002

TR-IT-0258

Speech-Act Recognition for Dialogue Processing

J. Joachim Quantz

1998年3月

Abstract

This report describes a statistical method for assigning speech acts to dialogue utterances. The method is based on a corpus of dialogues labeled with speech-act information. For an annotated corpus of 80 dialogs from the hotel-reservation domain we obtain an accuracy of 85-90% when using 70 dialogs for training and 10 for evaluation.

Speech-Act Recognition for Dialogue Processing

J. Joachim Quantz

Abstract

In this paper we present a data-based method for speech-act recognition. The basic idea is to use a corpus of annotated dialogues, in which utterances are labeled with speech acts, to learn correlations between words (or word sequences) and speech acts. We discuss various ways of enhancing a simple word-based recognition method and evaluate these methods by cross validation (i.e. by using m dialogues for training and n for evaluation). For 80 annotated dialogues taken from the ATR hotel-reservation corpus we obtain recognition rates of 85–90%.

1 Introduction

In this paper we present a data-based method for speech-act recognition in dialogue processing. In Section 2 we briefly describe the task of dialogue modeling, i.e. the interpretation of utterances wrt. their context in the overall dialogue. We propose to distinguish three subtasks of dialogue modeling, namely

1. determining the *communicative function* of utterances (and of turns);
2. determining the *propositional content* of utterances and of turns;
3. determining the *dialogue structure*, i.e. the relationships obtaining between utterances and turns.

In Section 3 we argue that speech acts (or dialogue acts, communicative acts, or illocutionary force types, as they are also called in the literature) are appropriate for representing the communicative function of utterances. Speech-act recognition thus means to determine the communicative function of utterances and is therefore an important sub-task in dialogue modeling.

In Section 4 we present a data-based method for speech-act recognition. The basic idea is to use a corpus of annotated dialogues, in which utterances are labeled with speech acts, to learn correlations between words (or word sequences) and speech acts. We discuss various ways of enhancing a simple word-based recognition method and evaluate these methods by cross validation (i.e. by using m dialogues for training and n for evaluation). For a corpus of 80 annotated dialogues we obtain recognition rates of 85–90% when using 70 dialogues for training and 10 for evaluation.

In Section 5 we discuss possibilities to continue work on speech-act recognition and finally give a conclusion in Section 6.

2 Dialogue Modeling

In this section we will briefly sketch our notion of *dialogue modeling*. We begin by pointing out its general features and its relevance for dialogue processing and then describe some of its subtasks in more detail. In doing so, we try to be as general as possible, i.e. we do not focus on any specific application scenario, but rather address the problem of dialogue processing in general. However, we will sometimes use concrete examples from speech translation or interactive dialogue systems for illustration.

2.1 Motivation for Dialogue Modeling

Dialogue Modeling is concerned with what is traditionally called *contextual* or *pragmatic* information. Thus instead of treating an utterance as an isolated unit, Dialogue Modeling aims at understanding it, given the background of the utterance's context in the overall dialogue. One can distinguish at least two aspects of such a context:

1. The context in which the dialogue takes place. Since most dialogue processing is currently restricted to a particular domain and/or scenario this context is usually fixed in advance and does not change during dialogue processing.

2. The context created by the dialogue itself. One of the tasks of dialogue modeling is to build up this context while processing the dialogue.

Note that the impact of the first aspect is not restricted to dialogue modeling, but also concerns other components in dialogue processing, such as speech recognition or transfer. This can be either implicitly, e.g. by a corpus used to train algorithms, or explicitly, e.g. when writing a transfer lexicon.

Thus it is in particular the second aspect of contextual information which is characteristic for dialogue modeling, namely dynamically building up a representation of the ongoing dialogue and providing relevant contextual information to other components. Which particular components this could be depends on the specific application of dialogue processing. The following is a rather coarse-grained distinction between three major components which should be applicable to all dialogue processing systems:

Analysis is concerned with transforming the acoustic input into a format that can be processed by 'Processing'. It thus comprises at least a component for *speech recognition*, which maps the acoustic input into strings. Depending on the particular application it might also comprise a *syntactic and/or a semantic analysis*.

Processing is concerned with processing the output of 'Analysis' and providing the input for 'Generation'. This component is highly application-dependent, e.g. it would be a transfer module in automatic dialogue interpreting or a database (plus query interface) in interactive dialogue systems.

Generation is concerned with producing a spoken utterance/turn from the output of 'Processing'. This comprises at least a module for synthesis, but can be more complex, depending on the output format produced by 'Processing'.

We will now briefly sketch how dialogue modeling can support Analysis and Processing.[1]

Dialogue Modeling and Analysis

The main task of the Analysis component is to map the acoustic input signal into a linguistic representation of the spoken utterance (e.g. a string, a syntactic parse tree, a semantic representation). And the main problem is to find the *correct* representation among the various possible representations. This is basically achieved as follows:

1. The various acoustic segments occurring in the input signal are analyzed and mapped into words. This mapping is probabilistic, i.e. an acoustic segment corresponds to a word with a certain probability. The result is a so-called *word lattice* in which the nodes are (logical) time points and the edges are pairs of words and their respective probabilities. Each path in such a lattice represents a possible analysis of the input signal (i.e. a sequence of words) and its respective probability.

2. The acoustic probabilities are combined with a-priori probabilities stemming from a *language model.* Such a language model is derived from a training corpus and says how likely a word or a sequence of n words (so called ngrams) is wrt the training corpus.

[1] It should be obvious that the boundaries between 'Processing' and the other two components are fuzzy. Syntactic and semantic analysis, for example, could be part of either 'Analysis' or of 'Processing'. For the purpose of our argumentation, however, we only need the general distinction between these modules and the actual boundaries are not relevant.

3. Additional criteria, e.g. syntactic well-formedness, can be used to further score or even filter the different hypotheses, i.e. sequences of words assigned to the input signal.

Dialogue modeling could support this process by providing "high-level" criteria for scoring the hypotheses. Though this sounds straightforward, it is not that easy to realize. The main problem is to find an appropriate *architecture* for integrating DM knowledge, i.e. knowledge from dialogue modeling, *efficiently* into the analysis process.

