Conveying the user's intention by generating speech act conditionals as indirect answers

Maryam Mohammadi, Judith Fischer, Maurice Langner and Ralf Klabunde
Ruhr-Universität Bochum
Department of Linguistics
Germany
{maryam.mohammadi|judith.fischer|maurice.langner|ralf.klabunde}@rub.de

Abstract

We show how *speech act conditionals* (SACs) answer polar questions indirectly in a question answering system in a sales setting. Based on a classification of SACs we develop a probabilistic model that generates SACs as positive or negative answers. Empirical studies have been performed to establish the input probabilities for the system and to test the adequacy of the generated answers. The results suggest that SACs are appropriate answers if the user's requirements are not well established.

1 Introduction

A system that provides an information-seeking user with adequate information is confronted with two selection problems that reflect different, but intertwined, preferences. The first one concerns the content of the answer. Since a question signals an underlying decision problem of the inquirer the system is only able to infer indirectly, this assumed requirement holds with a certain plausibility only. The second preference problem concerns the linguistic form of the answer. A question can be answered in numerous ways, but not every form of the answer is appropriate in the given context.

For example, a polar question can be answered directly (*yes/no*), possibly with iterative information (see A1), or indirectly by explicitely stating alternatives and triggering the implicature that the answer is *no* (A2), or indirectly by triggering the same implicature and mentioning the assumed requirement that motivated the question (A3):

Q: *Is there a supermarket nearby the apartment?*

A1: *Yes/no. (There is a/no supermarket nearby.)*

A2: *There is an organic shop around the corner.*

A3: *There is an organic shop around the corner, if you are looking for shopping facilities close to your apartment.*

A3 is an example of the answer type we are dealing with in this paper. These so-called speech act conditionals (SACs) – often called "biscuit" conditionals in remembrance to Austin (1970)'s example *there are biscuits on the sideboard if you want some* – show the link between preferred content and preferences for a linguistic form quite well: The antecedent of the SAC expresses the presumed requirement for the information given in the consequent, but mentioning this requirement should be tied to the common ground of system and user. Only when the requirement is still under discussion it is useful to mention it as a possible reason for asserting the information in the consequent.

In other words, SACs are conditionals where the *if*-clause expresses a condition for uttering the main clause, namely the circumstances under which the consequent is discourse-relevant, and not a condition for the truth of the main clause. Hence, contrary to standard conditionals, speech act conditionals do not have a meaning related to material implication; we perceive both propositions expressed as semantically unrelated. Instead, what matters is the speech act level of interpretation and, therefore, the felicity conditions for successfully using an SAC.

SACs have received some attention in formal semantics and pragmatics (Franke, 2006; Fulda, 2009; Siegel, 2006), since they raise the question whether a unified theory of the interpretation of SACs and other types of conditionals can be developed, but these studies neither consider computational issues concerning their interpretation and generation, respectively, nor do they explicate the dialogical character of their use. Work on the generation component in (spoken) dialogue systems concerns various aspects of user-adaptive information preparation and its linguistic realization, de-

Proceedings of the 14th Conference on Natural Language Processing (KONVENS 2018)
Vienna, Austria – September 19-21, 2018

pending on the user's level of domain expertise (e.g., Janarthanam and Lemon (2014)). Although there is some work on the generation of indirect answers (Green and Carberry, 1999; de Marneffe et al., 2009), to our knowledge SACs have not been adressed so far.

In this paper, we focus on SACs as indirect answers in an real estate setting where the system provides information about apartments to a user/customer who is looking for an apartment to rent. We ignore the use of SACs as politeness hedges in sentences like *It still works if I may say so* or *If I am being frank, you don't look good today*, since these SACs are not related to preference-oriented requirements.

The user asks polar questions about whether the potential apartment holds some attributes. The system as realtor, that has no access to the user's requirements, should determine the best (alternative) answer regarding the user's decision problem.

