From: AAAI Technical Report SS-98-01. Compilation copyright © 1998, AAAI (www.aaai.org). All rights reserved.

Using Unsupervised Learning For Engineering of Spoken Dialogues

Jens-Uwe Möller

Natural Language Systems Division,
Dept. of Computer Science, Univ. of Hamburg
Vogt-Koelln-Str. 30, D-22527 Hamburg, Germany
Phone: ++49 40 5494 - 2516 / Fax: ++49 40 5494 - 2515
http://nats-www.informatik.uni-hamburg.de/~jum/
mailto:jum@informatik.uni-hamburg.de

Abstract

Major steps towards dialogue models is to know about the basic units that are used to construct a dialogue model and possible sequences. In this approach a set of dialogue acts is not predefined by any theory or manually described during the engineering process, but is learned integrating different kind of data that are available in an avised spoken dialogue system. For this purpose an existing unsupervised learning algorithm was adopted and extended to the demands in natural language processing and speech processing, called CLASSITALL. It is embedded in a tool kit for dialogue engineering, defining basic modules. The DIAlogue MOdel Learning Environment supports an engineering-oriented approach towards dialogue modelling for a spoken-language interface. The architecture is outlined and the approach is applied to the domain of appointment scheduling.

Introduction

To build dialogue models for spoken dialogue systems based on human-to-human dialogues is a tremendous work: data acquisition (recording and transcribing), analysis (according to some dialogue structuring theory) and the development of recognition procedures for dialogue structures. The analysis of some data, like prosody are far beyond the effort for an industrial application.

To make good dialogue models affordable for many applications it is important to reduce the construction effort. Therefore, dialogue modelling tools like CSLUrp (Novick and Sutton 1996) neglect the variety of phenomena in spoken language and build up dialogue from a set of restricted slot-filling dialogues. Others apply a supervised learning algorithm using some dialogue structuring theory with a given set of dialogue acts (Litman 1994, Siegel and McKeown 1994, Lehnert and Soderland 1994, Mast, et al. 1996, Wermter and Löchel 1996).

In contrast to other learning approaches to dialog modelling the tool kit DIA-MOLE does not employ theory-based dialogue units because they are subject to human interpretation and often cannot be recognized from data available in a spoken-language system. A similar approach adopting unsupervised learning for dialogue acts relies on human-labeled tags (Andernach 1996). We pursue a data-driven approach and apply unsupervised learning to a sample set of spontaneous dialogues using multiple knowledge sources, i.e. domain and task knowledge, word recognition, syntax, semantics and prosodic information.

Given these data, DIA-MOLE supports segmentation of turns and interpretation of their illocutionary force based on a model of the task. As a result of learning we obtain domain- and task-specific dialogue acts (DDA) with associated features. Validations of the set of learned DDAs has shown that they are prominent for this domain and task. Dialogue act prediction was employed to evaluate our approach.

Predictions may be used by again other modules of the spoken dialogue system to adapt their environment. E.g. the word recognizer may use dialogue act predictions to choose a specific language model trained on these DDA classes to improve word recognition. Furthermore, a dialogue-planner module can use predicted DDA with their associated features as prototypes for generation to display a very natural behaviour in dialogue. A dialogue planner based on DDA predictions and case data is actually under development.

The architecture of DIA-MOLE also allows self-adapting dialogue models. If DIA-MOLE is integrated in a spoken dialogue system which is in practical use, all occuring