There are basically two different architectures for doing so:

Integrated Architecture: DM knowledge is used *together* with the language model when building the word lattice. The advantage of this approach is that hypotheses "inconsistent" with DM knowledge are "eliminated" as early as possible. However, such an integration is a non-trivial task and it is not clear what impact such an integration will have on the runtime performance.

Sequential Architecture: In a sequential architecture DM knowledge is used after the word lattice has been constructed on the basis of acoustic and language-model probabilities. From a software engineering point of view, this is preferable, since word lattices are a well defined interface format and the speech-recognition module would not be affected at all by the integration of DM knowledge. The drawback is that the speech recognition possibly generates lots of "useless" hypotheses which might even eliminate the "correct" hypotheses.

Two subtypes of a sequential architecture are possible:

Postprocessing/Reordering: DM knowledge is used to post-process the word lattice, e.g. by rescoring the different paths in the lattice. Since these lattices can be very big, this would require efficient scoring and search mechanisms. A slightly less ambitious approach is to take the N best paths and reorder them wrt DM knowledge.

Filtering/Backtracking: DM knowledge is used to check the best path: if it is consistent with DM knowledge it is taken, otherwise it is rejected and the next-best path is backtracked until a path consistent with DM knowledge is found.

Since it is not yet clear to what extent DM knowledge is actually useful for Analysis, a sequential architecture seems currently more reasonable than an integrated one. If it can be shown that Analysis benefits from an additional DM-based processing, one can then investigate whether a "real" integration might be preferable and if so, how to realize it.

It should also be noted, that there already has been proposed a rather efficient way of integrating DM knowledge into the analysis process, namely the use of *dialogue-dependent language models.* The idea is to use different language models for different dialogue contexts. Thus instead of generating a single language model from the training corpus, the training corpus is annotated with DM information such that several language models can be computed. During dialogue processing, DM knowledge is then used to pick a particular language model for analyzing the next utterance. Work along these lines is presented in [Popovici, Baggia 96].

Dialogue Modeling and Processing

Whereas it is not too difficult to describe the general characteristics of analysis without recurrence to a particular application, this is much more difficult for the central processing module. On a very abstract level, we might say that processing maps the representation of the user input into a representation of the system output. In speech translation, this mapping could consists in transforming a source-language representation into a target-language representation. In an interactive dialogue system, the mapping could involve the transformation of the input representation into a database query whose answer would then be transformed into an output representation; or it could involve the transformation of the input into a database update and the generation of a follow-up query to the user.

Again, there seem to be two major possibilities for integrating DM knowledge with 'Processing':

Pre-Processing: DM knowledge is added to the information passed from analysis to processing.

If Needed: Whenever 'Processing' needs DM knowledge, it poses a query to the dialogue module.

2.2 Subtasks of Dialogue Modeling

The considerations so far have been rather abstract and were more concerned with the general question of *how to integrate* DM knowledge into a dialogue-processing system. We will now consider *what specific knowledge* can be provided by DM, i.e. we will sketch the most important subtasks of dialogue modeling. In doing so, we will adopt the following terminological distinction betwen *turns* and *utterances*:

Turns are the entities from which a dialogue is built. In VERBMOBIL, a turn begins when the speaker pushes the start button and ends when she releases it. Given this scenario, it would thus be possible to have two consecutive turns spoken by the same speaker. An alternative would be to begin a new turn whenever the speaker changes.

What is important to realize in any case is that a turn can be rather long and can consist of more than one utterance.

Utterances are the entities obtained when segmenting a turn. Such a segmentation can be based on prosodic, syntactic, or pragmatic criteria and consequently 'utterance' is even less well defined then 'turn' (see Section 3.2).

We will basically distinguish three subtasks of domain modeling:

1. determining the *communicative function* of utterances (and of turns);
2. determining the *propositional content* of utterances and of turns;
3. determining the *dialogue structure*, i.e. the relationships obtaining between utterances and turns.

In the remainder of this section we will illustrate these subtasks and their relevance for dialogue processing.

An important characteristic of dialogues is that the utterances occurring in them differ considerably wrt to their *communicative function*:

(1) Hello

(2) Please wait a moment, please

(3) Would that be okay?

(4) and the price of a single would be ninety dollars

(5) oh, oh

(6) Thank you

(7) I am sorry

(8) How much will the cancellation charge be?

Section 4 introduces the notion of *speech acts* to represent the communicative functions performed by utterances and shows how speech acts can be automatically recognized. Knowing the speech act of an utterance is especially useful for processing and may also be useful for generation. There are also possibilities to use speech-act recognition (and prediction) to support analysis.

Whereas some of the utterances occurring in dialogues seem to have only a communicative function, most also have a *(propositional) content*. Given the example utterances above, we would thus say that (1) has the communicative function of greeting, but has no content; (8), on the other hand, has the communicative function of requesting information and its content is 'cancellation charge'.

Knowing the communicative function of an utterance and its content could thus be identified with understanding it.

The subtasks sketched so far have been concerned with determining properties of individual utterances. The representation of *dialogue structure*, on the other hand, models the relations obtaining between the individual utterances occurring in a dialogue. This involves, on the one hand, relations obtaining between utterances occurring in the same turn; on the other hand, it concernes relations obtaining between utterances occurring in different turns. A good example for the latter are a request for information by one dialogue partner and the provision of the respective information by the other partner:

(9)

a. (...)

b. And how much will the cancellation charge be?

(10)

a. I'll just check that for you, Miss Suzuki.

b. Just a moment, please.

c. Okay,

d. you had reserved a one hundred and eighty-six dollar single room.

e. That would make the cancellation charge ninety-three dollars.

3 Speech Acts

In this section we introduce our notion of speech acts. We begin by discussing different approaches such as communicative acts, dialogue acts, and illocutionary force types. In Section 3.2 we then describe speech-act labeling and some of the problems related to it.

3.1 Dialogue-oriented Speech Acts

In recent years there has been considerable research in Dialogue Processing concerning the notion of *Speech Acts*. Since many approaches show considerable differences to the original speech-act theory as developed in [Austin 62, Searle 69], they often employ different technical terms as well. These terms comprise, for example, *illocutionary force types*, *dialogue acts* [Jekat et al. 95], or *communicative acts* [Seligman et al. 94].

