Towards Measuring Scalability in Natural Language Understanding Tasks

Robert Porzel **Rainer Malaka**

European Media Laboratory
Schloss-Wolfsbrunnenweg 33
D-69118 Heidelberg, Germany
{robert.porzel,rainer.malaka@eml-d.villa-bosch.de}

Abstract

In this paper we present a discussion of existing metrics for evaluation the performance of individual natural language understanding systems and components as well as the commonly employed metrics for measuring the specific task difficulties. We extend and generalize the common majority class baseline metric and introduce an general entropy-based metric for measuring the task difficulty of arbitrary language understanding tasks. Finally, we show an empirical study evaluating this metric followed by a discussion of its role in measuring the scalability of language understanding systems and components.

1 Introduction

Current evaluation frameworks for uni- or multi-modal dialogue systems (Walker et al., 2000; Beringer et al., 2002) that allow for spoken language input do not include metrics for measuring the accuracy of the involved intention recognition systems, simply because such information is hard to extract automatically from log files. Furthermore no general computational method or framework for measuring the difficulty of natural language understanding tasks have been proposed so far. We are, therefore, faced with a lack of methods for measuring the difficulties of the individual tasks involved in the language understanding process. Such generally applicable methods, however, are needed for measuring the scalability of natural language understanding systems and components.

In this paper we first discuss existing metrics for measuring task-specific performances and the corresponding baseline metrics in natural language understanding in Section 2. We, then, propose a generalized baseline-based metric in Section 4.1 as well as a general entropy-based metric in Section 4.2. Both methods can be employed for measuring the difficulties of various understanding tasks and, consequently, for evaluating natural language understanding components involved in the intention recognition process. Section 5 provides a case study evaluation of the proposed methods. In Section 6 we discuss how an analysis of a specific system on tasks differing in their difficulty can yield a first approach for measuring the scalability of a natural language understanding systems and its components.

2 Evaluating Dialogue-, Speech- and Discourse Understanding Systems

In this section we will briefly sketch out the most frequently used metrics for evaluating the performances of the relevant components and systems at hand.

Evaluation of the Dialogue Systems Performance: For evaluation of the overall performance of a dialogue system as a whole frameworks such as PARADISE (Walker et al., 2000) for unimodal and PROMISE (Beringer et al., 2002) for multimodal systems have set a *de facto* standard. These frameworks differentiate between:

- dialogue efficiency metrics, i.e. elapsed time, system- and user turns
- dialogue quality metrics, mean recognition score and absolute number as well as percentages of time-outs, rejections, helps, cancels, and barge-ins,
- task success metrics, task completion (per survey)
- user satisfaction metrics (per survey)

These metrics are crucial for evaluating the aggregate performance of the individual components, they cannot, however, determine the amount of understanding *versus* misunderstanding or the system-specific *a priori* difficulty of the understanding task. Their importance, however, will remain undiminished, as ways of determining such global parameters are vital to determining the aggregate usefulness and felicity of a system as a whole. At the same time individual components and ensembles thereof

- such as the performance of the uni- or multi-modal input understanding system - need to be evaluated as well to determine bottlenecks and weak links in the discourse understanding processing chain.

Evaluation of the Automatic Speech Recognition Performance: The commonly used word error rate (WER) can be calculated by aligning any two sets word sequences and adding the number of substitutions S, deletions D and insertions I. The WER is then given by the following formula where N is the total number of words in the test set.

$$WER = \frac{S + D + I}{N} \times 100$$

Another measure of accuracy that is frequently used is the so called *Out Of Vocabulary* (OOV) measure, which represents the percentage of words that was not recognized despite their lexical coverage. WER and OOV are commonly intertwined together with the combined acoustic- and language-model confidence scores, which are constituted by the posterior probabilities of the hidden Markov chains and n-gram frequencies. Together these scores enable evaluators to measure the absolute performance of a given speech recognition system. In order to arrive at a measure that is relative to the given task-difficulty, this difficulty must also be calculated, which can be done by means of measuring the perplexity of the task (see Section 3).