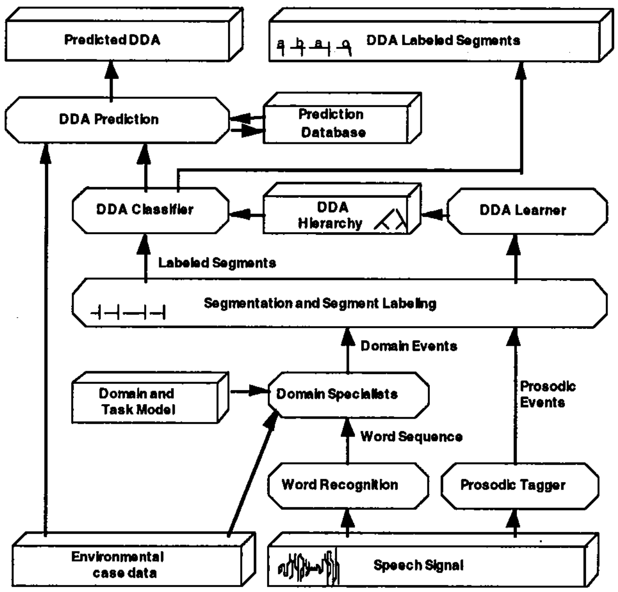

Figure 1: Interpretation in DIA-MOLE

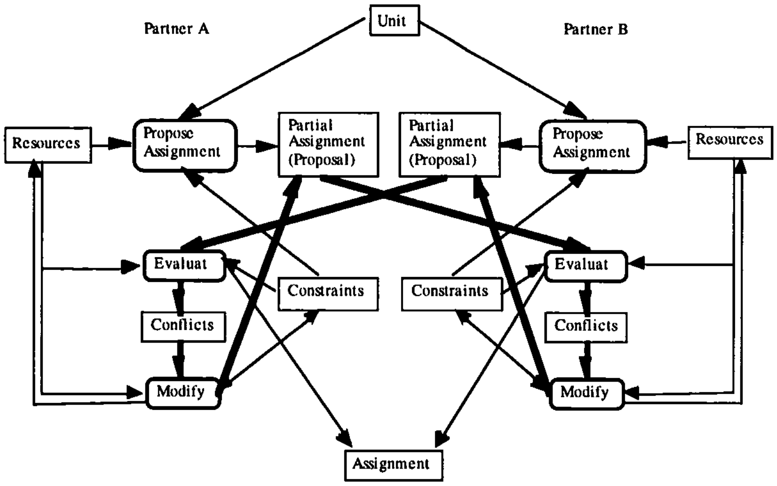

Figure 2: Functional structure of interactive appointment scheduling

dialogue turns, or more precisely, segments may be presented to both, the DDA classifier for further processing of that turn and to the DDA learner to improve the model in its application situation. For this reason we adopted an incremental learning algorithm within the DDA learning module.

DIA-MOLE Toolkit

Word recognizer (Hübener, Jost and Heine 1996) and prosodic tagger (Strom 1995) are modules that were developed independent of this work. Domain and task model allow us to interpret the intention of the communcative agent's utterances. DIA-MOLE does not prescribe any modelling technique for the domain and task, but we made good experience with the KADS method (Breuker and Wielinga 1986 , Sundin 1994).

Domain specialists carry out syntactic and semantic analysis according to the domain structure and the underlying task model. Spoken language does usually not consist of well-formed sentences (Weber and Wermter 1995, Eckert 1996), thus syntactic parsing in DIA-MOLE should be restricted to partial parsing. Again DIA-MOLE does not determine any parsing technique, but as a result, domain specialists provide domain-relevant events.

The segmentation module breaks turns into segments which are labeled with features from prosody (accent, phrase boundaries, senctence modality and focus (Elsner and Klein 1996)) and domain events. These labeled segments (see for example Figure 3) are passed either to the learning module that implements the CLASSITALL algorithm resulting in a classification hierarchy of DDAs or they can be passed to the classification module.

The resulting DDA hierarchy can be used to classify and automatically label segments either integrated into a dialogue system or in batch mode. From automatically labeled dialogues, the prediction module either learns about predicting a subsequent act in a dialogue or just predicts it. Predictions are based on a ngram-model ($n<=3$).

Typical events of a DDA class can be used in NL generation and speech synthesis for sentence planning, realisation and prosody.

