

Acknowledgements

This work was supported by the Austrian Science Fund (FWF) under project number P-13704. Financial support for ÖFAI is provided by the Austrian Federal Ministry of Education, Science and Culture.