The basic idea underlying these approaches is that utterances in a dialogue usually fulfill a specific *communicative function* and that representing this function explicitly is useful for several purposes.

Thus the speech act associated with an utterance abstracts over the particular linguistic realization used by the speaker, as well as over (parts of) the semantic content contained in the utterance. In this sense, speech acts belong to the level of *pragmatic analysis*.[2]

The most obvious problem in research on speech acts for dialogue processing consists in establishing the exact set of speech acts to be used. Note that there are several dimensions of this problem. For one thing, one has to decide how *domain-dependent* the chosen set of speech acts should be. Comparing the dialogue acts proposed in [Jekat et al. 95] with the communicative acts proposed in [Seligman et al. 94] the former ones seem much more domain-dependent than the latter ones. There are, for example, dialogue acts like 'suggest-support-date' which are suitable for appointment-scheduling dialogues but would not occur in the hotel-reservation domain.

However, the dialogue acts used in VERBMOBIL are modeled in a hierarchy, in which the domain-specific acts are subtypes of more general, domain-independent acts like 'suggest'. Comparing these general dialogue acts with the communicative acts in [Seligman et al. 94] there is a considerable overlap. And the remaining differences are not so much due to different domains, but rather to the different application scenario. In appointment-scheduling, two persons are jointly solving a problem by mutually suggesting, rejecting, commenting on dates. In the hotel-reservation scenario, on the other hand, persons exchange information, i.e. the person making a reservation provides information about her length of stay, about the desired type of room etc, while the other person provides information about the availability and costs of rooms, etc.

We would thus argue for a domain-independent set of speech acts which can then be further subcategorized into domain-dependent speech-acts.

A second important issue concerns the *degree of abstraction*. Consider the following examples

(11) May I ask your credit card number, please?

(12) Please tell me your credit card number.

(13) I would then need your credit card number, please.

[2] This is arguable, however, since the distinction between semantics and pragmatics is not straightforward [Levinson 83].

(14) Could you tell me your credit card number?

In a sense, all these utterances perform the same communicative function, namely informing the hearer, that the speaker wants her to provide certain information. One could therefore argue that all utterances express the same speech act.

Given the set of communicative acts proposed in [Seligman et al. 94], however, (11) is a 'permission-request', whereas (12) is an 'action-request', (13)) is an 'inform' and (14) is a 'yn-question'. This classification thus takes into account the syntactic construction used in the utterance and is therefore less abstract than a classification mapping all utterances to the same speech act.

3.2 Speech-Act Labeling

Dialogues labelled with speech acts are needed for several reasons. For one thing, data-based approaches to speech-act recognition need such dialogues as training data. But even for speech-act recognition which is based on hand-coded rules, labeled dialogues are needed as a basis for knowledge engineers to decide which knowledge is to be used in the rules. Finally, such dialogues are needed for evaluating the accuracy of speech-act recognition.

It should also be noted that labeling dialogues with speech acts is a very good method to test the adequacy of the chosen set of speech acts. In general, a cyclic approach towards fixing the set of speech acts seems to be necessary. I.e. the "initial" set of speech acts should be used for labeling several dialogues, and based on the shortcomings encountered during labeling, the set should then be accordingly modified. It is also a good idea to document the problems encountered in labeling in order to obtain rules of thumbs to resolve uncertainties in labeling.

In this section we assume that this process has been completed, i.e. that the set of speech acts has been permanently fixed. Even in this situation, labeling dialogues poses some non-trivial problems, some of which are related to the degree of freedom discussed above. We will briefly address the issues of *segmentation*, *multiple speech acts* and *depth of interpretation*.

In labeling a dialogue, each utterance is assigned one or more (see below) speech acts. Note, however, that the basic entities from which dialogues are built are *turns*, whereas utterances are not well-defined entitities.[3] Given a turn, there are usually several possibilities to split it into a sequence of utterances, i.e. to perform segmentation. For illustration, consider the following example:

(15) okay a single starting on the tenth and you'd be checking out on the sixteenth is that right

(16)

[3] Even the notion of a turn can become problematic if it is not well-defined in the scenario.

a. okay a single

b. starting on the tenth

c. and you'd be checking out on the sixteenth?

d. is that right?

(17)

a. okay

b. a single, starting on the tenth and you'd be checking out on the sixteenth

c. is that right?

At least the following criteria seem to be relevant for deciding how to segment a turn:

- prosodic information (especially pauses);
- syntactic boundaries (completed clauses or phrases);
- pragmatic units corresponding to meaningful speech acts.

From the point of view of speech-act labeling it seems most reasonable to rely on pragmatic criteria when performing segmentation. Prosodic and syntactic criteria should not be neglected completely, however.

For one thing, there are cases where segmentation is hardly possible without prosodic information. However, speech-act labeling usually takes only *transcripted dialogues* as input material. This is reasonable if one assumes that the prosodic information available in the input signal has been adequately encoded in the transcription, e.g. by means of punctuation.

There is another aspect of segmentation, however, which is of crucial importance, especially in the context of data-based speech-act recognition. Suppose you train your speech-act recognition with data in which turns are segmented according to a specific set of criteria. If in the actual system this module receives as input turns segmented on the basis of different criteria, this will probably have considerable impact on the recognition rate. It is therefore important to take into account the segmentation criteria which will be used in the actual system when segmenting the turns for speech-act labeling.

Once a turn has been segmented into utterances, each utterance is to be labeled with a speech act. Though this should be straightforward in most cases (given the set of speech acts has been designed adequately), there are frequently cases, where one could assign more than one speech act to an utterance. On the one hand, this can be due to the fact that the speech acts used for labeling overlap; on the other hand an utterance may really perform two different speech acts at the same time.

