Semantic Dialogue Modeling

Günther Wirsching[1], Markus Huber[2], Christian Kölbl[2],
Robert Lorenz[2], and Ronald Römer[3]

[1] Katholische Universität Eichstätt-Ingolstadt, Math.-Geogr. Fakultät
guenther.wirsching@ku-eichstaett.de
[2] Universität Augsburg, Institut für Informatik
{markus.huber,christian.koelbl,robert.lorenz}@informatik.uni-augsburg.de
[3] Brandenburgische Technische Universität Cottbus, Fakultät 3
ronald.roemer@tu-cottbus.de

Abstract. This paper describes an abstract model for the semantic level of a dialogue system. We introduce mathematical structures which make it possible to design a semantic-driven dialogue system. We describe essential parts of such a system, which comprise the construction of feature-values relations representing meaning from a given world model, the modeling of the flow of information between the dialogue strategy controller and speech recogniser by a *horizon of comprehension* and the *horizon of recognition results*, the connection of these horizons to wordings via *utterance-meaning pairs*, and the incorporation of new horizons into a state of information. Finally, the connection to dialogue strategy controlling is sketched.

Keywords: entity-relationship, weighted feature-values relation, semantic representation, utterance-meaning pairs, dialogue modeling.

1 Introduction

This paper describes an abstract model for the semantic level of a dialogue system. The task of such a system is to collect the data needed to perform certain actions. Technically, this can be described as extraction, insertion, deletion, and change, of entries in a database. We model the information available to the system using the mathematical notion *weighted feature-values relation*. The feature-values relation flowing through the system are algorithmically derived from a world model containing data and actions, where the data is given via an SQL database and an appropriate entity-relationship (abbreviated ER) diagram.

The connection between the semantic level and an automatic speech recogniser is given by *utterance-meaning pairs*, which we motivate by a model stemming from behavioristic psychology. Utterance-meaning pairs associate feature-values relations representing meaning to possible wordings expressing a meaning, and vice versa. Technically, the association from wordings to meanings can be realised by a weighted finite state transducer, which we call the *UMP-transducer*. This chaining of a representation of semantic by a feature-values relation on one hand,

and a language model describing possible wordings to express the meanings on the other hand, allows the design of a semantic-driven dialogue system.

On the semantic level, we store the background information of the system in a *state of information*, which is also a weighted feature values relation. The flow of information between the semantic level and a speech recogniser is given by dynamically generated *horizons* which contain, in each situation, the meanings which may play a role in the given situation. In a given dialogue turn, when a user input is expected, a *horizon of comprehension* is sent to the recogniser. Using utterance-meaning pairs, the recogniser is able to construct dynamically an appropriate language model which can be used for recognition. The recognition results are sent to the UMP-transducer, which converts them into a *horizon of recognition results*, also represented as weigthed feature-values relation. Now the task of the dialogue strategy controller is to incorporate the horizon of recognition results into the state of information, and to decide what to do next, based on the now available information.

The paper starts with a description of the world model, a formal definition of feature-values relation, and an indication how our algorithm constructs feature-values relations from the world model. The flow of information is illustrated by an example dialogue, followed by an introduction to utterance-meaning pairs, and a description how to construct a horizon of comprehension in a given situation. Finally, it is indicated how the dialogue strategy controller has to deal with the state of information and the horizons.

2 World Model and Feature-Values Relation

Our point of departure is a world model consisting of two parts: a set of data, and a set of possible actions. For definiteness and simplicity, we assume that the data is given via an SQL-database, together with an appropriate ER-diagram, but we emphasize that the structures which we use in the sequel can also be derived from other data structures. With respect to the action, we assume that a list of possible action is given, where to each action, possible sets of data needed to perform the action are specified.

