Memory-based understanding of user utterances in a spoken dialogue system

van den Bosch, A.

Published in:
Workshop Proceedings of the 6th International Conference on Case-Based Reasoning, August 2005

Publication date:
2005

Memory-based understanding of user utterances in a spoken dialogue system: Effects of feature selection and co-learning

Antal van den Bosch

ILK / Computational Linguistics and AI
Tilburg University
P.O. Box 90153, NL-5000 LE Tilburg, The Netherlands

Abstract. Understanding user utterances in human-computer spoken dialogue systems involves a multi-level pragmatic-semantic analysis of noisy natural language input streams. These analyses are heavily dependent on the dialogue context, and are complex due to the inherent ambiguity of language use, and to the errors induced by the intermediate speech recognition system. We review work on applying k-nearest-neighbour classification to this multi-level task split into (1) dialogue act classification (2) slot filling identification, and (3) communication problem signalling, showing that co-learning some of these tasks produces superior results over learning them in isolation. We also show that additional feature selection can produce succinct feature sets, illustrating the viability of simple keyword-based shallow understanding.

1 Introduction

Spoken dialogue systems (SDSs) are developed typically to assist people at controlling devices and at accessing various computer-based services, in situations in which speech is the preferred or the only possible medium of interaction. Interaction in SDSs proceeds typically via pairs of spoken system and user turns. Interpreting user turns takes considerable effort, involving at least three hurdles. First, speech tends to be looser, more disfluent, more elliptic, and more agrammatical than written or typed input, and linguistic analyses need to be robust enough to handle this. Second, automatic speech recognition (ASR) technology is error-prone, particularly in speaker-independent systems and in situations with background noise. A third hurdle is that in a goal-oriented dialogue a user turn is typically a concise utterance that conveys several messages simultaneously, and not all messages are explicitly voiced [1].

The automatic understanding of user utterances may therefore seem a complex task, ultimately involving full natural language understanding. On the other hand, domain-specific goal-oriented human-machine dialogue systems, such as transport information systems for flights or train trips, usually involve limited domains of conversation, with limited vocabulary and limited types of utterances, needed to arrive at the desired goal. Often, keyword spotting in user utterances

is enough to 'understand' what a user is conveying, since these domain-specific systems tend to give their users little freedom in generating answers, by taking the initiative, posing simple unambiguous questions, and assuming the users are cooperative. Yet, even cooperative users sometimes say unexpected things, and speech recognition errors may additionally interrupt the system's plan at any time, forcing both conversation partners to engage in sub-dialogues to correct the miscommunication.

In recent years there has been an increased interest in using statistical and machine learning approaches for the interpretation of user utterances in projects developing spoken dialogue systems. Dialogue act classification, i.e. to determine what the underlying intention of an utterance is (e.g., 'request', 'confirm', 'reject', etc.) is an example for which these data-driven approaches have been relatively successful [2–4]. The automatic detection of communication problems is another interpretation task to which machine learning approaches have been applied. Given the frequent occurrences of communication problems between users and systems due to misrecognitions, erroneous linguistic processing, incorrect assumptions, and the like, it is important to detect miscommunications in the interaction as soon as possible [5]. Earlier work reports that users signal communication problems when they become aware of them, and that it is possible to detect this automatically to some extent [6, 7].

As a toolkit, machine learning (in this study, the k-nearest neighbor classifier) offers more than a means to study the learnability of one subtask such as dialogue act classification or the detection of communication problem awareness. In this paper we extend the usage of the machine learning toolkit in two directions:

1. Separate classification tasks in understanding user utterances, such as dialogue act classification, communication problem detection, and semantic content analysis of the utterance, can be joined into combined classification tasks. To the machine learner a combined task is simply a task with more classes; if there is any intrinsic relation between the combined subtasks their joint task might be equally well learnable as, or better than, the individual tasks.
2. User utterances can be represented by many different features, such as the recognized words and measurements on the recorded sound waves, but also the system's prompts may be used as predictive features to analyze the utterance. Automatic feature selection can be of help in narrowing a larger set of potentially relevant features down to just the most essential ones, possibly with better generalization performance.