Evaluation of the Natural Language Understanding Performance: A measure for understanding rates - called *concept error rate* has been proposed for example by Chotimongcol and Rudnicky (2001) and is designed in analogy to word error rates employed in automatic speech recognition that are combined with keyword spotting systems. Chotimongcol and Rudnicky (2001) propose to differentiate whether the erroneous *concept* occurs in a *non-concept slot* that contains information that is captured in the grammar but not considered relevant for selecting a system action (e.g., politeness markers, such as *please*), in a *value-insensitive slot* whose identity, suffices to produce a system action (e.g., affirmatives such as *yes*), or in a *value-sensitive slot* for which both the occurrence and the value of the slot are important (e.g., a goal object, such as *Heidelberg*). An alternative proposal for concept error rates is embedded into the speech recognition and intention spotting system by Lumenvox[1], wherein two types of errors and two types of non-errors for concept *transcriptions* are proposed:

- A *match* when the application returned the correct concept and an *out of grammar match* when the application returned no concepts, or discarded the returned concepts because the user failed to say any concept covered by the grammar.
- A *grammar mismatch* when the application returned the incorrect concept, but the user said a concept covered by the grammar and an *out of grammar mismatch* when the application returned a concept, and chose that concept as a correct interpretation, but the user did not say a concept covered by the grammar.

[1] www.lomunevox.com/support/tunerhelp/Tuning/Concept_Transcription.htm

Neither of these measures are suitable for our purposes as they are known to be feasible only for context-insensitive applications that do not include discourse models, implicit domain-specific information and other contextual knowledge as discussed in (Porzel et al., 2004). Therefore this measure has also been called *keyword recognition rate* for single utterance systems. In our minds another crucial shortcoming is the lack of comparability, as these measures do not take the general difficulty of the understanding tasks into account. Again, this has been realized in the automatic speech recognition community and led to the so called *perplexity* measurements for a given speech recognition task. We will, therefore, sketch out the commonly employed perplexity measurements in Section 3.

The most detailed evaluation scheme for discourse comprehension, introduced by Higashinaka et al. (2002) and also extended by Higashinaka et al. (2003), features the metrics given in Table 2.

1.	slot accuracy
2.	insertion error rate
3.	deletion error rate
4.	substitution error rate
5.	slot error rate
6.	update precision
7.	update insertion error rate
8.	update deletion error rate
9.	update substitution error rate
10.	speech understanding rate
11.	slot accuracy for filled slots
12.	deletion error rate for filled slots
13.	substitution error rate for filled slots

Table 1: Discourse Comprehension Measurements

These metrics are combined by means of combining the results of an m^5 multiple linear regression algorithm and a support vector regression approach. The resulting weighted sum is compared to human intuitions and PARADISE-like metrics concerning task completion rates and -times. While this promising approach manages

to combine factors related to speech recognition, interpretation and discourse modeling, there are some shortcomings that stem from the fact that this schema was developed for single-domain systems that employ frame-based attribute value pairs for representing the user's intent.

Recent advances in dialogue management and multi-domain systems enable approaches that are more flexible than slot-filling, e.g. using discourse pegs, dialogue games and overlay operations for handling multiple tasks and cross-modal references (LuperFoy, 1992; Löckelt et al., 2002; Pfleger et al., 2002; Alexandersson and Becker, 2003). More importantly - for the topic of this paper - no means of measuring the *a priori* discourse understanding difficulty is given.

Measuring Precision, Recall and F-Measures: In the realm of semantic analyses the task of word sense disambiguation is usually regarded to be among the difficult ones. This means it can only be solved after all other problems involved in language understanding have been resolved as well. The hierarchical nature and interdependencies of the various tasks are mirrored in the results of the corresponding competitive evaluation tracts - e.g. the message understanding conference (MUC) or SENSEVAL competition. It becomes obvious that the ungraceful degradation of f-measure scores (shown in Table 2 is due to the fact that each higher-level task inherits the imprecisions and omissions of the previous ones, e.g. errors in the named entity recognition (NE) task cause recall and precision declines in the template element task (TE), which, in turn, thwart successful template relation task performance (TR) as well as the most difficult scenario template (ST) and co-reference task (CO). This decline can be seen in Table 2 (Marsh and Perzanowski, 1999).