We use *feature-values relations* as the mathematical structure carrying semantic information. Here is a formal definition:

Definition 1. *A* feature-values relation *(FVR) is a finite acyclic labeled directed graph* $R = (V, \to, \ell)$*, where*

- V *is a finite set of labels,*
- $\to \subset V \times V$ *is an acyclic relation,*
- $\ell : V \to L$ *is a labeling of vertices, where* L *is a set of labels.*

If an FVR is given, an *initial vertex* is, by definition, a vertex without incoming arrow, and a *terminal vertex* is one without outgoing arrow.

In [2], an algorithm which transforms a pair consisting of an SQL-database and an appropriate ER-diagram into a feature-values relation is described. The

algorithm starts by constructing, to each given entity type, an *elementary* FVR modeling the attributes and relations of the given entity type. Moreover it contains structures which we call *anchors*, where each anchor corresponds to an entity type which is involved in some relation with the given entity type, and to its role in the relation.

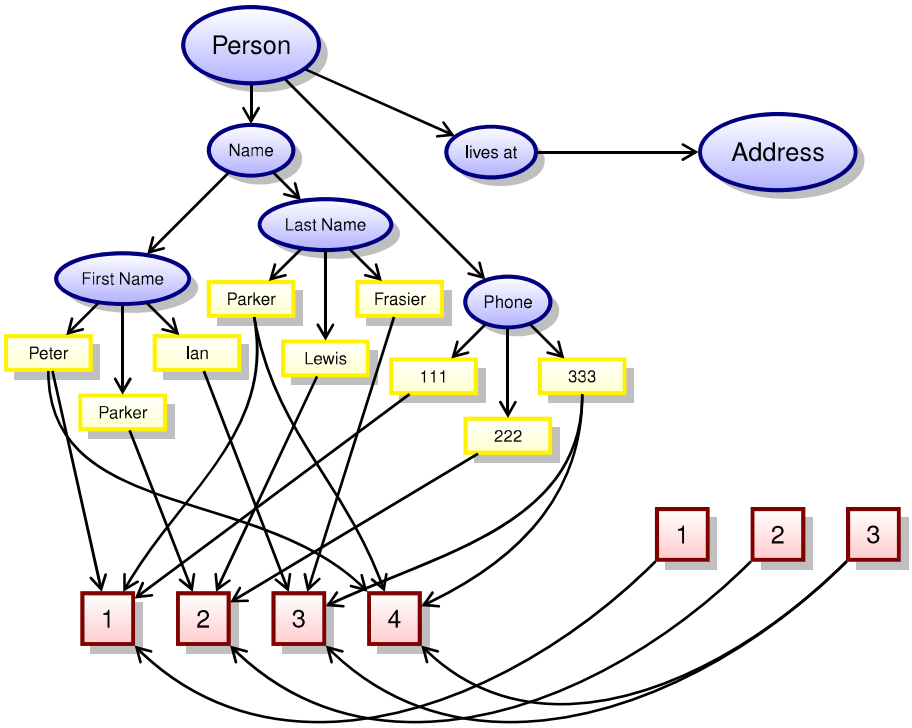

Fig. 1. The elementary feature-values relation associated to entity type "Person", with an anchor associated to entity type "Address". The ID-layer of the anchor is connected the ID-layer of the initial FVR according to the relation given by the database.

In our simple example, the given entity type is "Person" with attributes "First Name", "Last Name", and "Phone", and a relation "lives at" connecting each person to an entity of type "Address". The constructed elementary FVR uses the given entity type "Person" as *root feature*, which is an initial vertex in the elementary FVR, and an *ID-layer* consisting of a set of terminal vertices. By construction, there is a one-one-correspondance between IDs in the ID-layer and entities of the given type in our SQL-database.

In figure 1, there is also an anchor: it consists of a terminal vertex labeled "Address", which is reachable from the vertex labeled "Person" via the relation "lives at". an a set of initial vertices corresponding to the IDs of entities of type "Address" in the SQL-database. The anchor can be thought of as a placeholder for the elementary FVR constructed from the entity type "Address".