For example, the question *Are there any restaurants near the apartment?* could be answered by three different types of SACs:

1. *If you enjoy eating out, there is an Italian restaurant nearby.* (Positive SAC)

2. *If you enjoy eating out, there is an Italian restaurant in the neighbored quarter.* (Negative SAC)

3. *If you enjoy eating out, there is an Italian restaurant and a food-court nearby.* (Alternative SAC)

SACs as indirect answers come with three different functions. Their uses have different consequences in Q/A systems, but should be modeled in a common way. For example, answer (1) expresses that the real estate agent assumes that the user is able to implicate that the Italian restaurant is the only restaurant nearby, and that the question was motivated by the user's general pleasure of eating out. In sum, this positive SAC (PSAC) conveys: the answer is *yes*, the user shall implicate that the only restaurant nearby has been mentioned, and the supposed motivation of the user for asking this question has been mentioned by the antecedent of the SAC.

Things are different with SACs that function as a negative answer (NSAC). The NSAC given above signals the following information: The answer is *no* and given the assumed requirement for the question as expressed by the antecedent of the NSAC, this requirement can be satisfied by the restaurant in the neighbored quarter.

The third type are alternative speech act conditionals (ASACs), as we name them. By means of ASACs as answers, the system answers the question positively, but it offers two alternatives for the presumed requirement of eating out that are at least equally probable.

We see that the antecedent of positive, negative, and alternative SACs expresses the presumed requirement underlying the question, but these three types of SACs have slightly different discourse functions. While PSACs answer the question by providing an asserted proposition and mentioning the supposed motivation for the question (and possibly triggering an implicature), NSACs provide an alternative solution to the assumed motivation underlying the question and, by that, triggers the implicature that the answer has been negated. Alternative SACs offer more than one attribute with similar probability for the presumed requirement.

Since SACs lay out the assumed requirement, they should rather occur at the beginning of a question/answer sequence, where the user's requirements are still under discussion, than at the end. Such a discourse-dependend generation of SACs as indirect answers must be taken into account as well.

A user's presumed preferences have also been investigated in recommender systems as online sales applications. Recommender systems can be divided into two groups: Collaborative recommendation is based on the users (explicit or implicit) collaboration with one another. Content-based recommendation, in contrast, deals with information overload and selects the most interesting items from a given set. Both types of systems discriminate relevant from irrelevant information (Jannach et al., 2011). Techniques for content-based recommendation are quite similar to our approach, but recommendation tasks are only partially identical with the task of finding the most probable alternative for a presumed requirement since the system has no access to reliable information about the user.

2 Probabilistic model and its empirical background

We propose a model rooted in probability theory that generates SACs by strategic reasoning about possible requirements of the user. The model orients at current probabilistic approaches that attribute communication to basic cognitive principles concerning various kinds of decision making based on the agent's common ground ((Frank and Goodman, 2012), (Franke and Jäger, 2016), (Potts et al., 2016), (Stevens et al., 2016), (Zeevat and Schmitz, 2015)). We presume that each question is motivated by an underlying requirement of the user and the system elicits this requirement.

The represented partial information of the sales agent contains information on the attributes of the object under discussion, but lacks certainty about the underlying decision problems the user has. The user lacks knowledge on the configuration of the object under discussion, while he has full awareness of his requirements. The generation of answers therefore serves the function of enriching the common ground with the user's requirements such that the sales agent may react to decision problems while the user evaluates in which kind and degree the object under discussion satisfies his needs.

The basic objects in the database are the available flats with one being the current object under discussion, requirements r and attributes a. The user's question Q is about some attribute q of the object under discussion. Requirement r constitutes the underlying decision problem motivating q, on the base of which a may be offered as an equal or better substitute for satisfying r.

User responses may be accept the object, reject the object, or pose a follow-up question. The system's goal is helping the user to make the optimal decision efficiently by anticipating the requirements r that are relevant to the user.