NE	CO	TE	TR	ST
f $\leq$ .94	f $\leq$ .62	f $\leq$ .87	f $\leq$ .76	f $\leq$ .51

Table 2: F-measure-based ($\alpha = 0.5$) evaluation results of the best performing systems of the 7th Message Understanding Conference

Despite several problems stemming from the prerequisite to craft costly gold standards, e.g. tree banks or annotated test corpora, precision and recall and their weighable combinations in the corresponding f-measures (such as given in Table 2), have become a *de facto* standard for measuring the performance of classification and retrieval tasks (Van Rijsbergen, 1979). Precision p states the percentage of correctly tagged (or classified) entities of all tagged/classified entities, whereas recall r states the positive percentage of entities tagged/classified as compared to the normative amount, i.e. those that ought to have been tagged or classified. Together these are combinable to an overall f-measure score, defined as:

$$F = \frac{1}{\alpha\frac{1}{p} + (1-\alpha)\frac{1}{r}}$$

Herein α can be set to reflect the respective importance of p *versus* r, if $\alpha = 0.5$ then both are weighted equally. These measures are commonly employed for evaluating part-of-speech tagging, shallow parsing, reference resolution tasks and information retrieval tasks and sub-tasks.

An additional problem with this method is that most natural language understanding systems that perform deeper semantic analyses produce representations often based on individual grammar formalisms and mark-up languages for which no gold standards exist. For evaluating discourse understanding systems, however, such gold standards and annotated training corpora will continue to be needed.

3 Measuring Perplexity and Baselines

In this section we will describe the most frequently used metrics for estimating the complexity of the tasks performed by the relevant components and systems at hand.

Measuring Perplexity in Automatic Speech Recognition: Perplexity is a measure of the probability weighted average number of words that may follow after a given word (Hirschman and Thompson, 1997). In order to calculate the perplexity B, the entropy H needs to be given - i.e., the probability of the word sequences in the specific language of the system W. The perplexity is then defined as:

$$H = -\sum_{\forall W} P(W) log_2 P(W)$$

$$B = 2^H$$

Improvements of specific ASR systems can then consequently be measured by keeping the perplexity constant and measuring WER and OOV performance for recognition quality and confidence scores for hypothesis verification and selection.

Measuring Task-specific Baselines: Baselines for classification or tagging tasks are commonly defined based on chance performance, on an *a posteriori* computed majority class performance or on the performance of an established baseline classification method such as naive bayes, tf*idf or k-means . That means:

- what is the corresponding f-measure, if the evaluated component guesses randomly - for chance performance metrics,

- what is the corresponding f-measure if the evaluated component always chooses the most frequent solution - for majority class performance metrics,
- what is the corresponding f-measure of the established baseline classification method.

Much like kappa statistics proposed by Carletta (1996), existing employments of majority class baselines assume an equal set of identical potential mark-ups, i.e. attributes and their values, for all markables. Therefore, they cannot be used in a straight forward manner for many tasks that involve disjunct sets of attributes and values in terms of the type and number of attributes and their values involved in the classification task. This, however, is exactly what we find in natural language understanding tasks, such as semantic tagging or word sense disambiguation tasks (Stevenson, 2003). Additionally, baseline computed on other methods cannot serve as a means for measuring scalability, because of the circularity involved: as one would need a way of measuring the baseline method's scalability factor in the first place. Table 3 provides an overview of the existing ways of measuring performance and task difficulty in automatic speech recognition and understanding.