Speech Act	*Utterances*	*Percentage*	*Clerk*	*Customer*
inform	1278	31.7	33.8	29.0
acknowledge	649	16.1	16.4	15.7
temporizer	319	7.9	5.0	11.8
thank	234	5.8	5.9	5.7
desire	198	4.9	**0.2**	**11.2**
wh-question	189	4.7	**6.6**	**2.1**
yn-question	167	4.1	3.5	5.0
yes	167	4.1	**2.0**	**7.0**
action-request	164	4.1	**5.3**	**2.4**
confirmation-question	103	2.6	**4.2**	**0.4**
greet	83	2.1	2.1	2.0
accept	78	1.9	**0.9**	**3.3**
farewell	77	1.9	**2.8**	**0.7**
information-request	70	1.7	**2.6**	**0.6**
apology	57	1.4	**2.1**	**0.5**
offer	42	1.0	**1.8**	**0.0**
believe, promise, permission-request	~30	~0.8		
no	26	0.6	0.6	0.7
suggest	20	0.5	**0.9**	**0.0**
offer-follow-up, thank-response, alert	~4	~0.1		

Figure 1: Frequencies of speech acts in the 80 dialogues labeled

One way to overcome this problem is to address it explicitely in the documentation, i.e. to point out overlapping speech acts and to give criteria for choosing the "correct" one. For example, utterances performing the speech-act 'information-request' can be syntactically realized as yes-no-questions, which in turn are usually labeled with the speech-act 'yn-question':

(18) Could you tell me your credit card number?

In the definition of the respective speech acts it should therefore be settled whether the above utterance is to be labeled as an 'information-request' or a 'yn-question'.

Another possibility is to label such utterances with more than one speech act. This seems to make life easier for dialogue labeling, since it avoids making a decision. However, it somehow just pushes the problem over to the training and evaluation phases. Basically, one has to decide whether an utterance labelled with two acts expresses at least one of these acts or both at the same time. In other words would recognition be correct, if one of the annotated acts were assigned or if both were assigned.

For training and evaluating our own approach to speech-act recognition we labeled 80 dialogues from the ATR corpus of hotel-reservation dialogues. We used the files containing transcriptions of the English turns as input data and proceeded in two steps. First, each turn was manually segmented into utterances. In a second step, each utterances was assigned *one* speech act. The average number of utterances per dialogue is 50 and it took approximately 1 hour to label 10 dialogues.

We only labeled one speech act per utterance, i.e. we resolved all cases of ambiguity during labeling. Table 1 shows for each speech act the number of occurrences in the labeled dialogues and its relative frequency with respect to the total number of utterances, as well as with respect to the utterances spoken by the clerk and the customer, respectively (frequencies which differ considerably between clerk and customer are printed bold).

4 Speech-Act Recognition

There are several criteria on which *speech-act recognition*, i.e. the automatic assignment of a speech act to an utterance, can be based. The most obvious distinction is the one between *micro-structure* and *macro-structure*. On the micro-structural level, an utterance contains linguistic features which indicate the speech act (e.g. certain syntactic constructions like 'May I ...', 'When do you ...', 'I would like'). On the macro-structural level, an utterance occurs in a specific dialogue context which also has impact on the speech act performed by it.

Work on speech acts in dialogue processing has so far focussed on two main task: the one consists in predicting the most likely speech act(s) expected to be performed by the next utterance on the basis of the dialogue context; the other one consists in determining the most likely speech act performed by an utterance, given the utterance and the dialogue context. Both functionalities can benefit from each other:

- when determining the speech act performed by an utterance, the predicted speech act(s) can be used to take into account macro-structural information;
- when predicting the speech act performed by the next utterance one has to take into account the speech acts performed by the current utterance and its predecessors.

In the VERBMOBIL project, two approaches to speech-act recognition have been implemented. One was part of the "flat analysis" and used strings as input format [Mast et al. 95], the other was part of the "deep analysis" and used semantic representations as input [Schmitz, Quantz 95].

The approach taken in [Schmitz, Quantz 95] has two major drawbacks:

1. The weighted defaults on which the recognition is based have to be encoded by hand. Since the authors admit that their approach is highly domain-

dependent, it means that costly knowledge engineering has to be performed when the domain is changed.

2. The data format used as input is highly complex and has to be constructed by parsing a string syntactically and then building up a semantic representation. As a consequence, training or evaluation dialogues can only be processed if they are processed by the syntactic and the semantic model. Moreover, changes in the syntactic or semantic formalism may necessitate corresponding changes in the speech-act recognition.

Both drawbacks would be less severe, if the recognition rate of this "deep" approach would have been substantially higher than for the "flat" approach. However, this was not the case.

We therefore decided to try out a purely statistical approach to speech-act recognition, based on strings as input and annotated training dialogues. In the remainder of this section we will present several possibilities to implement such a statistical approach and evaluate their accuracy.

The most naive approach to statistical speech-act recognition could be based on the frequencies of the speech acts shown in Figure 1 by using them as probabilities. I.e. an utterance which we know nothing about expresses the speech act 'inform' with probability 0.32, the speech act 'acknowledge' with probability 0.16 and so on. Thus for each utterance the most probable speech act would be 'inform' and if we assigned this speech act to every utterance we could expect a hit rate of 32%. Any serious speech-act recognition should perform substantially better than this rather poor rate.

In the next section we will first introduce recognition methods based on micro-structural information, namely on the words occurring in the utterance. In Section 4.2 we use output from a tagger, i.e. tags describing the part-of-speech of the words occurring in the utterances as well as their semantic content for speech-act recognition. Though these methods yield recognition rates below the ones for word-based methods, combining the two is advantageous and increases the overall recognition rate.

In Section 4.3 we then describe recognition methods based on macro-structural information, i.e. on the speech-acts of the preceding dialogue context.

4.1 Using Wordforms

Recognition methods based on micro-structural information exploit the fact that utterances contain linguistic cues as to which speech act they express. Since we want to take simple sequences of words as input for our speech-act recognizer, we cannot use complex syntactic or semantic criteria. The most basic solution is then to compute for each word the frequency with which it occurs in each speech act. From this we can then compute the probability of a speech act given a certain word.

$$(19) \qquad P(s|w) \overset{def}{=} \begin{cases} 0 & \text{if } Occ(w) = 0 \\ \dfrac{Occ(w; s)}{Occ(w)} & \text{otherwise} \end{cases}$$

where $Occ(w)$ is the total number of occurrences of the word and $Occ(w; s)$ is the number of occurrences of the word in utterances expressing speech act s.