Input to the model are the prior probabilities of requirement r, a set R^q of possible requirements true of q and attributes q and a, respectively. The conditional probability $P(r|q)$ will be determined by Bayes' rule, which allows us to trace back the probability $P(r|q)$ that a user posing question q is motivated by requirement r to the task of finding relevant questions for expressing a requirement:

$$P(r|q) = \frac{P(q|r) \times P(r)}{\sum_{r' \in R^q} P(q|r') \times P(r')} \tag{1}$$

Depending on whether or not the object under discussion has attribute q, the system chooses between a positive or negative answer. In case the model leads to generating a speech act conditional, it chooses between a PSAC, an NSAC, or an ASAC. For example, for a certain apartment as the object under discussion, assumed requirement r = gardening, q = garden (*Does the apartment have a garden?*) and a = balcony, the SACs are generated as follows:

	$r = [\![$ If you want to do some gardening $]\!]$
NSAC:	... the appartment has a balcony.
PSAC:	... the appartment has a garden.
ASAC:	... the appartment has a balcony and a garden.

2.1 Empirical background

We performed three studies to support the assumptions made in this model. Each study was designed using Testable.org and carried out via Amazon Mechanical Turk. Participants received a small compensation for their work.

For determining the requirements, we determined ten common questions that people mostly ask before renting an apartment. We run our first experiment to determine the prior probabilities for our model. With two different questionnaires, 120 subjects (7 of them failed to pass the experiment) were presented a set of requirements or attributes randomly, and they were asked to rate for each item how relevant they may be in an apartment sales context. In order to receive the probabilities of both interlocutors in the dialogue, we divided the participants into two groups taking either the role of the user or the real estate agent. The results define the prior probability distribution of the attributes and requirements.

A further set of studies concerns the acceptability of the different types of SACs (PSACs, ASACs, NSACs) as indirect answers. Again we divided the participants into the group of users and real estate agents and compared direct *yes/no*-answers with SACs. 5 questionnaires with the same 30 questions offered different types of answers randomly. Participants were asked to read one question and answer per page and rate the acceptability level on a scale from 0 to 100. 241 out of 250 subjects (119 as customers and 122 as realtors) successfully participated in the experiment (Figure 1).

The one-way ANOVA test yields $F(4,2405)$: $217.3, P < 0.001$ and illustrates a significant difference between answers. There were no significant differences between the user and the sales agent group; both of them show almost the same level of acceptability in all different types of answers. Post hoc comparisons using the Tukey HSD test indicated that the mean scores for the direct answers

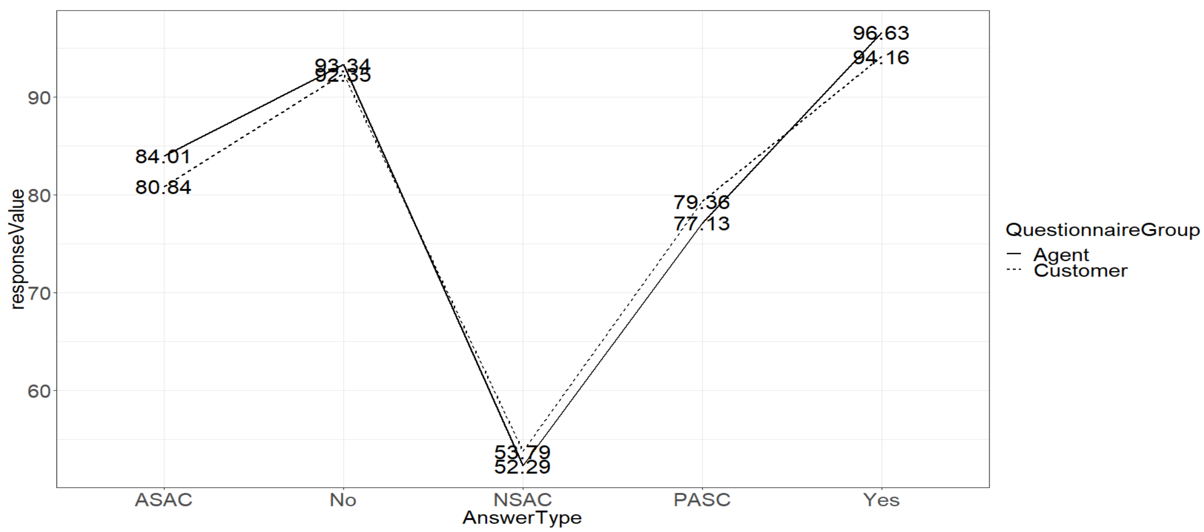

Figure 1: *Acceptability of all types of answers from the customer's and sales agent's perspective.*

were not significantly different than the PSAC and ASAC answers. However, the NSAC significantly differ from the rest.