Domain	Performance	Complexity
automatic speech recognition	WER/OVV	Perplexity
natural language understanding	CER	none
MUC tasks (NE, TE, TR, ST, CO)	f-measure	baselines
unimodal dialogue system	PARADISE	none
mulitmodal dialogue system	PARADISE	none

Table 3: Summary of Measurements

4 Measuring Task Difficulty

4.1 Proportional Baseline Rates

As a precursor step before this we need a clear definition of a natural language understanding task. For this we propose to assume a MATE-like annotation point of view, which provides a set of disjunct levels of annotations for the individual discriminatory decisions that can be performed on spoken dialogue data, ranging from annotating referring expressions, e.g. named entities and their relations, anaphora and their antecedents, to word senses and dialogue acts. Each task must, therefore, have a clearly defined set of markables, attributes and values for each corpus of spoken dialogue data.

As a first step we will propose a uniform and generic method for computing task-specific majority class baselines for a given task T_w from the entire set of task, i.e. $T = \{T_1, \ldots, T_z\}$ and $T_w \in T$.

A gold standard annotation of a task features a finite set of markable tokens $C = \{c_1, \ldots, c_n\}$ for task T_w, e.g. if $n = 2$ in a corpus containing only the two ambiguous lexemes *bank* and *run* as markables, i.e. c_1 and c_2 respectively. For a member c_i of the set C we can now define the number of values for the tagging attribute of `sense` as: $A_i = \{b^i_1, \ldots, b^i_{n_i}\}$. For example, for three senses of the markable *bank* as c_1 we get the corresponding value set $A_1 = \{$`building`, `institution`, `shore`$\}$ and for *run* as c_2 the value set $A_2 = \{$`motion`, `storm`$\}$. Note that the value sets have markable-dependent sizes. For our toy example containing the two markables c_1 for *bank* and c_2 for *run* they are:

b^i_j	b^i_1	b^i_2	b^i_3
A_1	`building`	`institution`	`shore`
A_2	`motion`	`storm`	

For computing the proportional majority classes we need to compute the occurrences of a value j for a markable i in a given gold standard test data set. We call this V_{ij}. Now we can determine the most frequently given value and its number for each markable c_i as:

$$V_i^{max} = max_{ij \in \{1,\ldots,b_i\}} V_{ij}$$

For example, given a marked-up toy corpus containing our ambiguous lexemes as shown below as task T_1:

> The run$_{storm}$ on the bank$_{building}$ on Monday caused the bank$_{institution}$ to collapse early this week. It employees can therefore now enjoy a leisurely run$_{motion}$ on the bank$_{shore}$ of the Hudson river. It is uncertain if the bank$_{institution}$ can be saved so that they can run$_{motion}$ back to their desks and resume their work.

This results in the list of the value occurrences shown below with V_i^{max} set in bold face:

V_{ij}	b^i_1	b^i_2	b^i_3
c_1	1	**2**	1
c_2	**2**	1	

We define the total number of values for a markablec_i as:

$$V_i^S = \sum_{j=1}^{n_1} V_{ij}$$

With V_i^{max} we define the majority classe baseline as:

$$B_i = \frac{V_i^{max}}{V_i^S}$$

If we always choose the most frequent attribute for markable c_i, the percentage of correct guesses correspomds to B_i. We can now calculate the total number of values as:

$$V^S = \sum_{i=1}^{n} V_i^S$$

Based on this we can compute the task-specific proportional baseline for task T_w, i.e., B_{T_w}, over the entire test set as:

$$B_{T_w} = \frac{1}{V^S} \cdot \sum_{i=1}^{n} V_i^S B_i = \frac{1}{V^S} \cdot \sum_{i=1}^{n} V_i^{max}$$