The probability of a speech act given an utterance $u = w_1, ..., w_n$ can then be defined as:

$$(20) \qquad P_w(s|u) \overset{def}{=} \begin{cases} P(s) & \text{if } k(u) = 0 \\ \sum\limits_{i=1,...,n} \dfrac{P(s|w_i)}{k(u)} & \text{otherwise} \end{cases}$$

$$(21) \qquad k(u) \overset{def}{=} |\{w_i : 1 \leq i \leq n \wedge occ(w_i) \neq 0\}|$$

Note that the function k, which returns the number of "known" words in the utterance, is used to cancel the effect of words not occurring in the training data (they assign probability 0 to all speech acts). This ensures that the sum of the probabilities of all speech acts is 1. Note further that for utterances which contain only words not occurring in the training data, each speech act gets assigned its overall probability.

The recognition method then assign each utterance $u = w_1, ..., w_n$ the speech act s_j for which $P_w(s_j|u)$ is maximal. The evaluation results for this simple word-based recognition method are as follows:[4]

N-Train/N-Eval	10/10	20/10	30/10	40/10	50/10	60/10	70/10
word_1	68.8	71.0	69.8	71.0	71.3	71.8	70.8

Though the simple word-based method yields an accuracy of over 70% and is thus considerably better than 32% it is still rather weak. We will now extend this method in three directions:

1. Instead of treating all words alike, we will "weigh" each word, so that its impact on determining the speech act can become more or less important.

2. We will take into account the speaker of an utterance, i.e. we will compute separate probabilities for clerk's and customer's utterances.

3. In addition to isolated words we will also consider sequences of two and three words (so-called bigrams and trigrams).

[4] In Appendix A we describe the details of our evaluation.

Exceptionality

The idea behind weighing the words is the following: whereas some words maybe really specific for certain speech acts (e.g. 'thank' or 'when'), others do not contribute much to the speech act but rather appear "randomly". Without weighing, these random words can cancel the effect of the speech-act specific words. Suppose, for example, a sequence of 7 words, two of which strongly indicate a specific speech act, whereas the other 5 words are not associated with any specific speech act. We would thus expect that they indicate the speech act 'inform' with probability 0.32 thereby pushing the score for 'inform' to 1.55. This will be very hard to beat by the other two words.

An easy and straightforward way to overcome this problem is to compute the *exceptionality* of each word and to give more exceptional words more impact than normal words. Exceptionality, in turn, can be determined by computing the distance between a word's speech-act frequencies and the overall speech-act frequencies. Let us make this more formal (assuming that $s_1, ..., s_m$ is our set of speech acts):

(22)
$$ex(w) \overset{def}{=} (1 - \frac{1}{Occ(w)+1})^{\omega} \sum_{j=1,..,m} |P(s_j) - P(s_j|w)|^{\delta}$$

A few remarks seem in order. First, note that we use the exponent δ to strengthen the impact of differences between the overall probability of a speech act and its probability relative to the word. Second we use the factor before the sum to take into account the number of overall occurrences of the word, i.e. for words occurring more often, the factor will be higher then for words occurring less often. The exponent ω is used to control how fast the importance rises wrt occurrences.

Knowing the exceptinalities of all words, we can compute the weights for the words occurring in an utterance. We sum up the exceptionalities of all the words occurring in an utterance and the weight of each word then is its own exceptionality divided by the total exceptionality of the utterance. Thus if we have an utterance comprising three words w_1, w_2, w_3, and $ex(w_1) = 140$, $ex(w_2) = 40$ and $ex(w_3) = 20$, then the weight for w_1 would be 0.7, the weight for w_2 0.2, and the weight for w_3 0.1. Note that this guarantees that the probabilities for the speech acts "remain" probabilities, i.e. summing up the probabilities assigned to the individual speech acts for an utterance one obtains 1.

Here is the formula for computing the probability of a speech act with the exceptionality-weighted word-based method:

(23)
$$P_x(s|u) \overset{def}{=} \begin{cases} P(s) & \text{if} \sum_{j=1}^{n} ex(w_j) = 0 \\ \sum_{i=1}^{n}(P(s|w_i)\frac{ex(w_i)}{\sum_{j=1}^{n} ex(w_j)}) & \text{otherwise} \end{cases}$$

Note that in this definition, words not occurring in the training data are taken care of by the ex function—they have exceptionality 0.

The accuracy of this method, which we will refer to in the following as *word_1E*, is substantially better than for the simple word-based recognition method:

N-Train/N-Eval	10/10	20/10	30/10	40/10	50/10	60/10	70/10
word_1	68.8	71.0	69.8	71.0	71.3	71.8	70.8
word_1E	75.4	78.4	79.8	81.5	82.1	83.5	83.0

Speaker Dependency

A second way to improve the recognition method is to take into account the speaker of an utterance. This could be advantageous since dialogues in the hotel-reservation domain are not symmetric, i.e. customer and clerk perform rather different roles in the dialogue (as is reflected by the differences in frequencies for some speech acts shown in Figure 1). Instead of computing probabilities based on all utterances, a speaker-dependent recognition method computes separate probabilities for utterances spoken by the clerk and those spoken by the customer.

As shown below this extension (called word_1SE) yields slightly better results than word_1E:

N-Train/N-Eval	10/10	20/10	30/10	40/10	50/10	60/10	70/10
word_1	68.8	71.0	69.8	71.0	71.3	71.8	70.8
word_1E	75.4	78.4	79.8	81.5	82.1	83.5	83.0
word_1SE	74.3	78.3	79.9	82.2	82.3	83.1	83.6

N-Grams

We will now consider a third possibility to improve the word-based recognition method. Instead of considering only single words occurring in an utterance, we take into account so-called *bigrams* and *trigrams*, i.e. sequences of two and three words. Thus we compute the frequency with which sequences as 'May I' or 'I would like' occur in each act. In general, we would expect that bigrams give better results than words, that trigrams give better results than bigrams, and so on. There is, however, the *sparseness* problem, i.e. the amount of data available for training is usually not sufficient to use NGrams with a high N. One way of coping with this problem is to use a technique called *smoothing*.