Contrary to our expectations on indirect answers, the mean value of NSACs gained the lowest rating, although a detailed study reveals interesting results. The data analysis of NSAC answers (Figure 2, left) shows that there were 5 questions with high mean values ($65 < \mu < 89$) and 5 questions with low mean values ($21 < \mu < 37$). This crucial difference motivated us to reconsider the experiment and focus on NSACs to figure out the reason behind the disagreement.

A closer look at the relevance concerning question q, requirement r and alternative a, that is, $P(q|r)$ and $P(a|r)$, shed light on the problem. The *epistemic relevance* between two propositions deals with the speaker's evaluation of the degree of confidence upon which a and q are related to r and results in the degree under which the requirement could be satisfied. This can be weighted by the Δ rule (Oberauer et al., 2007): $\Delta p = P(a|r) - P(q|r)$.

The new questionnaires were designed under the same framework and format of the prior experiment, but this time the answers were provided in three groups based on Δp. When, for instance, $\Delta p > 0$, attribute a in an NSAC for the assumed requirement r seems to be more relevant than the question attribute q.

The results of the study meet the expectations. The ANOVA test (with 49 participants) indicates a significant difference when $\Delta p < 0$ rather than $\Delta p >= 0$. Figure 2, right side, shows that when the probable epistemic relevance between the alternative answer and the requirement is less than the relevance between the question and the requirement ($\Delta p < 0$), participants gave a lower acceptability score ($\mu = 47.28$). On the other hand, when the relevance is quite equal or more, the acceptability is higher (μ=63.20 and μ=62.38 respectively). Therefore, NSACs meet the acceptable criteria as indirect answers if the alternative attribute is likely for r.

The final empirical study concerns the question where to generate an SAC in a Q/A sequence. For this, we confined our study to questions about activity and interior attributes. The results should be generalizable to each category of requirements, however.

We ran an experimental study to figure out the best place for mentioning requirements in a Q/A sequence. Our hypothesis was that at the beginning of a sequence, uttering the assumed requirements in an SAC is more acceptable than towards the end.

We designed four long Q/A sequences (18 questions and answers) between a realtor and a user on the aforementioned topics.

The user explicitly states that she is going to ask questions on a certain topic. When a new topic in the conversation starts, SACs were added at the beginning, in the middle or at the end of the Q/A sequence. The participants were asked to rate the agent's attitudes toward sharing information, the distribution of new information, and whether the user's probable needs have been fulfilled. We also asked participants to share their

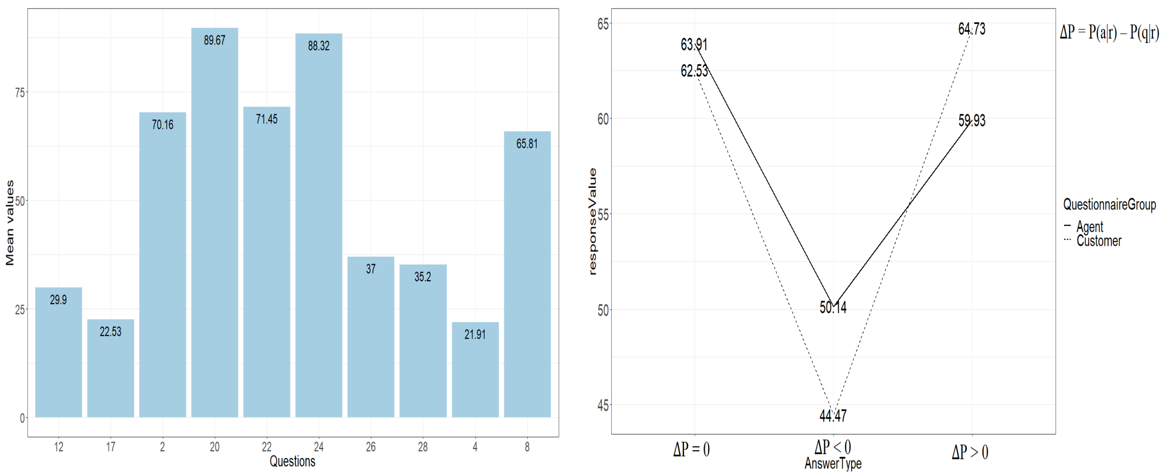

Figure 2: *Left: NSACs answers per question. Right: Different* Δp.