All modules developed within the DIA-MOLE can be configured and assembled giving a spoken dialogue system. DIA-MOLE actually runs in SICStus- and Quintus-Prolog on different platforms. Integration with the word recognizer and prosody recognizer only runs on Solaris.

Application Domain

The domain of appointment scheduling was chosen as an application domain, because speech data and a whole speech system environment for this domain were available at our department within the VERBMOBIL project. The correctness of the word recognition that is used in DIA-MOLE is about 80%.

Domain Modelling

For the domain of appointment scheduling we first did an analysis of the problem and a typical problem solving method using the modelling method KADS. Starting out from the problem solving method *propose-and-revise* we developed, as a first step for the application of DIA-MOLE to this domain, a functional structure of interactive appointment scheduling. Figure 2 shows the structure with a main cycle of *propose-evaluate-modify* switching between both dialogue partners. Rectangular boxes indicate data and rounded boxes stand for processes. Though they are differenciated in a problem solving model both, data and processes may be verbalised in a natural language dialogue. A unit is one appointment to schedule, and resources are the calendars of the dialogue partners.

Domain Events and Prosodic Events

As a second step domain specialists recognising contributions to one of the data or process were developed. A proposal in this domain may consists of a time interval and a location, but usually they are underspecified. Therefore, the domain specialist for date and time expressions is coupled with a specific context model yielding significantly better results. While only 25.6% of the date and time expressions without context were non-ambiguous, with the help of context the right interpretation could be found for 84.5%.

As a representation for lexical semantics we use Conceptual Graphs (Sowa 1984). Entries to lexical semantics are automatically derived from the WordNet ontology. A filter and re-interpretation module was applied to data from the prosodic tagger. Domain events and prosodic events are given to the segmentation module.

```
attitude (POSITIVE | NEGATIVE)
location (LOCAL | GLOBAL)
conflict (CONFLICT)
date-and-time-interval (SAME | NEW |
    ALTERNATIVE)
date-and-time-specificity (SPECIFY | GENERALIZE
    | SAME)
assignment(ASSIGNMENT)
phonMod (QUERY | ASSERTION | CONTINUATION)
turn(EXIT)
```

Figure 3: List of segment features that were used in the domain of appointment scheduling

Segmentation

Rules on segmentation of turns are based on prosodic information and domain knowledge. The major segmentation rule inserts a boundary just behind an prosodic event, that follows domain events. If there are no prosodic events, this results in bigger segments comprising multiple dialogue acts. Compared to a manual segmentation based on RST theory our rules show a precision of more than 95% for the determination of segment boundaries. Connectives are the major source of errors: At least in our corpus, prosody suggests that connectives are placed at the end of a first segment, already indicating that another segment will come. This reflects human utterance planning process. Human segmentations are probably influenced by German syntax and place a connective at the beginning of the second segment. We did not write special rules to circumvent this effect, as it does not influence further processing.

Segment Labeling

In a fourth step information from prosody and domain specialist are assigned as features to segments. Figure 3 shows the set of features and values used in this application. Context information is used to pursue moves in the domain. Probability values stemming from the underlying processes are also added with the features yielding case descriptions for the learning algorithm.

Learned DDA-Classes

In contrast to manually labeled dialogue acts (Jekat, et al. 1995), learned DDA classes distinguish for example explicit and implicit rejection. This means that they do not characterize the illocutionary force by interpretation, but their illocutionary force relies on acts in the domain and its task model.

The DDA-classes in Figure 4 are based on a set of 187 spontaneously spoken dialogues from the VERBMOBIL corpus with 4521 turns. A * marks, that another segment from the same speaker will follow. It is interesting to find conversational phenomena reflected in the classes, e.g. that disagreement very seldomly stands for its own, while agreement can do this in a dialogue. The actual set does not consider general dialogue information, e.g. on greetings which will be added in a future version. Furthermore pruning within the algorithm favors frequent segment features thus suppressing clear and distinct smaller classes. This could be circumvented by weighting features.