This paper is structured as follows. The next section describes our data, showing how our SDS corpus was turned into training examples for machine learning, and how the pragma-semantic processing tasks were formulated; it also describes the classifier employed in the study and the way we evaluate its performance. The experimental results are presented in Section 3. We discuss our findings and draw conclusions in Section 4.

2 Experimental setup: Data, algorithms, evaluation

2.1 Data collection

The research material on which this study is based is collected from interactions with the OVIS dialogue system [8]. OVIS, for 'Openbaar Vervoer Informatie Systeem' (Public Transport Information System), prompts the user for four slot values in order to retrieve trip information from an on-line database of train timetables: the departure and arrival station names, as well as the date and time (i.e., hour and minute, and time of the day) of either the departure or the arrival. During the development of OVIS a corpus of dialogue transcripts with syntactic and semantic annotations was compiled for evaluation purposes, consisting of transcribed dialogues sampled from a range of telephone calls of naive users of OVIS. Figure 1 is an illustration of the type of dialogues contained in this corpus, translated to English. The composite codes associated to each user utterance (delimited by '_' and ';' when an utterance has two dialogue acts) represent the components of interest in this paper, and are explained later.

Turn	Utterance	Annotation
S1	Good evening. This is the automatic information system of public transportation. (...) From which station to which station do you want to travel?	Q_VA
U1	I need to go from Schiphol to Nijmegen on Tuesday next week.	S_VAD_OK
S2	From where to where would you like to travel on Tuesday twelve December?	Q_VA;I_D
U2	From Schiphol to Nijmegen.	S_VA_PROB
S3	At what time do you want to travel from Schiphol to Nijmegen?	Q_H;I_VA
U3	Around quarter past eleven in the evening.	S_TH_OK
S4	So you want to leave around eleven thirty-eight in the morning?	E_TH
U4	No, in the evening.	N;S_T_PROB

Fig. 1. Translation of the first three turn pairs of dialogue nr. 004/005 sampled from the OVIS corpus. User turns are annotated by our label set on the three components of the understanding task: TRA, SLOT, and PR, respectively. System prompts are analysed in terms of system TRAs and SLOTs.

The material used in the present study consists of 441 transcribed dialogues coming from interactions with 80 naive users who communicated with the system over regular phone lines. Audio files containing the recorded user input, the transcribed, semantically annotated user utterances, as well as the ASR output containing word graphs were available to us for research. The dialogues amount to 3,738 pairs of system and user turns. The number of turn pairs in the dialogues ranges from 2 to 29, the average number being 8. The number of words uttered in a user turn ranges from 1 to 15, the average being three, while the ASR on average recognizes about 7 words per utterance (including alternatives). 43.2%

of the turns are conceptually inaccurately recognised by the system, whereas the word error rate is reported to vary between 8-26% depending on the phone models and language models used in speech recognition [8].

2.2 Class label design

The three components of the pragma-semantic analysis of each user turn are labelled in terms of three sets of labels. The labelling is based on two earlier annotations of the OVIS corpus by [9] and [7]. First, the task-related act (TRA) interpretation component is defined by a label set representing basic actions that a user performs in information-seeking dialogues. The following five labels are sufficient to represent the TRAs in the OVIS corpus:

- S ('slot-filling'), provide information with respect to the query (e.g. 'from Amsterdam to Tilburg')
- Y, give an answer that expresses affirmative input in the given dialogue context (e.g. 'yes', 'that's right', etc.)
- N, give a negative answer (e.g. 'no thanks', 'it's not necessary', 'go back', 'this is incorrect', etc.)
- A, accept incorrectly verified information (e.g., by not signalling a system error)
- NSTD, give a non-standard reply (e.g., to remain silent, to provide a fully irrelevant input).