idea and intuition about each conversation. The ANOVA test shows a difference between the beginning and the middle group, and the end position; $F(2,687) : 2.89, P < 0.1$. The result shows the participant's dissatisfaction in mentioning the requirements at the end of the sequence.

We perfomed the same study without mentioning the topic, but with a short sequence of 7 Q/A-pairs. In this study, no significant difference between the three groups of sequences exists, but when the Q/A sequence gets longer, the acceptability of a speech act conditional at the beginning of the sequence stays almost the same, while the acceptability rate in the middle position decreases. Hence, the study supports our assumption that SACs are better suited in a Q/A setting where the user's requirements are not well-established.

2.2 The probabilistic model

The system has to anticipate the underlying decision problem that induces the user to ask for question attribute q. For this, we define a benefit that depends on whether the chosen requirement r is suitable for q or not. The benefit of looking up requirement r for attribute q is defined as:

$$B(r|q) = 1, \text{ if } r \in R^q; \text{ else } 0$$

The benefit B may be diminished by a dialogue-sensitive cost κ for realizing the requirement. This cost encodes the burden from choosing a more complex answer containing r in comparison to a straightforward *yes/no* as answer. The cost κ_c is a dynamically calculated value that depends on the recent dialogue history and the category of requirements c. In our domain, we distinguish between five categories c: interiors, transportation, food, shopping and activities.

The parameter κ_c represents for a category of requirements c the degree of common ground established between user and sales agent. Since SACs express the assumed requirement of the client in its antecedent, they rather occur at the beginning of a Q/A sequence than at the end, as shown by the study described above. Hence, the sales agent's assumptions evolute during the ongoing discourse.

The category-dependent assessment of the state of the common ground is what κ_c expresses: for each category c the value of κ signals the amount of shared information concerning the requirements of the user:

$$\kappa_c = \sum_{i=1}^{n} P(r_i^c | a_i)$$

with n the number of Q/A sequences realized so far. We add the conditional probabilities for each requirement r_i in category c, given the attributes a_i that have been asked for so far. With a high value of κ_c the expected utility decreases. Thus, the value of κ_c influences the generation of SACs.

For example, when the user asks several times about attributes concerning transportation issues, after some time the system does not generate an SAC since $\kappa_{transportation}$ receives a value that blocks the generation of an SAC.

Since the requirement of the user is not known to the sales agent, his strategy is to maximize the utility of a chosen requirement. This is handled by the expected benefit EB for a requirement, given the attribute a_c of category c and the set of all

Proceedings of the 14th Conference on Natural Language Processing (KONVENS 2018)
Vienna, Austria – September 19-21, 2018

possible requirements R^q of the current question q_c:

$$EB(r^a|a_c,R^q) = \sum_{a_c \in R^q} P(r^a|a_c) \times B(r^a|a_c)$$

Attribute a_c can be the attribute the user is asking for ($a_c = q_c$). In this case the benefit B results invariably in 1 and the conditional probabilities will just be added. But if we compare an alternative attribute a of category c with question attribute q_c, and q_c is not true of the apartment, we consider only the requirements for the original question q_c. This guarantees that the model does not generate a SAC using a requirement that is irrelevant for the attribute that the user asked for:

Q: *Does the apartment have a big kitchen?*
A: **If you want natural light, it has a balcony.*

This preselection of requirements has a positive side effect: it guarantees that the answer is relevant in a Gricean sense. In other words, we determine the epistemic similarity of the question attribute and the alternative attribute in the given dialogue context by comparing their relevance for possible decision problems that may have motivated the user to utter the question. Accordingly, the expected utility of r and q can be determined by:

$$EU(r|q_c) = EB(r|q_c,R^q) - \kappa_c$$

A speech act conditional is generated if the expected utility is larger than 0, because in this case it is more advantageous to linguistically realize the requirement than to not mention it. If more than one r causes $EU(r|q) > 0$ to be true, than the maximal value is chosen in order to generate the most relevant speech act conditional (see the pseudocode of the decision tree for SACs in Table 1.)