Thus, B_{T_w} calculates the average of correct guesses for the majority baseline. Returning to our toy example for c_1 we get $V_1^S = 4$ and for c_2 we get $V_2^S = 3$. Additionally, we also get different individual majority class baselines for each markable, i.e., for c_1 we get $B_1 = \frac{1}{2}$, and for c_2 we get $B_2 = \frac{2}{3}$. We also get a total number of values given for C (c_1 and c_2), i.e., $V^S = 7$. Now we can compute the overall baseline B_{T_1} as:

$$\frac{1}{7} \cdot ((4 \cdot \frac{1}{2}) + (3 \cdot \frac{2}{3})) = \frac{4}{7} \approx 0.57$$

If we extend the corpus by an additional ambiguity, i.e. that of the spatial and temporal readings of the lexeme *on*, to yield an annotated corpus such as given below as task T_2:

> The run$_{storm}$ on the bank$_{building}$ on$_{temporal}$ Monday caused the bank$_{institution}$ to collapse early this week. It employees can therefore now enjoy a leisurely run$_{motion}$ on$_{spatial}$ the bank$_{shore}$ of the Hudson river. It is uncertain if the bank$_{institution}$ can be saved so that they can run$_{motion}$ back to their desks and resume their work.

We get the list of the value occurrences shown below:

C	b_1^i	b_2^i	b_3^i
c_1	1	**2**	1
c_2	**2**	1	
c_3	**1**	1	

Now we can compute the overall baseline B_{T_2} again as:

$$\frac{1}{9} \cdot ((4 \cdot \frac{1}{2}) + (3 \cdot \frac{2}{3}) + (2 \cdot \frac{1}{2})) = \frac{5}{9} \approx 0.55$$

The reduction by .02 points, in this case, indicates that a method that for each markable always chooses the most frequently occurring one would perform slightly worse on the second corpus as compared to the first. Note that this proportional baseline measure is able to compute the performance of such a majority class-based method on any data set for any task. It does as such provide a picture depicting a problem's or task's inherent difficulty, but only if the distribution of values for the markables at hand is fairly homogeneous. However, if we assume distributions of markable values such as shown below, we get identical values for B_{T_3} and B_{T_4}.

T_3	b_1^i	b_2^i	b_3^i	b_4^i	b_5^i
c_3	**16**	16	0	0	0
T_4	b_1^i	b_2^i	b_3^i	b_4^i	b_5^i
c_4	**16**	4	4	4	4

That is, we get:

$$\frac{1}{32} \cdot (32 \cdot \frac{1}{2}) = \frac{1}{2} = 0.5$$

for both task baselines B_{T_3} and B_{T_4} with T_3 featuring the task distribution depicted as c_3 and T_4 that of c_4, despite the fact task T_2 was undoubtedly the more difficult one. To create a more applicable measure for task difficulty - i.e. one that also applies for cases of heterogeneous value distributions - we need do need to calculate an entropy metric that takes the individual value distributions into account.

4.2 Measuring Markable-specific Entropy

As a means of illustrating such a markable-specific entropy metric we can look at the *value space* for each markable and define a minimal amount of binary decisions that are on average necessary for solving the problem and compute what part of the problem is solved by them. For example, looking at the markable c_4 from above we find that the problem can be solved by means of the following decisions: With one decision we can partition the space between b_1^4 and the rest (b_2^4 through b_5^4) thereby assigning 16 times the value b_1^4 to c_4. This decision already solves 50% of the problem. Next we need a second decision for partitioning the value space between $b_2^4 \wedge b_3^4$ and $b_4^4 \wedge b_5^4$ and a third for cutting between b_2^4 and b_3^4 as well as b_4^4 and b_5^4 respectively. Therefore, three decisions are needed for assigning the value 4 to b_2^4 and solving 12.5% of the problem. In the case of c_4 the same holds for b_3^4, b_4^4 and b_5^4, giving us the following decision and solution table with $d(b_i^j)$ standing for the average amount of binary decisions necessary for solving b_i^j:

T_4	b_1^4	b_2^4	b_3^4	b_4^4	b_5^4
c_4	16	4	4	4	4
$d(b_i^4)$	1	3	3	3	3
solved	50%	12.5%	12.5%	12.5%	12.5%

Looking at the markable c_3 we find that the problem can be solved as shown below:

T_3	b_1^3	b_2^3	b_3^3	b_4^3	b_5^3
c_4	16	16	0	0	0
$d(b_i^3)$	1	1	0	0	0
solved	50%	50%	0%	0%	0%

As an illustrative approximation of a task's entropy we can now compute the aggregate amount of decisions weighted by their contribution to the overall solution (given as its probability - i.e. 50% = .5). For c_4 this yields:

$$2 = (1 \cdot .5) + (3 \cdot .125) + (3 \cdot .125) + (3 \cdot .125) + (3 \cdot .125)$$

And for T_3 we get:

$$1 = (1 \cdot .5) + (1 \cdot .5)$$

In these cases we can now say that solving the markable-specific value distribution of c_4 is more difficult than solving that of c_3, indicated by the increase of 1 point in this quasi-entropy measure. Note that if we had a binary decision procedure that solves $f\%$ of the cases correctly than we get an average error rate for $T_4 f^2$ of $0.9 \cdot 0.9 = 0,81$ whereas for T_3 only 0.9.

After this approximate illustration of measuring task difficulty via the notion of its entropy, we can now compute a corresponding markable-specific entropy measure H_{c_i} based on the standard formula:

$$H_{c_i} = -\sum_{j=1}^{n_i} P(V_{ij}) log_2 P(b_j^i)$$

This computation yields $H_{c_3} = 1$ and $H_{c_4} = 2$, which also reflects the difference in difficulty of T_3 (consisting of the sole markable c_3) versus T_4 (consisting of the sole markable c_4).

4.3 Combing Markable-specific Entropies

We propose to apply an analogous way of combining the individual markable-specific entropies, by a weighted average, whereby the markable-specific weights are determined by V_i^S and the averaging based on V^S. As an example we return to our sample tasks T_1 and T_2.

V_{ij}	b_1^i	b_2^i	b_3^i	H_{c_i}	V_i^S	$H_{c_i} \cdot V_i^S$
c_1	1	2	1	1.5	4	6
c_2	2	1		≈ 0.92	3	≈ 2.76
c_3	1	1		1	2	2

Based on this we can define H_{T_w} as follows:

$$H_{T_w} = \frac{\sum_{i=1}^{n} (H_{c_i} \cdot V_i^S)}{V^S}$$

Correspondingly, we get for task T_1 consisting of markables c_1 and c_2 a value $H_{T_1} \approx 1.25$ and for task T_2 consisting of markables c_1, c_2 and c_3 a value $H_{T_2} \approx 1.35$.

In much the same way as the proportional baseline rate - only more generally applicable - this increase of 0.1 points in task entropy reflects the increase in task difficulty from T_1 to T_2. Now, that we have a clearly defined way of measuring task-specific difficulties - based on their markable-specific entropies - we can evaluate our approach by means of a larger experiment described below.

5 Evaluating the Metrics

In the following we will report on the results of a corpus study to evaluate the task-specific entropy measurement proposed above. In our mind such a study can be performed in the following way: Given a marked up corpus as an evaluation *gold standard* we can alternate the corpus' difficulty in three potential ways:

- eliminate parts of the corpus so that the number of values of the individual markables is decreased, we will call this *vertical pruning*;
- eliminate parts of the corpus so that the number of the individual values is decreased, we will call this *horizontal pruning*;
- eliminate parts of the corpus so that both the number of markables and their respective values is reduced, we will call this *diagonal pruning*.

Since each of these procedures can increase and reduce the overall task difficulty, we can use them to test if our proposed task entropy measure is able to reflect that in a non toy-world example. For our study we employ the SMARTKOM (Wahlster, 2003) sense-tagged corpus employed in the word sense disambiguation study reported by (Loos and Porzel, 2004). An overview of the markables and their value distributions is given in Appendix 1.