- Suggesting a new time interval to consider for planning
- Emphasizing an alternative time suggestion *
- Alternative time suggestion

- Acknowledgement and more specific time suggestion
- Acknowledgement and more specific time suggestion *
- Emphasizing an option for a more specific time interval (as the only one)
- Asserted more specific time interval *
- More specific time intervals

- Acknowledgement *
- Emphasized acknowledgement *
- Emphasized disagreement *

- Positive evaluated alternative time interval usually indicating a new scheduling approach
- Evaluated and emphasized alternative time interval

- Emphasized conflict
- Conflict *
- Simply mentioning a conflict

- Disagreement *
- Suggestions of locations

- Acknowledgement and assertion of a conflict
- Demand for scheduling an appointment
- A first time suggestion

- Acknowledgement

- Turns without any domain contribution

Figure 4: Learned DDA classes

Prediction

We also evaluated the quality of the learned DDAs by testing whether they are well suited for dialogue act prediction. For this reason we compared our approach with prediction rates on manually labeled dialogue acts reported by Reithinger and Maier (Reithinger and Maier 1995). They report a hit rate for the first prediction of 29% and 45% with the first and second together when considering every turn in the data. For learning we used again 187 dialogues with 4521 turns. Prediction rates based on an unseen test set of 79 dialogues with 1495 turns are given in Figure 5. Though bothered with word recognition errors, prediction rates are compareable to human labeled dialogue acts. The hierarchical representation allows an abstraction of dialogue acts. Abstracting just one level results in better predictions than with human labeled dialogue acts. We assume that further experiments with weighting features will result in better hit rates.

Generating System Utterances Using DDA Classes

One problem of unsupervised learning of DDAs is that their meaning is not directly accessable to humans, since the category characteristic is described as a set of feature-value-pairs which additionally result from a maximal difference to all other categories. Although DDAs are not subject to human interpretation they can directly fulfill a useful task in a dialogue system, notably in a dialogue planner. If domain constraints demand for some specific feature values that have to be communicated to the dialogue partner, e.g. *attitude negative*, the most probable predicted DDAs can be used to plan further content, e.g. *date-and-time-interval alternative* and finally utter a very natural dialogue contribution. Notice, that DDAs do not only contain information about the propositional content, but also about prosodic features which can be used for spoken language generation. This is an important contribution of DDAs since the integration and manual editing of prosodic rules into language generation still causes tremendeous problems.

CLASSITALL - an Integrated Learning Algorithm for Different Kind of Data

For the classification of utterance segments into pragmatical oriented dialogue act classes we use CLASSITALL (Möller 1997). It is an incremental, polythetic and unsupervised learning algorithm based on COBWEB (Fisher 1987) or its decendent CLASSIT (Gennari, Langley and Fisher 1989), respectively. CLASSITALL integrates numeric and symbolic values and adds features for dealing with uncertain and incomplete knowledge. The latest version is even able to process Conceptual Graphs as structured values within the same framework.

CLASSITALL incrementally builds up a classification hierarchy from a set of cases. A *case* description is a data set consisting of a set of attribute-value pairs. *Nodes* or *classes* in the classification hierarchy will be called *categories*. For abstracting categories, COBWEB (Fisher 1987) uses a *measure of clustering quality*. From this quality measure, a value called *category utility* is derived by transformation and normalization that allows to determine where to integrate new cases. CLASSIT (Gennari, et al. 1989) replaced descrete values from COBWEB with numeric values using about the same algorithm to compute the hierarchy. This paper relates on those previous works without reciting formulas and algorithms.