If attribute q_c is true of a flat f, the decision tree determines whether there is some requirement r^q in the set of possible requirements R^q, which trigger the expected utility of r and q_c to be positive (> 0). If this is not the case, none of the requirements is relevant enough to outweigh the cost of generating a more complex answer. If more than one r satisfying the condition is found, the model chooses the most probable one. Following this decision, the model checks whether there is some alternative attribute a_c that is true of f, whose expected utility $EU(r|a_c)$ is larger or equal to $EU(r|q_c)$. If such an attribute is found, the model generates an ASAC naming both attributes, q_c and a_c. Else, the model generates a PSAC.

On the other hand, if attribute q_c is false of flat f, the model checks whether there is some alternative attribute a_c satisfying requirements p^q such that the expected utility $EU(r|a_c)$ is positive. If $EU(r|a_c)$ is negative, the decision tree terminates, generating a direct negative answer. If some a_c is found, the model checks whether the probability $P(r^q|a_c)$ is larger than the threshold τ:

$$\tau = \frac{\sum_i P(r_i|a)}{|(r,a)|}$$

$|(r,a)|$ is the number of all requirement-answer combinations. This threshold determines whether a requirement is probable enough to be worth the effort made to be uttered. In other words, if the probability is higher than τ, the underlying decision problem is obvious enough to be uttered. In this case, the system generates an NSAC. If the requirement is not that obvious, the system generates an indirect answer.

3 Overall evaluation of the system

The Q/A system described in this paper and used for the experimental studies is available via the anonymized URL `realtorservice.duckdns.org`.

We compared the Q/A system that is able to generate SACs dynamically with a baseline system that generates direct answers only. This baseline system has been provided by setting high κ values so that no SACs will be generated. Let us call the system that is able to generate SACs as answers the dynamic system and the other one the static system.

In using each system, participants were asked to play the role of a person who is asking questions about a flat for her/his friend. The participants were informed about requirements for their friend and they have been prompted to ask questions in order to find out whether the flat is appropriate or not. We mentioned that they are interacting with a Q/A system and that our goal is to evaluate the quality of the generated answers.

13 out of 50 participants failed the experiment with the dynamic system since they have asked less than 4 question, which is not enough to determine language efficiency. The questions were answered with SACs and direct yes/no answers. At the end of the experiment participants answered 10 questions

Proceedings of the 14th Conference on Natural Language Processing (KONVENS 2018)
Vienna, Austria – September 19-21, 2018

Algorithm 1 An algorithm to determine the content for speech act conditionals

Input: A database with category-related attributes A_c and requirements R, an object under discussion f with attributes from A_c, a probability distribution $P(a|r)$ of attributes satisfying the requirement, a user question about the attribute q, threshold τ

Initialize: $\forall c : \kappa_c = 0$, τ

while user response $\neq$ accept(f) or reject(f) **do**:
  if $f(q_c)$ == true:
    if $argmax(EU(r^q|q_c)) > 0$:
      if $argmax(EU(r^q|a_c)) \geq argmax(EU(r^q|q_c))$:
        generate ASAC(a_c, q_c, r)
      else
        generate PSAC(q_c, r)
    else
      generate direct positive answer
  if $f(q_c)$ == false:
    if $argmax(EU(r^q|a_c)) > 0$:
      if $P(r^q|a_c) \geq \tau$:
        generate NSAC(a, r)
      else
        generate indirect answer
    else
      generate direct negative answer
  $\kappa_c := \kappa_c + \sum_{i=1}^{n} P(r_i^c|a_i)$

Table 1: The pseudocode for content determination for SACs

on the quality of the answers on a feedback page for the final evaluation.

We performed the same study with the static system. The answers were direct yes/no answers or, by random, simple alternative answers. 11 out of 50 participants failed this test.