The SLOT interpretation component is defined by task-related information units for which information is supplied by the user:

- V ('vertrek', departure station)
- A ('aankomst', destination station)
- D (day_of_travel)
- T (time_of_day_of_travel, i.e., morning, afternoon, evening)
- H (hour_and_minute_of_travel).
- VOID (in case no slots are treated in the turn)

In the backward-pointing communication problem detection level we label user turns as PROB, when it identifies the point at which the user becomes aware of the communication problem, since he or she has just heard a system prompt not in accordance with information provided in earlier exchanges in the dialogue. The label OK is used to annotate cases when no communication problems occur.

2.3 Feature design

Most of the cues we utilise for understanding user utterances have shown their worth in earlier work; we simply collect all of these features and use all of them for the classification of all three components. Table 1 lists the employed features according to their origin: whether they come directly from the system's dialogue manager (DM) or the speech recogniser, or whether they come from prosodic processing of the audio recording of the user input made by the ASR. The resulting vector of each user turn has a total of 2,482 features. For more information, cf. [10].

Table 1. Overview of the employed features.

Aspect	Feature
DM: prompt	▷ sequence of last 10 prompt types
DM: lexical	▷ 934-bits bag-of-words vector of current and previous prompt
ASR: confidence	▷ highest summed confidence score in current word graph
	▷ same, normalised by number of nodes in path
	▷ score difference between most confident and second-most confident path in current word graph
ASR: branching	▷ branching factor in the word graph of current and previous utterance
ASR: lexical	▷ 1,518-bits bag-of-words vector of current and previous user turn
	▷ word string in most confident path in current word graph
	▷ length of most confident string
Audio: pitch	▷ maximum, minimum, mean pitch, and standard deviation
	▷ position of maximum and minimum pitch
Audio: loudness	▷ maximum and mean energy, and standard deviation
	▷ position of maximal energy
Audio: duration	▷ duration of turn
	▷ duration of initial pause
Audio: speech rate	▷ tempo

2.4 Classifier

We use a memory-based learner MBL as the machine-learning classifier of choice. The MBL algorithm is a descendant of the k-nearest neighbour (k-NN) classifier [11, 12]. Memory-based, or instance-based learning is a type of 'lazy' learning; the classifier simply stores a representation of all training examples in memory, without abstracting away from individual instances during the learning process. The classification procedure of MBL[1] finds, for each new test instance, k nearest-neighbor examples from memory, and subsequently extrapolates the nearest neighbors' majority class to the new instance. MBL computes the distance between a memory example X and the new unlabelled instance Y for each feature according to a distance function $\Delta(X, Y)$, that computes the distance of X and Y as the sum of the differences between the individual corresponding features of X and Y, as in $\Delta(X, Y) = \sum_{i=1}^{n} w_i \, \delta(x_i, y_i)$, where n is the number of features, and w_i is an additional optional feature weight of the ith feature.

Different functions can be used for δ, such as overlap or Hamming distance, modified value difference, and Jeffrey divergence. After the k nearest neighbours are identified with the distance function, their majority class label is transferred as the predicted class to the new instance Y. What constitutes the majority class

[1] In our experiments we used the implementation of MBL called IB1 from the TiMBL software package [13], version 5.1.

of a set of k nearest neighbours can optionally be determined by unweighted democratic voting, or by various distance-weighted voting schemes. For details on all parameters, functions, and metrics in the MBL algorithm, cf. [13].

It is unknown which (combination of) parameter settings yield the best generalisation performance on some task. Ideally, one would want to tune algorithm parameters automatically: such a procedure however contains a search problem for finding optimal parameter settings given a particular data set. The method of wrapped progressive sampling or WPS [14] offers a solution to this. The procedure implemented in WPS combines finding a set of optimised algorithmic parameters for a range of machine learning algorithms through classifier wrapping combined with progressive sampling of training data. For more details, cf. [14]. We applied WPS to all experiments reported in this paper.

The performance of the learner on the various tasks is measured in terms of F-score [15] computed per class, and micro-averaged over classes, averaged over 10-fold cross-validation experiments. For the cross-validation experiments the data was partitioned at the level of dialogues, not labeled examples.