	1st Prediction	*1st+2nd Prediction*
23 classes	23.70%	38.34%
abstraction to 9 classes	32.91%	54.51%

Figure 5: Hit rate for dialogue act predictions

Clustering of Symbolic, Numeric, Uncomplete and Uncertain Data

In CLASSITALL we adopt the attribute-value paradigm from COBWEB resp. CLASSIT and integrated numeric and symbolic values. For this purpose, we had to normalize numeric values. The category value for numeric values is then modified to

$$\sum_k P(C_k)\cdot\left(\sum_i \frac{1}{\sigma_{ik}2\sqrt{\Pi}} - \sum_i \frac{1}{\sigma_{ip}2\sqrt{\Pi}}\right)\Big/K$$

Function 1: Normalized category utility for numeric values in CLASSITALL[1]

The expressiveness of the data that could be processed by the learning algorithm was extended towards the needs of dialogue modelling and spoken language. Input of probability values for attribute-value pairs enables us to use probability values stemming from underlying modules in a spoken language system directly for processing within the learning algorithm. They may express the quality of data. Moreover, it is possible to exert influence on the importance of specific attributes and values for the resulting classification hierarchy.

$$P(Ai = Vij) <=> \frac{\sum_c (A_{ci} = V_{ij})Q_{ij}}{\sum_c A_{ci}Q_{ix}} \text{, where } Q_{ab}=AQ_{ab}\,CQ_{ab}$$

Function 2: Attribute-value probability for symbolic values in CLASSITALL

Computation of the symbolic and numeric category utility had to be changed to incorporate these extensions. In Function 2 *AQ* is the quality of an attribute having one specific value and *CQ* the quality of a whole case, both given as probability values, c stands for cases. Similar changes had to be made for numeric values. For numeric values mean and standard deviation in CLASSITALL are weighted with the quality Q_i resulting in Functions 3, where $Q_i=AQ_i\,Cq_i$.

$$\mu\omega = \frac{\sum_i V_i Q_i}{\sum_i Q_i} \quad \text{and} \quad \sigma\omega = \sqrt{\frac{\sum_i (\mu\omega - V_i Q_i)^2}{\sum_i Q_i}}$$

Functions 3: Weighted mean and weighted standard deviation for numeric attributes in CLASSITALL

[1] σ_{ik} is the standard deviation of an attribute in a given category, σ_{ip} is the standard deviation for an attribute in its parent category, and K is the number of cases

Clustering Structured Values

CLASSITALL also integrates structured values, represented in the Conceptual Graphs formalism (Sowa 1984). The development of an efficient algorithm for classifying structured values according to the Conceptual Graph formalism enables us to integrate syntactic and semantic structures into the learning of DDAs.

Classification of semantic networks typically means to build up an overall subsumption hierarchy like a taxonomy for KL-One terms (Woods 1991) or a hierarchy of Conceptual Graphs (Levinson 1991, Ellis 1992). In contrast to other approaches to conceptual clustering of structural knowledge (Kietz and Morik 1994, Bournaud and Ganascia 1995, Mineau 1995) that build up a unique classification tree, thus applying monothetic learning, we stick to the COBWEB paradigm of polythetic clustering and allow for multiple complex attribute clustering.

Instead of classifying a graph into a complete graph hierarchy CLASSITALL a) provides with every category a list of set-up points within a virtually existing, but not computed overall graph hierarchy that should be tested, and b) construct a partial generalisation space consisting only of maximally extended common generalisations.

For integration of structured values into CLASSITALL one major problem is to compare them. While symbolic values are easy to compare - either they are equal or not - the similarity of numeric values is harder to determine. Similarity for real-numeric values could be computed by discretizing them into some intervals (c.f. (Lebowitz 1985)). Gennari et al. (Gennari, et al. 1989) chose another approach and stored a continuous normal distribution for each attribute, expressed and computed using mean value μ and standard derivation σ, causing some changes to the original COBWEB algorithm to deal with the numeric category utility, i.e. *cutoff* and *acuity*. Comparison of structured values is even more complicated, because structured values may consist of arbitrarily overlapping graphs. Usually, the degree of overlapping of graphs is regarded as a measure of semantic similarity between graphs. The category utility for clustering of graphs in a hierarchy should be dependent on how much graphs in a category share with each other.