Starting with an elementary FVR and putting copies of elementary FVRs, as far as needed, in appropriate anchors, we have a recursive construction of FVRs

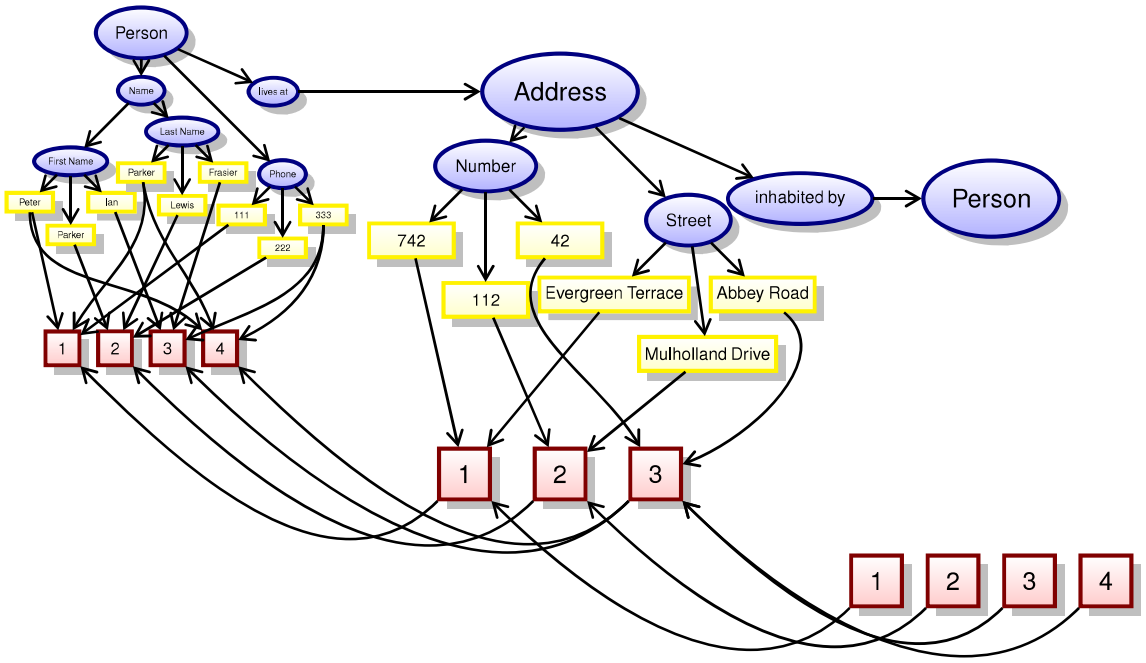

Fig. 2. Recursive construction of an FVR to entity type "Person", with the anchor associated to entity type "Address" filled by the corresponding elementary FVR

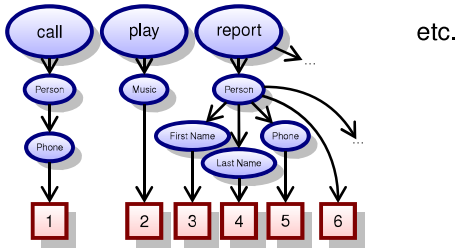

Fig. 3. A set of possible actions, represented as FVR

from an ER-diagram and an appropriate SQL-database. The recursion depth is, in principle, arbitrary (but finite).

The set of actions can also be represented as FVR, as is indicated in figure 3.