3 Experimental results

Table 2 displays the outcomes of our systematic series of 10-fold cross-validation experiments on each of the three tasks, the three combinations of these tasks in pairs ('Double task' in the table), and the single combination of all three tasks ('Triple task'). Under the columns headed by '−' we observe that learning the tasks in isolation is not necessarily the best scenario. Although it is for the TRA task, the SLOT task is learned better when co-learned with the TRA task, and in the case of the PR task we even observe an optimal result when it is co-learned with the other two tasks. The values of the best results (91.7, 87.7, and 90.8, respectively) show that all three components can be learned at fairly reliable levels. This is attested also by comparing them to the baseline score listed in the second column of Table 2. This baseline is quite sharp and informed: it represents the F-score of guessing the TRA, SLOT, and PR labels based on the previously asked question by the system. If users would be totally predictable in the way they answer questions, and if their answers would be recognized perfectly, this score would be 100. The present best results reduce the remaining error left by the baseline by one third to a half.

Next, we performed the same series of experiments, introducing automatic feature selection before learning and testing on a cross-validated training and test set. Automatic feature selection, typically based on some heuristic search for the fittest feature subset by running wrapping experiments on the training data, produces a subset of features estimated to perform well on unseen data; preferrably at least as good as, and smaller than the original set of features. We adopted a simple yet powerful bi-directional hill-climbing feature selection method proposed by Caruana and Freitag [16], feeding it with the empty set as starting point. The generalization performance results of these experiments are displayed in Table 2 in the columns headed by '+'. Compared to the experiments

Table 2. Average F-scores (scaled to 100) on the three basic tasks, performed in isolation, as part of a double task, and as part of the triple task, without feature selection (−) and with feature selection (+), with standard deviations printed below each F-score.

		Individual			Double task		Triple task	
Task	Baseline	−	+	Task	−	+	−	+
TRA	78.7	91.7	91.5	+SLOT	91.3	90.3	90.7	91.1
	2.5	0.7	1.1		0.6	1.3	1.2	1.6
				+PR	91.6	90.4		
					0.9	1.0		
SLOT	77.8	86.7	84.8	+TRA	87.7	84.6	86.6	85.1
	2.2	2.0	1.9		2.0	2.4	2.3	2.0
				+PR	86.5	84.5		
					1.8	1.3		
PR	81.3	87.2	88.2	+TRA	89.7	89.3	90.8	89.0
	3.9	2.3	2.0		1.4	1.0	1.2	1.2
				+SLOT	89.9	88.8		
					1.5	1.5		

without feature selection, we observe mild decreases in performance, especially on the SLOT task and on several 'Double' tasks in which one point of F-score is lost, while we also observe smaller decreases and some increases in performance on the TRA and PR tasks.

These results are all the more surprising given that they are based on reductions of the original set of 2,482 features down to just 8 to 18 on average (a compression rate of over 99%), as displayed in Table 3. For example, for the TRA task just over 8 features on average are needed to attain virtually the same result as using all 2,482 features. A few more features are needed for the other tasks and task combinations, especially those involving the SLOT task, which involves on average about nine lexical features that are triggered by the recognition of words by the ASR.

A second remarkable fact is that most tasks involve small amounts of features from all types; apart from specific words from the user utterance and system prompt, typically one feature representing some aspect of the dialogue history is selected, and one feature of the recorded audio of the user utterance. Table 4 lists the top-18 most frequently selected features over all feature selection experiments; the single-most frequently selected feature (selected in all experiments) is the type of question the system most recently asked, which is also the single feature on which the baseline of Table 2 is based, and arguably the most important feature of all [17]. The single audio feature is typically either the length of the utterance measured in the number of words, or in seconds. It can further be seen in Table 4 that the list of most-selected features represents a succinct, minimal vocabulary that covers the most essential words in train travel from both parties.