Let us have a look at an example from the domain of appointment scheduling and reduce the semantics of utterances of dialogue partners to a simple description of domain-relevant parts. If we want to cluster these utterances using only the *domain* attribute (Figure 4), we would expect to be case A and B in the same category, while E should be in an other one. The algorithm and internal representation has been described elsewhere (Möller 1997).

Combining and Integrating Different Learning Methods in a Dialogue System

In DIA-MOLE different methods for learning and natural language processing are are used on different levels of processing.

Statistics for Recognition from the Speech Signal

For determining prosodic events a statistical component using a Gaussian distribution classifier over a set of 30 energy features from the speech signal is applied. This is further described in (Strom 1995). Hidden Markov models are used to describe automata for word recognition. The HTK toolset is used to learn and apply these models (Hübener, et al. 1996). Possible word graphs are weighted and pruned by knowledge on probable word sequences given by n-grams.

Both approaches do not return simple symbolic results, but a set of solutions weighted by probability. It seems to be reasonable to consider the quality of the results indicated by their probability with processing on higher level. For a parser in a dialogue system specialised on analyzing relevant parts of an utterance it has turned out that it offers the same performance when it is based on either the best chain or a whole word graph.

Syntactic and Meaning Analysis

A pure symbolic approach is applied in those cases, where grammar rules are known beforehand, or where distinct actions have to be performed. In the domain of appointment scheduling this is e.g. the parsing and interpretation of time expressions and the intertwinned update of the dialogue context. The advantage of this manual engineered approach is that known surface constructs and classes of these can be considered even if they are not present in the available corpus.

Statistical and trainable grammar instead should be used, when there is a great variety of surface realisation that could be well trained from a corpus. This technique has not yet been used within the framework, but seems to be adequate e.g. for the recognition of more abstract constructs like

case	attribute	value
A	domain	`[Appointment]->(Temporal-Locating)->[Time-Interval]->(Start-Time)->[Time-Point].`
B	domain	`[Appointment]->(Temporal-Locating)->[Time-Interval]->(End-Time)->[Time-Point].`
E	domain	`[Appointment]->(Spacial-Locating)->[Location].`

Figure 4: Categories describing a single case

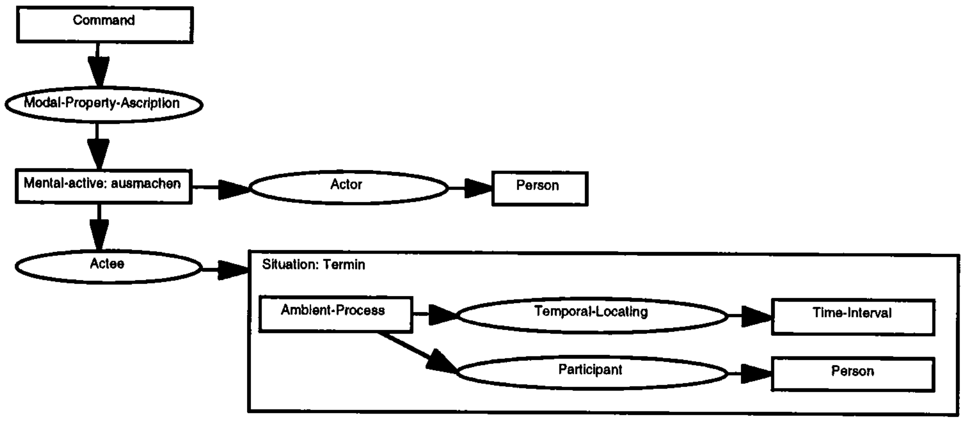

Figure 5: Semantic representation of the sentence *ja prima dann lassen Sie uns doch noch einen Termin ausmachen wann w"are es Ihnen denn recht* (Ok, let us schedule an appointment when would you like)

conflict which on the other hand are involved in less interpretation.