3 An Example Dialogue

USER: I want to call Parker.
SYSTEM: Is Parker the first name?
USER: No, I mean Peter Parker.
SYSTEM: Which Peter Parker do you want to call?
USER: Change that terrible song to something from Johnny Cash.
SYSTEM: Which album by Johnny Cash?
USER: At San Quentin
⟨`system starts playing`⟩

SYSTEM: Which Peter Parker do you want to call?
USER: The one who lives at 742 Evergreen Terrace.
⟨`calling the selected partner`⟩

4 Meaning and Utterance

The control of a dialogue system relies on the *meanings* of what the user says. The system has to gather those pieces of information which are needed for performing a specific task but are not yet given by the user. Which means it has to ask for it. The following properties will help to clarify our ideas how we model *meanings* flowing through a dialogue system:

- Important for dialogue control are the *meanings* of each utterance, not the precise wording.
- *Meaning* can be represented by a feature-values relation.
- *Meaning* is conveyed by an *utterance*.

In order to get an idea how *meaning* is connected to an *utterance*, we have a look on an idea from psychology. Skinner [4] applies the formal scheme, crucial for behaviorism,

$$\text{Stimulus} \longrightarrow \text{Response} \longrightarrow \text{Consequences}$$

to "verbal behavior" as follows:

Stimulus: the context of a verbal behavior,
Response: the utterance itself,
Consequences: possible impacts in the given context.

Moreover, he asserts that *meaning*

- is not a property of the utterance,
- is to be constructed from context and consequences.

In these terms, the ideal aim of behavioristic psychology is to describe, given stimulus and consequences, the set of possible fitting utterances, together with a probability distribution on this set. If this aim could be reached, it would also be perfect for the speech recognition task in dialogue modeling: a given probability distribution on a given set of utterances can be transformed into a language model apt for configuring a speech recognizer. The language model would be optimal for the speech recognition task, if it represents the 'true' probability distribution of utterances in the given situation defined by stimulus and consequences.

With this language modeling aim in mind, a formalization of the Stimulus-Response-Consequences scheme into a mathematical concept "utterance-meaning pair" is described in [8]. Here we note just the definition:

Definition 2. *An* utterance-meaning pair *consists of an* utterance*, described as a word sequence, and a* meaning*, given by a feature-values relation.*

Note that, at this stage, we do not specify the way in which the word sequence describing the utterance is given to the system. In fact, there are different possibilities:

1. As a sequence of words in usual graphemic notation.
2. A phonetic transliteration, taking into account possible slurring of words, or other phonetic variations.
3. Either of the above, enriched by additional prosodic and/or dynamic information.

Moreover, note that the relation "utterance $\leftrightarrow$ meaning" usually is many-to-many:

- Two different utterances may have the same meaning.
- One utterance may have more than one meaning.

Example 1. Here is a simple example of an utterance with two possible meanings:

- Utterance: "I want to *call Parker*"
- Meaning 1: Action = Call, First Name = Parker.
- Meaning 2: Action = Call, Last Name = Parker.

Utterance-meaning pairs are the "atoms" for *semantic dialogue modelling*. In functional regard, which is the important one for dialogue modelling, we may always view the set of utterance-meaning pairs as a mathematical relation, i. e., as a subset of the cartesian product of a set of possible utterances with a set of possible meanings. But it is generally not necessary to store the needed utterance-meaning pairs in a large list. In many cases, it is preferable to define them implicitly in a grammar, and to use a finite state transducer (abbreviated FST) to configure a speech recognizer with a set of utterance-meaning pairs. In addition, the FST may be enhanced with weights, the computation of which is, ideally, based on statistical data from language observations.

5 Horizon of Comprehension

In semantic dialogue modelling, we view a speech recogniser as a black box with three input channels and one output channel, as depicted in figure 4.

In this setting, the *horizon of comprehension* is to be given as a set of possible meanings. As before, there is no need to store this as a (possibly large) set of meanings; it suffices to have an implicit algorithmic description enabling the system to construct this set. The horizon of comprehension can also be endowed with a weight for each meaning. These weights should, for instance, represent *a priori* knowledge about which meaning the user is more (or less) likely to use. If possible, the weights can be chosen to encode Bayesian prior probabilities to each possible meaning.