Our trigram method thus does not rely solely on probabilities induced by the trigrams occurring in an utterance, but also takes into account probabilities induced by the single words and by bigrams. The respective factors are 0.6 for trigrams, 0.3 for bigrams and 0.1 for individual words. Note that these factors have to add up to 1, to guarantee that one keeps probability values. That is, summing up the probabilities of all the speech acts $s_1, ..., s_m$ one would obtain 1.

To make this formal, we first define the probabilities for bigrams and trigrams (in the following we use the empty word ε for w_0 and w_{n+1}):

$$
\begin{aligned}
P_2(s|u) &\overset{def}{=} \sum_{i=0}^{n} \frac{P(s|w_i \circ w_{i+1})}{k_2(u)} \\
k_2(u) &\overset{def}{=} |\{w_i \circ w_{i+1} : 0 \leq i \leq n \wedge occ(w_i \circ w_{i+1}) \neq 0\}| \\
P_3(s|u) &\overset{def}{=} \sum_{i=1}^{n} \frac{P(s|w_{i-1} \circ w_i \circ w_{i+1})}{k_3(u)} \\
k_3(u) &\overset{def}{=} |\{w_{i-1} \circ w_i \circ w_{i+1} : 1 \leq i \leq n \wedge occ(w_{i-1} \circ w_i \circ w_{i+1}) \neq 0\}|
\end{aligned}
$$

Given these definitions, we can now define the smoothed trigram method. Again, we have to take into account the possibility that none of the words, bigrams, or trigrams occurring in the utterance have been previously encountered in the training dialogues:

$$
P_n(s|u) \overset{def}{=} \begin{cases} P_w(s|u) & \text{if } k_2(u) = k_3(u) = 0 \\ \lambda_1^2 P_w(s|u) + \lambda_2^2 P_2(s|u) & \text{if } k_2(u) \neq k_3(u) = 0 \\ \lambda_1^3 P_w(s|u) + \lambda_2^3 P_2(s|u) + \lambda_3^3 P_3(s|u) & \text{otherwise} \end{cases}
$$

Below we show the accuracy results (with and without exceptionality/speaker dependency) for word bigrams (word_2), trigrams (word_3), and for a smoothed trigram method (word).

N-Train/N-Eval	10/10	20/10	30/10	40/10	50/10	60/10	70/10
word_1	68.8	71.0	69.8	71.0	71.3	71.8	70.8
word_1E	75.4	78.4	79.8	81.5	82.1	83.5	83.0
word_1SE	74.3	78.3	79.9	82.2	82.3	83.1	83.6
word_2	80.7	82.0	82.8	84.0	84.7	85.4	84.8
word_2E	78.6	81.3	82.6	84.2	84.9	86.5	87.1
word_2SE	76.3	81.0	81.8	84.2	84.9	86.1	86.9
word_3	81.1	83.8	84.5	86.0	86.6	86.7	87.5
word_3E	78.9	81.8	83.0	84.7	85.4	86.1	87.2
word_3SE	77.6	82.8	82.7	85.2	86.2	86.8	88.0
word	81.6	83.6	84.4	85.8	86.6	86.6	87.0
wordE	80.2	82.9	84.2	85.4	86.6	87.3	88.3
wordSE	78.3	83.3	83.2	86.0	86.8	87.5	88.8

These results show that bigrams and trigrams clearly yield better results than individual words, and that a smoothed trigram method furter enhances the recognition accuracy. They also show that the gain of exceptionality is not as impressive for these methods as it is for the individual-words method.

We think that the accuracy of 88.8% obtained for smoothed trigram with exceptionality and speaker dependency is a very good result. In the remainder of this section we will take into account additional information for speech-act recognition, namely tagging information and contextual, i.e. macro-structural information. It can be shown that integrating these information sources further increases the recognition rate (though only slightly).

4.2 Using Tagging Information

The recognition methods presented so far are all based on a single information source, namely the wordforms occurring in an utterance. We will now briefly discuss the use of information produced by a tagger for our speech-act recognition. Such a tagger assigns each word occurring in an utterance a tag which can contain syntactic and/or semantic information.

Note that using a tagger is different from using a syntactic or semantic representation for two important reasons:

1. the output from a tagger is a simple list of tags and not a complex recursive term (though some taggers can also produce such terms);

2. taggers are usually quite robust, i.e. they produce output even if they have problems in parsing the input.

We used the output of a tagger developed at ATR by Ezra Black and Stephen Eubank, which in the version we used (March'97) produced nearly 3000 different tags containing syntactic and semantic information. In addition to using these tags we also used smaller sets of part-of-speech tags and semantic tags, in which the original tags can be mapped.

Below we show the results for recognition methods based on tags (tag_N), part-of-speech tags (pos_N) and semantic tags (sem_N) all obtained with exceptionality and speaker dependency. The definitions of the respective formulae are analogous to the ones used for word bigrams and trigrams and their smoothed combination.

N-Train/N-Eval	10/10	20/10	30/10	40/10	50/10	60/10	70/10
tag_1SE	63.2	65.6	66.5	67.4	68.5	68.4	68.3
tag_2SE	74.8	77.2	78.2	79.9	81.0	81.7	82.4
tag_3SE	75.3	78.2	79.8	81.4	82.9	83.4	84.4
tagSE	75.8	78.6	80.1	81.7	82.8	83.3	84.1
pos_1SE	38.8	39.6	39.7	39.7	40.2	39.7	39.9
pos_2SE	61.9	63.6	65.0	66.0	66.9	66.8	67.6
pos_3SE	71.7	74.6	74.6	76.4	76.5	76.8	77.7
posSE	71.8	73.9	74.3	76.1	75.9	76.0	76.5
sem_1SE	48.3	50.1	49.6	50.2	51.0	50.5	50.9
sem_2SE	58.0	63.2	63.1	64.9	65.6	66.8	67.7
sem_3SE	63.2	67.6	68.3	70.2	72.2	72.2	72.7
semSE	64.1	68.4	68.7	70.4	72.2	72.7	73.1

The results show that recognition methods based on tags do not perform as good as those based on words. However, combining heterogenous information sources seems to slightly increase the recognition rate. The following table shows the results for a method using wordforms and tags (word_tag_SE) compared to the method using only wordforms (smoothed trigrams):

N-Train/N-Eval	10/10	20/10	30/10	40/10	50/10	60/10	70/10
wordSE	78.3	83.3	83.2	86.0	86.8	87.5	88.8
word_tagSE	80.2	84.0	84.7	87.0	87.5	88.3	89.4

4.3 Using the Speech-Act Context

Recognition methods based on the macro-structure use the overall dialogue context and not the current utterance as information source. Their primary aim is usually to *predict* the next speech act on the basis of this information, and it is perhaps a bit misleading to treat them as recognition methods. Our aim here is mainly to show that they perform considerably poorer than recognition methods based on micro-structure, but that they are useful when used in combination with these methods.