Table 3. Average numbers of selected features for all seven tasks, divided over sources and types of features, and totalled.

	# Features				
	System		User		
Task	History	Words	Audio	Words	Total
TRA	1.3	2.5	2.0	2.3	8.1
SLOT	1.0	3.4	1.4	9.0	14.6
PR	1.1	5.4	0.1	4.7	11.3
TRA+SLOT	1.1	7.1	0.8	9.5	18.5
TRA+PR	1.5	3.3	1.2	3.3	9.2
SLOT+PR	1.1	6.0	1.0	8.8	16.9
TRA+SLOT+PR	1.3	6.7	1.0	9.3	18.3

Table 4. The eighteen most frequently selected features, grouped by source and type.

Source	Type	Selected features
System	History	Previous question
	Words	*waar* (where), *welk* (which), *naar* (to)
User	Audio	Utterance length (# words in recognized string, seconds)
	Words	Complete recognized string, *nee* (no), *ja* (yes), *aankomen* (arrive), *'s avonds* (in the evening), *'s middags* (in the afternoon), *'s ochtends* (in the morning), *morgenochtend* (tomorrow morning), *uur* (o'clock), *van* (from), *naar* (to), *in* (in).

4 Conclusions

Based on a case study we have explored the learnability of three aspects of understanding user utterances in goal-oriented spoken dialogue systems, and especially their co-learnability as combined learning tasks. The case study indeed showed that co-learning is possible, and even warranted to arrive at the optimal scores for two of the three tasks. In both of these cases, the SLOT and PR tasks, it can be argued that the TRA task (which is best learnt in isolation) is in fact the 'supertask' of which SLOT and PR are subtasks. SLOT is the hierarchically lower subtask of detecting particular slots under the 'slot-filling' act of TRA. PR, the task of detecting the user's awareness of communication problems, can perhaps be seen as yet another main type of act as part of TRA.

Second, we observed by performing feature selection experiments that only a small subset of 8 to 18 features is sufficient to attain near-top performance. The 18 most frequently selected features tell the story about how the tasks should be performed: first, the 'baseline' outcome is determined by the question just asked by the system. In case the user does not respond as expected, possibly signalling

a communication problem, this can be detected by checking the values of the other features: for example, the length of the user's utterance (e.g. if a 10-word answer is returned to a yes/no question, something unexpected is happening), and domain-specific 'hot' words that tell whether the user is talking about times or places, possibly to correct earlier misrecognized information that the system needs to backtrack to, but has assumed to be recognized correctly so far.

The strength of the top-18 features is reflected in Table 5 which provides also the upper bound of the system in case of perfect ASR performance; as said, the OVIS data also has a transcribed version of all user utterances besides the audio recordings. If the top-18 features would be used to learn the task we would arrive at performances which are within standard deviation range of the upper bound results.

Table 5. F-scores on the three tasks as parts of the triple task with all features (left), the top-18 most frequently selected features (middle), and the upper bound score on each task given transcribed user utterances (right).

Task	All features	Top 18 features	Upper bound
TRA	91.7 ± 0.7	92.3 ± 1.1	93.3 ± 0.8
SLOT	86.7 ± 2.0	88.4 ± 2.0	90.8 ± 1.2
PR	87.3 ± 2.3	89.2 ± 1.7	91.8 ± 1.1

This paper extends earlier results [7, 18, 19, 17, 10] on the same corpus; it reaffirms the findings reported in this series of papers and expands them with the new feature selection results. Similar results have already been attained by applying rule learning to the same data, yielding comparable feature sets and performances for the individual tasks, but worse performances for the co-learning tasks. We view the co-learning results as a recommendation to test the co-learnability of subtasks with memory-based learning in any domain, whenever there is reason to believe they are related tasks and could be co-learned.

As for dialogue systems, we believe our results with the k-nearest neighbor approach using small amounts of features reaffirms the simple power of classic keyword-based methods that have been superseded by research into the integration of deeper syntactic, semantic, and pragmatic analyses. This approach has yet to demonstrate its robustness to noisy environments, which the present approach clearly has – it will arguably be the best option as long as no truly robust general-purpose syntactic and semantic parsers exist.