Figure 3 shows that in interacting with the static system 9 particiants asked more than 10 questions to make a decision about the apartment, while only 1 participants raised more than 10 questions with the dynamic system. There is also a tendency to rather accept the apartment than rejecting it when SACs have been used. SACs are obviously more informative, and their use seems to cast a positive light on the apartment.

Although the comments show more satisfaction when using the dynamic system, the analysis of the participants rating did not show a significant difference between both systems (see Figure 4). However, for the questions on the feedback page *How probable is that a human agent generates the same answers?* and *How probable is that you found out the answers were generated by a machine if we hadn't mentioned?* we received significant differences. The dynamic system has been evaluated better for generating humanlike answers.

In sum, the generation of speech act conditionals has a positive effect on the efficiency of the dialogue sequence, and they have been rated as quite natural.

A final note on modeling the user might be in order. Currently, adapting the SACs to the assumed user's requirements is primarily managed by the κ parameter that is essentially working as a counter. In order to make user adaptation more dynamic, we are going to replace κ by user types learned from dialogue sequences. For example, if the dialogue history suggests a strong interest in family-related attributes, requirements concerning the way to school etc. should receive a lower probability for being expressed than other requirements.

Acknowledgments

This work has been supported by the Deutsche Forschungsgemeinschaft (DFG) (Grant nr. KL 1109/6-2 and BE 4348/3-2, project 'Bayesian approaches to preference-based answer generation in dialogue'.