Pragmatics and Dialog Processing using CLASSITALL

For the classification of utterance segments into pragmatical oriented dialogue act classes we use CLASSITALL. Different kind of data are integrated at that stage. Figure 3 shows the set of *symbolic* domain events and prosodic events that are attached as features to segments. Figure 5 gives an example for the semantics of a sentence represented using Conceputal Graphs. Concept and relation names accord to CG-PENMAN (Schirdewan 1996). Figure 6 shows the input to CLASSITALL. In practice, semantics is not derived for whole utterances because we do not perform an overall parsing, but partial parsing. Instead some smaller graphs often just containing a single concept or a relation with two concepts are present in our data. Nevertheless, even clustering of single CG concepts is different from simple symbolic values, as a subsumption hierarchy is used while clustering.

Conclusion

DIA-MOLE learns classes of dialogue acts exclusively from automatically derived data in a spoken dialogue system. Only a domain model and domain specialists have to be developed. This had two positive effects, first, we avoided the enormous effort to label dialogues, and as a consequence of this, second we are able to use very large amounts of data for learning.

It has been show in previous work, that these classes are different from manually labeled dialogue acts, but their properties (recognisability, predictability and use in a dialogue planner) are at least equivalent. The interpretation of DDAs is somewhat artificial as they usually are just a node number with feature probabilities and their primary goal lies in the use for other system components.

The learning algorithm CLASSITALL is introduced that is able to combine different knowledge representations: Symbolic, numeric, structured and uncertain data. By using this incremental learning algorithm it is possible to self-adapt dialogue models in use. A dialogue-planner based on such self-adapting dialogue models is topic of further research.

```
$segment       1.0 n001k-0-0
attitude       symbolic  1.0  POSITIVE
assignment     symbolic  1.0  ASSIGNMENT
phonMod        symbolic  0.89 <QUESTION>
semantics      structure 1.0  [Command]-> (Modal-Property-Ascription)-> [Mental-Active: ausmachen]
                              - ->(Actee)-> [SITUATION: Termin [Ambient-Process] - ->(Temporal-
                              Locating) -> [Time-Interval] | ->(Participant) -> [Person],] | -
                              >(Actor) -> [Person],.
text           comment        ja prima dann lassen Sie uns doch noch einen Termin ausmachen wann
                              w"are es Ihnen denn recht
```

Figure 6: Feature description of a segment presented to the learning algorithm. The columns contain attribute name, type of attribute, quality and value

References

Novick, D. G. and Sutton, S. 1996. Building on experience: Managing spoken interaction through library subdialogues. In Proc. Twente Workshop on Language Technology, Dialogue Management in NL-Systems, 51-60. Enschede.

Litman, D. J. 1994. Classifying Cue Phrases in Text and Speech Using Machine Learning. In Proc. Annual Meeting of the American Association for Artificial Intelligence, 806-813. Seattle.

Siegel, E. V. and McKeown, K. R. 1994. Emergent Linguistic Rules from Inducing Decision Trees: Disambiguating Discourse Clue Words. In Proc. Annual Meeting of the American Association for Artificial Intelligence, 820-826. Seattle.

Lehnert, W. and Soderland, S. 1994. Corpus-Driven Knowledge Acquisition for Discourse Analysis. In Proc. Annual Meeting of the American Association for Artificial Intelligence, 827-832. Seattle.

Mast, M., Kompe, R., Harbeck, S., Kiessling, A., Niemann, H., Nöth, E., Schukat-Talamazzini, E. G., and Warnke, V. 1996. Dialog Act Classification with the Help of Prosody. In Fourth International Conference on Spoken Language Processing, 1732-1735. Philadelphia, PA, USA.

Wermter, S. and Löchel, M. 1996. Learning dialog act processing. In Coling96, 740-745. Kopenhagen.

Andernach. 1996. A Machine Learning Approach to the Classification of Dialogue Utterances. In NeMLaP-2, Ankara, Turkey.

Hübener, K., Jost, U., and Heine, H. 1996. Speech recognition for spontaneously spoken German dialogs. In 4th Int. Conference on Spoken Language Processing, Philadelphia.