Acknowledgements

This research was funded by NWO, the Netherlands Organisation for Scientific Research. This paper is largely based on earlier work performed in collaboration with Piroska Lendvai, Emiel Krahmer, and Marc Swerts, to whom the author is indebted.

References

1. Krahmer, E., Swerts, M., Theune, M., Weegels, M.: Error detection in spoken human-machine interaction. International Journal of Speech Technology **4** (2001) 19–30
2. Reithinger, N., Maier, E.: Utilizing statistical dialogue act processing in Verbmobil. In: Proc. of ACL. (1995)
3. Choi, W., Cho, J., Sea, J.: Analysis system of speech acts and discourse structures using maximum entropy model. In: Proc. of ACL. (1999)
4. Stolcke, A., Ries, K., Coccaro, N., Shriberg, E., Bates, R., Jurafsky, D., Taylor, P., Martin, R., Ess-Dykema, C.V., Meteer, M.: Dialogue act modeling for automatic tagging and recognition of conversational speech. Computational Linguistics **26** (2000) 339–373
5. Walker, M., Langkilde, I., Wright, J., Gorin, A., Litman, D.: Learning to predict problematic situations in a spoken dialogue system: Experiments with How may I help you? In: Proc. of ACL. (2000)
6. Hirschberg, J., Litman, D., Swerts, M.: Identifying user corrections automatically in spoken dialogue systems. In: Proc. of NAACL. (2001)
7. Van den Bosch, A., Krahmer, E., Swerts, M.: Detecting problematic turns in human-machine interactions: Rule-induction versus memory-based learning approaches. In: Proc. of ACL. (2001) 499–506
8. Strik, H., Russel, A., van den Heuvel, H., Cucchiarini, C., Boves, L.: A spoken dialog system for the Dutch public transport information service. Int. Journal of Speech Technology **2** (1997) 121–131
9. Veldhuijzen van Zanten, G.: Semantics of update expressions. Technical report, IPO, Eindhoven University (1996)
10. Lendvai, P.: Extracting Information from Spoken User Input. A Machine Learning Approach. PhD thesis, Tilburg University (2004)
11. Cover, T., Hart, P.: Nearest neighbor pattern classification. Institute of Electrical and Electronics Engineers Transactions on Information Theory **13** (1967) 21–27
12. Aha, D., Kibler, D., Albert, M.: Instance-based learning algorithms. Machine Learning **6** (1991) 37–66
13. Daelemans, W., Zavrel, J., van der Sloot, K., van den Bosch, A.: TiMBL: Tilburg Memory Based Learner, version 5.1, Reference guide. ILK Technical Report 04-02, Tilburg University (2004) Available from http://ilk.uvt.nl.
14. Van den Bosch, A.: Wrapped progressive sampling search for optimizing learning algorithm parameters. In: Proc. of 16th Belgium-Netherlands Conference on Artificial Intelligence. (2004) 219–228
15. Van Rijsbergen, C.: Information Retrieval. Buttersworth, London (1979)
16. Caruana, R., Freitag, D.: Greedy attribute selection. In: Proceedings of the 11th International Conference on Machine Learning. (1994) 28–36
17. Lendvai, P., van den Bosch, A., Krahmer, E.: Machine learning for shallow interpretation of user utterances in spoken dialogue systems. In: Proc. of EACL Workshop on Dialogue Systems: Interaction, adaptation and styles of management. (2003) 69–78
18. Lendvai, P., Van den Bosch, A., Krahmer, E., Swerts, M.: Multi-feature error detection in spoken dialogue systems. In: Proc. Computational Linguistics in the Netherlands (CLIN '01), Rodopi Amsterdam (2002)
19. Lendvai, P.: Learning to identify fragmented words in spoken discourse. In: Proc. of EACL Student Research Workshop. (2003) 25–32