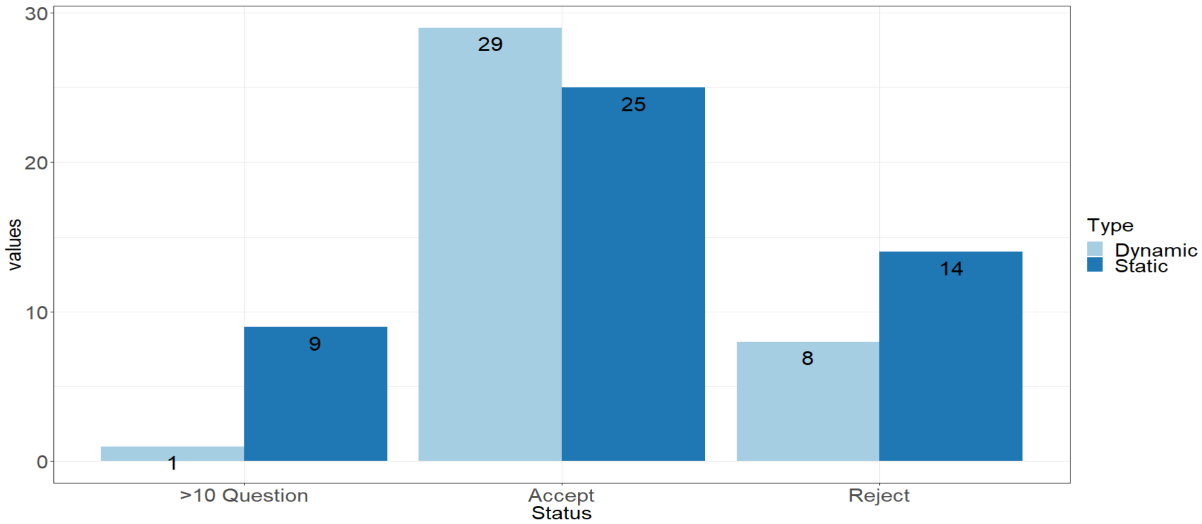

Figure 3: *Comparison of dynamic and static system.*

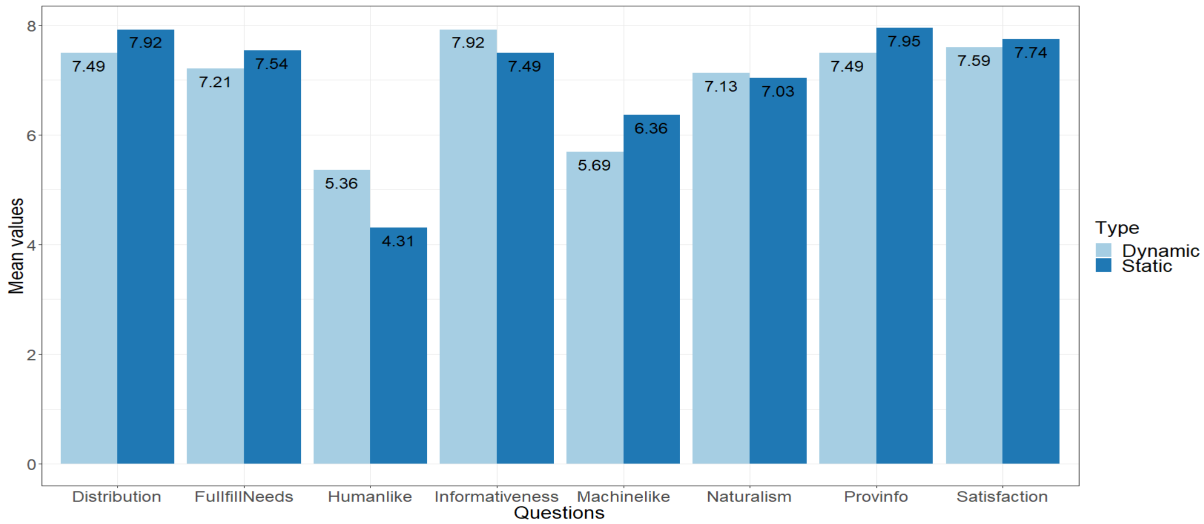

Figure 4: *Evaluation of feedback.*

References

J.L. Austin. 1970. ”Ifs and Cans”. *Philosophical Papers, second ed. London: Oxford University Press*, 1970: 205–232.

M.C. de Marneffe, S. Grimm, and C. Potts. 2009. Not a simple yes or no: Uncertainty in indirect answers. *Proceedings of SIGDIAL 2009*; 136-–143.

K. DeRose, R.E. Grandy. 1999. Conditional Assertions and ‘Biscuit’ Conditionals. *Noûs*, vol. 33, no. 3: 405-–420.

M.C. Frank, N. Goodman 2003. Predicting pragmatic reasoning in language games. *Science 336, (6084).*

M. Franke. 2007. The Pragmatics of Biscuit Conditionals. *Proceedings of the 16th Amsterdam colloquium*, 91–96.

M. Franke and G. Jäger. 2016. Probabilistic pragmatics, or why Bayes’ rule is probably important for pragmatics. *Zeitschrift für Sprachwissenschaft 35.*

J.F. Fulda 2009. Towards a unified theory of if’s – the theory of conditional elements: Further evidence from conditionally self-falsifying utterances. *Journal of Pragmatics, 41; 1440–1448.*

N. Green and S. Carberry. 1999. Interpreting and Generating Indirect Answers. *Computational LInguistics, 25 (3); 389–435.*

S. Janarthanam and O. Lemon. 2014. Adaptive Generation in Dialogue Systems Using Dynamic User Modeling. *Computaional Linguistics, 40 (4); 883–920.*

D. Jannach, M. Zanker, A. Felfernig, and F. Gerhard. 2011. *Recommender Systems.* Cambridge.

K. Oberauer, A. Weidenfeld, and K. Fischer 2007. What makes us believe a conditional? The roles of covariation and causality. *Thinking and Reasoning*, 13(4): 340-–369.

C. Potts, D. Lassiter, R. Levy, and M.C. Frank. 2016. Embedded implicatures as pragmatic inferences under compositional lexical uncertainty. *Journal of Semantics 33.*

M.E.A. Siegel 2006. Biscuit conditionals: Quantification over potential literal acts. *Linguistics and Philosophy, 29; 167–203.*

J. Stevens, A. Benz, S. Reuße, and R. Klabunde. 2016. Pragmatic question answering: A game-theoretic approach. *Data and Knowledge Engineering 106.*

E. Swanson. 2003. Biscuit Conditional and Common Ground. *Second North American Summer School in Language, Logic and Information Student Session Proceedings*.

H. Zeevat and H.-C. Schmitz. 2015. *Bayesian Natural Language Semantics and Pragmatics.* Springer.