The easiest way to build a recognition method based on macro-structure is to compute probabilities of NGrams of speech acts. Given a sequence of speech-acts we can then predict, which speech act is the most probable to follow this sequence.

The accuracy results for this method are as follows for different values of N (act_NASE). Again we used the smoothing technique to combine different NGrams of speech acts (act_ASE). All results were obtained with exceptionality and speaker dependency.

N-Train/N-Eval	10/10	20/10	30/10	40/10	50/10	60/10	70/10
act_1ASE	40.7	41.5	41.6	41.3	42.1	42.6	42.3
act_2ASE	42.3	46.1	46.4	47.5	47.4	47.4	47.9
act_3ASE	39.4	42.5	43.7	45.4	45.4	45.9	46.7
act_4ASE	36.6	41.3	40.9	42.7	43.0	44.4	44.4
act_5ASE	34.2	36.9	36.8	39.0	39.8	39.8	40.9
actASE	41.9	44.7	45.2	47.1	47.2	47.6	48.2

Compared to the micro-structural recognition methods these results are pretty bad, but this should not be too surprising. In general it is not possible to precisely predict the speech act of the next utterance. However, it is often possible to predict a list of 2 or 3 speech acts which are likely to follow (see [Alexandersson et al. 95] which also discusses plan-based approaches to speech-act prediction).

Though the accuracy of the macro-structural method is already pretty bad, it gets worse, if we make the scenario more realistic. The above evaluation has been more in the spirit of evaluating the accuracy of prediction than of recognition: we evaluated the accuracy of predicting the next speech act given the *correct context* of speech-acts. Thus the NGram information we used as a basis were the previous speech acts as they were *annotated in the evaluation dialogues*. However, to really evaluate the accuracy of the macro-structural method from a recognition point of view, we have to take the *recognized speech acts* as a basis for computing the NGrams.

The following table shows the accuracy of a smoothed recognition method based on speech-act NGrams which uses the recognized speech acts (actR).

N-Train/N-Eval	10/10	20/10	30/10	40/10	50/10	60/10	70/10
actRSE	32.2	33.3	33.7	35.5	37.2	38.5	31.8

As one would expect this has considerable impact on the recognition rate: if the recognition rate is around 50% given the correct context it means that there will hardly ever be the correct context to predict the next speech act. The accuracy is thus hardly above the 32% and though there seems to be some improvement when increasing the number of training dialogues, there is a sharp recline for the 70/10 combination.

However, combining macro-structural information with the information used so far yields a further increase in the recognition rate (though a rather small one):

N-Train/N-Eval	10/10	20/10	30/10	40/10	50/10	60/10	70/10
wordSE	78.3	83.3	83.2	86.0	86.8	87.5	88.8
word_actRSE	78.1	83.8	84.7	86.3	87.3	88.0	88.7
word_tagSE	80.2	84.0	84.7	87.0	87.5	88.3	89.4
word_tag_actRSE	80.5	84.2	84.9	86.9	87.6	88.4	89.6

Act	Labelled	Assigned	Correct	Recall	Precision
inform	127	123	117	92.1	95.1
acknowledge	67	75	63	94.0	84.0
thank	33	34	33	100.0	97.0
temporizer	27	27	26	96.3	96.3
desire	21	30	21	100.0	70.0
yes	21	16	11	52.4	68.8
wh-question	20	21	18	90.0	85.7
yn-question	15	14	8	53.3	57.1
information-request	13	12	8	61.5	66.7
confirmation-question	11	12	7	63.6	58.3
greet	11	9	9	81.8	100.0
action-request	10	9	7	70.0	77.7
farewell	10	8	8	100.0	97.1
offer	6	6	6	100.0	100.0
believe	6	6	5	83.3	83.3
accept	4	1	1	25.0	100.0
apology	4	5	4	100.0	80.0
promise	3	2	2	66.7	100.0
permission-request	3	2	2	66.7	100.0
no	2	2	2	100.0	100.0
offer-follow-up	1	1	1	100.0	100.0

Figure 2: Precision and Recall for wordSE (70/10 1 Run 86.5% accuracy)

5 Discussion and Future Work

In this section we briefly discuss the results obtained so far and sketch possible directions of future work.

Let us begin our analysis by taking a closer look at how good the recognition method does in recognizing the individual speech acts. That is, for each speach act, we count how often it is labelled, how often it is assigned, and how often it is assigned correctly. Given these numbers, we can compute the so-called *Recall* and *Precision*:

$$\text{Recall} \overset{def}{=} \begin{cases} 0 & \text{if Labelled } = 0 \\ \frac{\text{Correct}}{\text{Labelled}} & \text{otherwise} \end{cases}$$

$$\text{Precision} \overset{def}{=} \begin{cases} 0 & \text{if Assigned } = 0 \\ \frac{\text{Correct}}{\text{Assigned}} & \text{otherwise} \end{cases}$$

Figure 2 shows recall and precision for a single 70/10 evaluation of the smoothed trigram method (word_SE) whose overall accuracy was 85.5%. As can be seen, for some speech acts recall and precision is close or equal to 100% whereas for others it is below 70%.

One might have hoped that including macro-structural information would yield better results wrt ambiguities arising between 'yes' and 'acknowledge', or between 'yes' and 'accept' (since such ambiguities can usually be resolved only on the basis of the dialogue context). However, this was not the case, the increase in accuracy seems to be due basically to more correct recognition of the speech act 'inform'.