We can now compute the entropy for the whole task as:

$$H_{T_{whole}} = \frac{1966.18083}{2100} \approx 0.94$$

For the horizontal pruning we removed all markables were a single value assumed more than 90% of the entire set. Intuitively that makes the task harder because we took out the easy cases which amounted to about 20% of the entire corpus.

We can now compute the entropy for the horizontally eased task as:

$$H_{T_{horizontal}} = \frac{1718.07959}{1548} \approx 1.11$$

For the vertical pruning we removed all values of b_3^i of the entire set. Intuitively that makes the task easier

because less decisions are necessary to solve those markables that had values in b_3^i.

$$H_{T_{vertical}} = \frac{1894.89386}{2073} \approx 0.91$$

For the diagonal pruning we again removed all values of b_3^i of the entire set making the task easier and removed horizontally all markable where the majority class was under 60%, i.e. the hardest cases.

$$H_{T_{diagonal}} = \frac{1729.4254}{1936} \approx 0.89$$

6 Conclusion and Future Work

We have discussed various measures for evaluating performance of individual components and systems and for estimating the corresponding task complexities. Additionally, we demonstrated the feasibility to employ an entropy-based metric for tasks that are heterogeneously structured in terms of their markable/attribute setup as well as attribute/value distribution. That means it can be applied to any corpora even if they feature disjunct attributes with different values of their set sizes. In a first study, on such a heterogeneous task, we have shown that the results of this generally applicable entropy-based metric line up correspondingly to increases and decreases in task difficulty.

In our minds this metric for measuring task difficulty can now be employed to approach the question of measuring scalability. Since it is now feasible to manipulate task sizes and difficulties in a controlled and measurable fashion, future experiments and studies can be performed that do almost exactly the opposite from current evaluations of systems and components. That is, instead of keeping the task - test corpus - identical and measuring the performance of different methods, we can now keep the method identical and measure its performance on tasks differing in their difficulty. Hereby, some open question still have to be solved, such as evaluating and determining suitable performance measures and formalizing the specific dimensions of scalability that can be measured using this approach, e.g. scalability in terms of performance on problems that are equally difficult but vary in size*versus* problems that vary in size and difficulty to name a few.

References

Jan Alexandersson and Tilman Becker. 2003. The Formal Foundations Underlying Overlay. In *Proceedings of the Fifth International Workshop on Computational Semantics (IWCS-5)*, Tilburg, The Netherlands, February.

Nicole Beringer, Ute Kartal, Katerina Louka, Florian Schiel, and Uli Türk. 2002. PROMISE: A Procedure for Multimodal Interactive System Evaluation. In *Proceedings of the Workshop 'Multimodal Resources and Multimodal Systems Evaluation*, Las Palmas, Spain.

Jean Carletta. 1996. Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics*, 22(2):249–254.

Ananlada Chotimongcol and Alexander Rudnicky. 2001. N-best speech hypotheses reordering using linear regression. In *Proceedings of Eurospeech*, pages 1829–1832, Aalborg, Denmark.

Ryuichiro Higashinaka, Noboru Miyazaki, Mikio Nakano, and Kiyoaki Aikawa. 2002. A method for evaluating incremental utterance understanding in spoken dialogue systems. In *Proceedings of the International Conference on Speech and Language Processing 2002*, pages 829–833, Denver, USA.

Ryuichiro Higashinaka, Noboru Miyazaki, Mikio Nakano, and Kiyoaki Aikawa. 2003. Evaluating discourse understanding in spoken dialogue systems. In *Proceedings of Eurospeech*, pages 1941–1944, Geneva, Switzerland.

Lynette Hirschman and Henry Thompson. 1997. Overview of evaluation in speech and natural language. In R Cole, editor, *Survey of the State of the Art in Human Language Technology*. Cambridge University Press, Cambridge.

Markus Löckelt, Tilman Becker, Norbert Pfleger, and Jan Alexandersson. 2002. Making sense of partial. In *Proceedings of the sixth workshop on the semantics and pragmatics of dialogue (EDILOG 2002)*, pages 101–107, Edinburgh, UK, September.