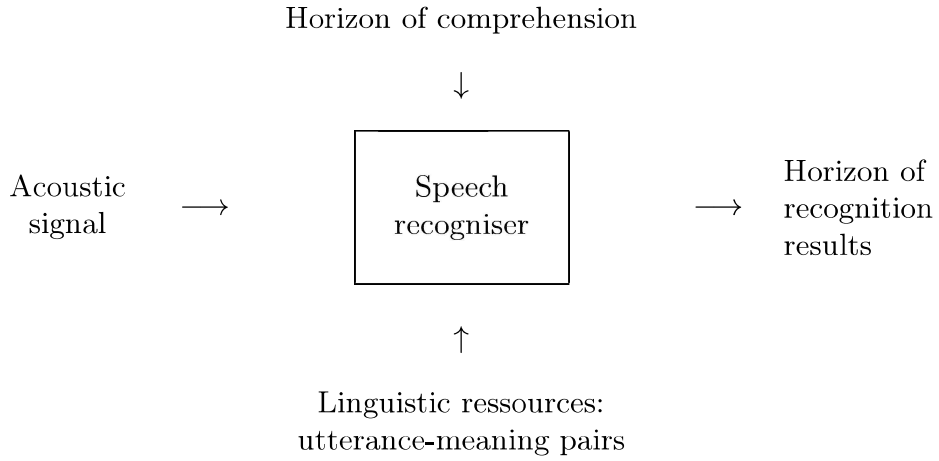

Fig. 4. Configuration of the speech recogniser in semantic dialogue modeling

At each dialogue turn, the speech recogniser is to be given all meanings which should be understandable in the actual context. In a given context (a given dialogue turn), the horizon of comprehension is just the set of meanings which should be understandable in this context. As described in [8], this set can be divided into five parts:

$\mathcal{E}$ **(Horizon of Expectation):**
Set of meanings exactly asked for by the prompt.
$\mathcal{U}$ **(Underanswering):**
Set of meanings answering the prompt only partially.
$\mathcal{O}$ **(Overanswering):**
Set of meanings containing more information than asked for by the prompt.
$\mathcal{D}$ **(Deviating answer):**
Set of meanings overanswering part of what has been asked for.
$\mathcal{G}$ **(Generally available meanings):**
Set of generally available meanings,
e.g., aborting or interrupting the current task.

Each of these sets is a set of meanings. Having, in the background, a set UMP of given utterance-meaning pairs, UMP defines a map associating to each given set $\mathcal{M}$ of meanings a set $w(\mathcal{M})$ of utterances u with the property that there is a meaning $m \in \mathcal{M}$ such that $(u, m) \in \text{UMP}$.

Example 2. Let us consider a context defined by

USER: "I want to call Parker."
SYSTEM: "Is Parker the first name?"

Now the system is waiting for an answer, and the speech recogniser should be configured in a way enabling it to understand any reasonable answer. Here are some examples of utterances for the different parts of the horizon of comprehension:

"No, Parker is the last name." $\in w(\mathcal{E})$
"I don't know." $\in w(\mathcal{U})$
"No, I mean Peter Parker." $\in w(\mathcal{O})$
"I don't know, but he lives at 742 Evergreen Terrace." $\in w(\mathcal{D})$
"Abort calling Parker." $\in w(\mathcal{G})$
"Change that terrible song to something from Johnny Cash." $\in w(\mathcal{G})$

Now we are ready to explain figure 4 more specifically.

- The *horizon of comprehension* is, clearly, context-dependent; it changes from dialogue turn to dialogue turn. In each situation, it depends on the most recent system prompt, on information which the system had received previously, and on the general context of the dialogue which includes all possible executable actions.
- The *linguistic resources* have to be structured in such a way that, for a given set $\mathcal{M}$ of meanings, the set of "wordings" $w(\mathcal{M})$ is easily accessible. Ideally, these sets are endowed with weights for each wording, which can be combined with weights from the horizon of comprehension to give a probabilistic language model for the recogniser.
- The *horizon of recognition results* (abbreviated HoRs) is a weighted set of possible meanings, where the weights are computed from recognition scores. A method for this computation is given in [6]. Note that we don't need the precise wordings of the recognition results, dialogue control works exclusively with meanings. In figure 4, we understand that parsing is included in recognition.