The following table lists the most common recognition errors for two runs of a 70/10 cycle indicating the percentage of misrecognitions of the respective type:

Recognized	*Annotated*	
acknowledge	yes	15.3%
inform	promise	7.6%
inform	confirmation-question	6.8%
inform	yn-question	6.8%
inform	desire	5.1%
inform	suggest	5.1%

A more detailed analysis of precision and recall might be useful in order to identify those areas of the recognition method which still have a high error rate. Once these areas are identified one can then devise special strategies to cope with them.

With respect to the error rate, another issue is of interest. We have defined our recognition methods in a way that they compute a probability distribution of speech acts for each utterance. This allows us not only to pick the speech act with the highest probability, but by looking at the probability itself we might learn how *confident* we can be that we have made the right choice.

In order to test this hypothesis, we computed the average probability of the assigned speech act for cases in which the assignment was correct and for cases in which the assignment was wrong. The results for the test run used to compute recall and precision were that the average probability assigned to the best speech act for correct recognitions was 0.85, whereas for wrong recognitions it was only 0.68. Thus the average probability of the best speech act seems to be considerably higher if it is the right one, i.e. the probability assigned to the best speech act might be used as a confidence measure for the correctness of the recognition itself.

Finally, a word should be said about the weighing of the individual factors in recognition methods using several heterogenous information sources. We chose the weights by trying out different possibilities (largely on the basis of the accuracy obtained for the single-feature methods) and took the ones which performed best for some sample runs. This is clearly far from satisfactory. We also experimented with *genetic algorithms* to determine the optimal weights. These experiments basically yielded two results:

1. there are in general many different assignements of weights yielding the same (optimal) accuracy rate;
2. the optimal weights found by the genetic algorithm differed from the weights chosen by hand but did not produce substantially better accuracy rates.

Nevertheless, when dealing with a large number of features it would be rather useful to have a machine-learning tool integrated for determining the best distribution of weights.

6 Conclusion

In this paper we have presented a data-based method for speech-act recognition. We have used a corpus of 80 annotated dialogues taken from the ATR hotel-reservation domain for training and evaluation of several methods based on (sequences of) wordforms, tags, and speech-act context and obtained recognition rates between 85 and 90%.

Though these results still have to be validated by a more thorough evaluation based on a larger and more heterogeneous corpus, they indicate that a rather simple statistical approach to speech-act recognition will yield recognition rates above 85%.

There is, however, a very big caveat: the results were obtained on pre-segmented, transcripted data. Before fine-tuning the recognition methods it seems rather more important to us, to evaluate the methods on actual speech-recognizer output.

References

[Alexandersson et al. 95] J. Alexandersson, E. Maier, N. Reithinger, "A Robust and Efficient Three-Layered Dialogue Component for a Speech-to-Speech Translation System", *Proceedings of EACL-95*, Dublin, 188–193

[Austin 62] J.L. Austin, *How to do things with Words*, 1962

[Jekat et al. 95] S. Jekat, A. Klein, E. Maier, I. Maleck, M. Mast, J.J. Quantz, *Dialogue Acts in Verbmobil*, Verbmobil Report 65, 1995

[Levinson 83] S.C. Levinson, *Pragmatics*, Cambridge: Cambridge University Press, 1983

[Mast et al. 95] M. Mast, H. Niemann, E. Nöth, S.G. Schukat-Talamazzini, "Classification of Dialogue Acts with Semantic Classification Trees and Polygrams", *IJCAI-95 Workshop: Learning in NLP*

[Popovici, Baggia 96] C. Popovici, P. Baggia, "Specialized Language Models Using Dialogue Predictions", Computation and Language E-Print Archive, http://xxx.lanl.gov/cmp-lg/, Article 9612002

[Schmitz, Quantz 95] B. Schmitz, J.J. Quantz, "Dialogue Acts in Automatic Dialogue Interpreting", *Proceedings of Theoretical and Methodological Issues in Machine Translation, TMI-95*, 33–47

[Searle 69] J.R. Searle, *Speech Acts*, London, 1969

[Seligman et al. 94] M. Seligman, L. Fais, M. Tomokiyo, *A Bilingual Set of Communicative Act Labels for Spontaneous Dialogues*, ATR Technical Report TR-IT-0081

A Evaluation

This section describes in some detail how the evaluation figures presented in Section 4 were obtained.

First, it should be noted that we used the wordforms as they occurred in the transcriptions:

1. punctuation symbols were treated as words;
2. no spelling correction was performed;
3. no conversion from upper-case characters (at sentence beginnings) to lower-case characters was performed ('Okay' and 'okay' were treated as different words);
4. "melted" words like 'I'd' were treated as a single word;
5. expressions marked in the transcriptions, e.g. '[well]' were treated as a single word, which contains the brackets.

We did so in order to minimize the effort needed to use the available material. Though punctuation won't be available in the speech-recognizer output, we kept it, since a question mark carries information usually available in prosodically annotated speech-recognizer output (e.g. rise vs fall).

Let us now briefly describe our training and evaluation strategy which has been the same for all recognition methods described in this paper:

1. We randomly select n dialogues from the 80 labeled dialogues and use them as training material, i.e. we compute the probabilities used by the respective recognition method on the basis of these dialogues.

2. We randomly select 10 dialogues from the 80 labeled dialogues and use them for evaluation. This set of evaluation dialogues *always is disjunct* from the set of dialogues which has been used for training.

3. For each utterance in the evaluation dialogue we compute the speech act with the method to be evaluated. We then compare the computed speech act with the annotated speech act and if they are identical we increase the hit counter.

4. The accuracy of the method is computed by multiplying the hit counter with 100 and dividing it by the total number of utterances in the evaluation dialogues.

5. We evaluated each method for n = 10,20,...,60,70 and for each n we performed 10 training-evaluation cycles and then computed the average accuracy.

6. All methods were evaluated with the same (random) training and evaluation dialogues, i.e. we first picked 10 choices of n training dialogues and 10 evaluation dialogues and then used these choices for the training and evaluation of each method.