Berenike Loos and Robert Porzel. 2004. Resolution of lexical ambiguities in spoken dialogue systems. In *Proceedings of the 5th SIGdial Workshop on Discourse and Dialogue*, Boston, USA. Sumitted.

Susann LuperFoy. 1992. The representation of multimodal user interface dialogues using discourse pegs. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*, Newark, Del., 28 June – 2 July 1992, pages 22–31.

Elaine Marsh and Dennis Perzanowski. 1999. MUC-7 evaluation of IE technology: Overview of results. In *Proceedings of the 7th Message Understanding Conference*. Morgan Kaufman Publishers.

Norbert Pfleger, Jan Alexandersson, and Tilman Becker. 2002. Scoring functions for overlay and their application in discourse processing. In *KONVENS-02*, Saarbrücken, September – October.

Robert Porzel, Iryna Gurevych, and Rainer Malaka. 2004. In context: Integrating domain- and situation-specific knowledge. In W. Wahlster, editor,*SmartKom*

- *Foundations of Multimodal Dialogue Systems*. Sprigner, Berlin.

Mark Stevenson. 2003. *Word Sense Disambiguation: The Case for Combining Knowldge Sources*. CSLI.

C. J. Van Rijsbergen. 1979. *Information Retrieval, 2nd edition*. Dept. of Computer Science, University of Glasgow.

Wolfgang Wahlster. 2003. SmartKom: Symmetric multimodality in an adaptive an reusable dialog shell. In *Proceedings of the Human Computer Interaction Status Conference*, Berlin, Germany.

Marilyn A. Walker, Candace A. Kamm, and Diane J. Litman. 2000. Towards developing general model of usability with PARADISE. *Natural Language Engeneering*, 6.

Appendix 1

C	V_i^S	b_1^i	b_2^i	b_3^i	b_4^i
altstadt	7	42.86	0	0	57.14
am	29	24.14	24.14	0	51.72
an	27	0	7.41	0	92.59
auf	43	0	6.98	9.30	83.72
aus	27	0	14.81	0	85.19
bin	37	56.76	10.81	0	32.43
bis	10	40.00	50.00	0	10.00
ein	87	0	1.15	0	98.85
ersten	7	0	14.29	0	85.71
geben	15	0	93.33	0	6.67
gibt	111	91.89	0	0	8.11
kirche	8	12.50	0	62.50	25.00
htte	25	0	92.00	0	8.00
in	214	70.56	12.15	0	17.29
ins	74	94.59	0	0	5.41
is	177	22.03	0.56	0	77.40
ist	55	18.18	1.81	0	80.00
kann	98	0	69.39	0	30.61
kino	226	39.82	44.69	0	15.49
kirche	6	16.67	66.67	0	16.67
kommen	6	50.00	16.67	0	33.33
kommt	31	0	74.19	0	25.81
laufen	16	6.25	81.25	0	12.50
luft	49	0	95.92	0	4.08
mchte	149	97.99	0	0	2.01
nach	38	36.84	23.68	0	39.47
nehmen	12	0	41.67	0	58.33
schlo	61	21.31	27.87	26.23	24.59
schlsser	2	0	50	0	50
sind	28	17.86	0	0	82.14
um	50	76.00	0	0	24.00
vom	26	26.92	0	0	73.08
von	82	2.44	30.49	0	67.07
vor	4	0	50.00	0	50.00
war	14	21.43	0	0	78.57
welch	3	0	33.33	0	66.67
welche	25	0	88.00	0	12.00
will	21	90.48	0	0	9.52
zeig	26	73.08	0	0	26.92
zeige	7	85.71	0	0	14.29
zeigen	27	59.26	14.81	0	25.93
zu	85	0	10.59	0	89.41
zum	55	47.27	0	0	52.72

[2]Horizontal single lines indicate removed markables in horizontal pruning, vertical single lines indicate removed values in vertical and diagonal pruning and horizontal double lines indicate removed markables in diagonal pruning.