6 The State of Information

On the semantic level, the necessary information is stored in a *State of Information* (abbreviated SoIn). The mathematical structure of the SoIn is weighted FVR, where the weights, and, if necessary, also the structure are changed during the dialogue. An example for a SoIn is given in figure 5.

6.1 Storing Information

The first task of the global SoIn is to store the information collected by extracting possible meanings from the utterances of the user.

Parallel Worlds. On the uppermost level, the global SoIn is a set of unconnected conflicting SoIns representing *parallel worlds*. Each parallel world is equipped with a confidence indicating how sure the system is that this world is what the user intended.

Concentrating on a specific parallel world, the next level consists of a stack of sub-Soins, where each sub-SoIn corresponds to a possible topic.

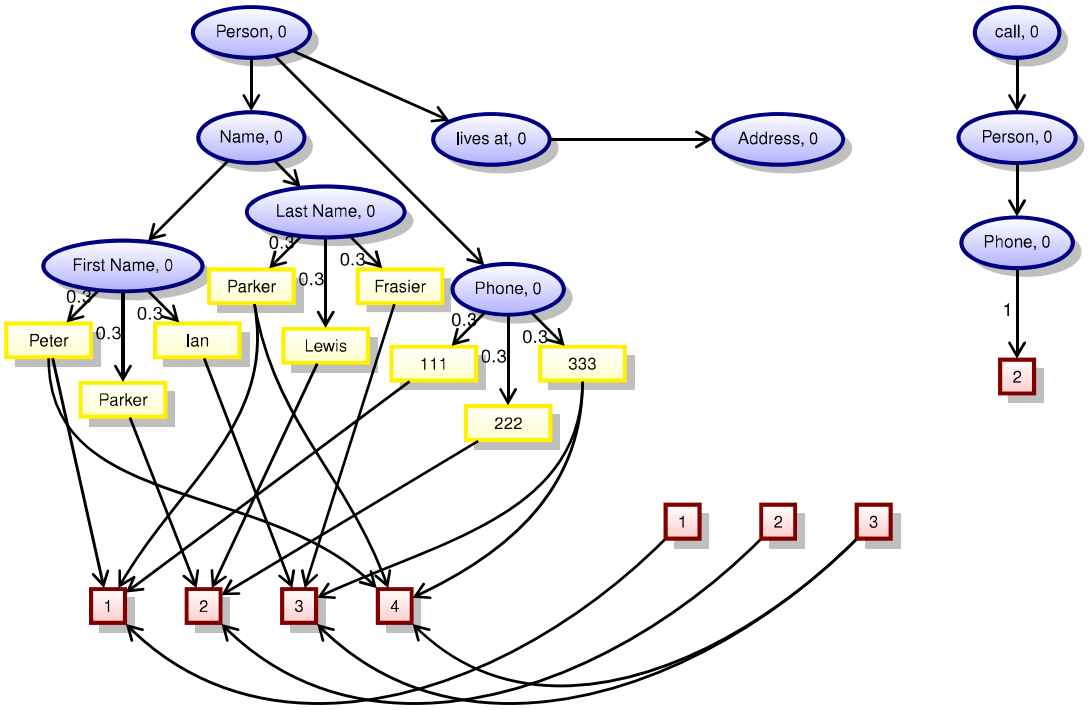

Fig. 5. A State of Information

Mathematical Structure Essentially, each sub-SoIn consists of two parts:

1. The *action part* incorporates identifiers for the possible actions the system is able to perform. Each possible action is equipped with a weight giving an estimate for the probability that the user intends this action. At each dialogue turn, these estimates are to be updated from the weights of the meanings understood by the speech recogniser.
2. The *data part* containing the references to data from the database. It is a weighted FVR, where the weights are appropriately initialized and updated at each dialogue turn. The FVR representing the data part of a sub-SoIn is built recursively from elementary FVRs extracted from the ER-diagram and the data. At each dialogue turn, both the recursion depth and the weights depend on the dialogue history up to that turn.