Strom, V. 1995. Detection of accents, phrase boundaries, and sentence modality in German. In EUROSPEECH 1995, 2039-2041. .

Breuker, J. and Wielinga, B. 1986. Models of Expertise. In Proc. of the European Conference on Artificial Intelligence ECAI-86, 306-318. .

Sundin, U. 1994. Assignment and Scheduling. In *CommonKADS Library for Expertise Modelling - Reusable Problem Solving Components*, J. V. d. V. Breuker, W., Ed. Amsterdam: IOS Press: 231-264.

Weber, V. and Wermter, S. 1995. Towards Learning Semantics of Spontaneous Dialog Utterances in a Hybrid Framework. In AISB95, .

Eckert, W. 1996. Understanding of Spontaneous Utterances in Human-Machine-Dialog. In Proc. Twente Workshop on Language Technology, Dialogue Management in NL-Systems, 139-148. Enschede.

Elsner, A. and Klein, A. 1996. Erkennung des prosodischen Fokus und die Anwendung im dialogaktbasierten Transfer. Verbmobil, VM-Memo 107, ??

Sowa, J. F. 1984.*Conceptual Structures - Information Processing in Mind And Machine*. Reading, MA: Addison-Wesley.

Jekat, S., Klein, A., Maier, E., Maleck, I., Mast, M., Dr.-Ing., and Quantz, J. J. 1995. Dialogue Acts in VERBMOBIL. Verbmobil, VM-Report 65, ??

Reithinger, N. and Maier, E. 1995. Utilizing Statistical Dialogue Act Processing in VERBMOBIL. Verbmobil, VM-Report 80, ??

Möller, J.-U. 1997. CLASSITALL: Incremental and Unsupervised Learning in the DIA-MOLE Framework. In European Conference on Machine Learning, Workshop Notes on Empirical Learning of Natural Language Processing Tasks, 95-104. Prague, Czech Republic.

Fisher, D. H. 1987. Knowledge Acquisition via Incremental Conceptual Clustering. *Machine Learning*: 139-172.

Gennari, J. H., Langley, P., and Fisher, D. H. 1989. Models of Incremental Concept Formation. *Artificial Intelligence*: 11-61.

Woods, W. A. 1991. Understanding Subsumption and Taxonomy: A Framework for Progress. In Principles of Semantic Networks: Explorations in the Representation of Knowledge, J. F. Sowa, Ed. San Mateo, CA: Morgan Kaufmann Publ.: 45-94.

Levinson, R. 1991. Multi-Level Hierarchical Retrieval. In 6. Annual Workshop on Conceptual Graphs, 67-81. SUNY Binghampton.

Ellis, G. 1992. Compiled Hierarchical Retrieval. In *Conceptual Structures - current research and practice*, T. E. Nagle, J. A. Nagle, L. L. Gerholz, and P. W. Eklund, Eds. New York, London: Ellis Horwood: 271-293.

Kietz, J.-U. and Morik, K. 1994. A Polynomial Approach to the Constructive Induction of Structural Knowledge. *Machine Learning*: 193-217.

Bournaud, I. and Ganascia, J.-G. 1995. Conceptual Clustering of Complex Objects: A Generalization Space based Approach. In Conceptual Structures: Applications, Implementation and Theory - Proceedings of the Third International Conference on Conceptual Structures, 173-187. Santa Cruz, USA.

Mineau, G. W. 1995. Automatic Structuring of Knowledge Bases by Conceptual Clustering. *IEEE Transactions on Knowledge and Data Engineering*, vol. 7: 824-829.

Lebowitz, M. 1985. Categorizing numeric information for generalisation. *Cognitive Science*, vol. 9: 285-309.

Schirdewan, R. 1996. Transformation des PENMAN Upper Models in Conceptual Graphs. In *Computer Science Dept.* Hamburg: University of Hamburg.