6.2 The Update Process

The update of the SoIn after an utterance from the user was processed by the speech recogniser is based on the result of this processing, the HoRs, which again is a weighted FVR.

Initially, the HoRs is a list of meanings, where each meaning comes with a score representing its *Bayesian a posterior* probability. (Here the recognition is modeled as Bayesian update process where the prior is given by the language model and the result is the posterior.) The HoRs is separated into sets of meanings with common feature structure, where each feature structure corresponds to

a parallel world; see [6] for more details. Then each set of meanings belonging to one parallel world is incorporated into the appropriate parallel world. According to [3] all involved weighted FVRs can be converted to weighted FSTs and the update can be computed by FST-algorithms.

6.3 The Dialogue Strategy Controller

For the time being, we consider the dialogue strategy controller (abbreviated DiSCo) as a black box with the following specification:

Input: the old SoIn plus the updated SoIn.
Output: a new SoIn plus a horizon of comprehension for the next dialogue turn.
Task: apply strategies to disambiguate collected meanings, and decide what is the next piece of information to be asked for.

References

1. Huber, M., Kölbl, C., Lorenz, R., Wirsching, G.: Ein Petrinetz-Modell zur Informationsübertragung per Dialog. In: Proceedings of the 15th German Workshop on Algorithms and Tools for Petri Nets, AWPN 2008, Rostock, Germany, September 26-27, pp. 15–24 (2008)
2. Huber, M., Kölbl, C., Lorenz, R., Römer, R., Wirsching, G.: Semantische Dialogverarbeitung mit gewichteten Merkmal-Werte-Relationen. In: Hoffmann, R. (Hrsg.) Elektronische Sprachsignalverarbeitung 2009, Tagungsband der 20. Konferenz, Dresden, 21. bis 24. Studientexte zur Sprachkommunikation, vol. 54, pp. S.25–S.32 (September 2009)
3. Kölbl, C., Huber, M., Wirsching, G.: Endliche gewichtete Transduktoren als semantischer Träger. In: Kröger, B.J., Birkholz, P. (Hrsg..) Elektronische Sprachsignalverarbeitung 2011, Tagungsband der 22. Konferenz, Aachen, 28. bis 30. Studientexte zur Sprachkommunikation, vol. 61, pp. S.176–S.183 (September 2011)
4. Skinner, B.F.: Verbal Behavior. Prentice Hall, Englewood Cliffs (1957)
5. Wirsching, G., Huber, M.: Kölbl, C.: The confidence-probability semiring. Technischer Bericht 2010–04, Institut für Informatik der Universität Augsburg (2010)
6. Wirsching, G., Kölbl, C., Huber, M.: Zur Logik von Bestenlisten in der Dialogmodellierung. In: Kröger, B.J., Birkholz, P. (Hrsg.) Elektronische Sprachsignalverarbeitung 2011, Tagungsband der 22. Konferenz, Aachen, 28. bis 30. Studientexte zur Sprachkommunikation, vol. 61, pp. S.309–S.316 (September 2011)
7. Wirsching, G.: Semirings Modeling Confidence and Uncertainty in Speech Recognition, Preprint Mathematik, KU Eichstätt-Ingolstadt (2011), http://edoc.ku-eichstaett.de/6083/
8. Wirsching, G., Kölbl, C.: Language Modeling with Utterance-Meaning Pairs. Technischer Bericht 2011–12, Institut für Informatik der Universität Augsburg (2011)