Communicating an understanding of intention: Speech act conditionals and modified numerals in a Q/A system

Christoph Hesse[⋆], **Maryam Mohammadi**[†], **Maurice Langner**[†], **Judith Fischer**[†], **Anton Benz**[⋆], **Ralf Klabunde**[†]
[†]Ruhr Universität Bochum, Department of Linguistics
[⋆]Leibniz-Zentrum Allgemeine Sprachwissenschaft, Berlin
{maryam.mohammadi|maurice.langner|judith.fischer|ralf.klabunde}@rub.de
{benz|hesse}@leibniz-zas.de

Abstract

This paper aims at the generation of *speech act conditionals* (SACs) and modified numerals in answers in an interactive question answering system. SACs and modified numerals in indirect answers to a polar question do not only provide surplus information concerning the question, but also an indication why the answer might be relevant. The model we develop is based on a probabilistic approach to content determination that generates SACs and modified numerals based on an estimation about the user's requirements. Acceptability studies show that positive, negative and alternative SACs are appropriate answers in a real estate domain where users ask about properties of apartments they take interest in, and that modified numerals can be used strategically to mark qualitative differences between apartments.

1 Introduction

Speakers tend to answer polar questions indirectly if a direct answer would be inappropriate, be it for politeness reasons or since a simple *yes* or *no* is informationally underspecified. Since questions signal the inquirer's underlying requirement the listener does not have access to, his primary inferential task is to estimate what the most probable requirement of the inquirer might be. For example, if a client seeking an apartment asks a realtor *Is there a basement for the apartment?*, the realtor could assume the client needs the basement as a storage room. Hence, in case no basement is available, he might just answer *Storage rooms can be rented in the neighboring house.* In this case, the client will hopefully infer that no basement is available, and that the realtor assumes he needs the basement for storing items.

The assumed requirement could also be made explicit by the realtor. He could mention his assumption (*You seem to need a place for storing some of your belongings. Storage rooms can be rented in the house next-door*), or he uses a so-called speech act conditional (SAC): *If you need space for storing, storage rooms can be rented in the neighboring house.* Which one of these indirect answers is appropriate depends on discourse-dependent and stylistic reasons, but using an SAC enables one to express, by means of the antecedent, why the information in the consequent is discourse-relevant. SACs signal the link between the assumed requirement of the inquirer and its impact on the asserted information in the consequence.

Comparatively modified numerals such as *more than one hundred* can also be used to signal an understanding of requirements. The answer *There's a bus stop more than 4 miles away* to the question *Is there a bus stop nearby?* communicates that the realtor has understood that "nearby" signals an underlying decision problem where 'closer' is 'better' and that a train station 4 miles away is not 'nearby.'

This paper aims at the generation of SACs and modified numerals in indirect answers in a question answering system. In what follows, we will first describe the pragmatics of three types of speech act conditionals when used as answers to a polar question. Section 3 presents the probabilistic model of content determination for generating these SAC types and a procedure for generating modified numerals. It results in a decision tree that checks whether certain utilities are met in order to generate a suitable SAC vs. a simple *no* and *yes*, respectively. Section 4 goes into the empirical grounding of the model, and concludes with model evaluation.

2 The pragmatics of speech act conditionals and modified numerals as indirect answers

Speech act conditionals, often called “biscuit” conditionals in remembrance to Austin (1970), are conditionals like *there are biscuits on the sideboard if you want some*. These 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.

Contrary to classical conditionals, SACs 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. The antecedent seems to assure that the consequent is understood in a suitable way. For example, the SAC given above seems to legitimate the assertion that there are biscuits on the sideboard: The reason for mentioning the propositional content in the consequent is the assumption of the speaker that the addressee is hungry.

Two broad classes of SACs have been identified in the literature. The first class – the class we are interested in – constitute “problem-solving” SACs (Csipak, 2015), i.e. SACs indicating that the assertion of the consequent is in some way discourse-relevant. The second class are SACs that indicate a kind of topic shift in a conversation like *If I am being frank, you are looking tired*. However, we ignore SACs of this kind in this paper since they touch various aspects of topic organisation and politeness effects that are beyond the scope of our work.

SACs have received some attention in formal semantics and pragmatics (Franke, 2007; 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 their use as answers.

SACs can be used as indirect answers to polar questions. While indirect answers are typically negative ones since the surplus information given in that anwer is about alternatives, SACs can be used as indirect negative and positive answers.

SACs as indirect answers come with three different pragmatic functions. Their uses have different consequences in Q/A systems but should be modeled in a common way. For example, the question of the customer in the real estate domain *Is there a restaurant nearby?* can be answered by the real estate agent saying *If you enjoy eating out, there is an Italian restaurant in the vicinity*. The real estate agent might assume that the customer is able to infer that the Italian restaurant is the only restaurant nearby, and that the question was motivated by the customer’s general pleasure of eating out. In sum, this positive speech act conditional (PSAC) conveys: the answer is *yes*, the customer shall infer that the only restaurant nearby has been mentioned, and the supposed motivation of the customer for asking this question has been mentioned by the antecedent of the SAC.

Things are different with SACs that function as a negative answer to a polar question (NSAC). If the answer to the aforementioned question is *If you enjoy eating out, there is an Italian restaurant in the neighboring quarter*, it signals the following information: The answer is *no* and given the assumed requirement for the question as expressed by the antecedent of the SCA, 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. An ASAC as suitable answer to the aforementioned question would be *If you enjoy eating out, there is an Italian restaurant as well as a food court nearby*. By means of this answer, the system answers the question positively, but it offers two alternatives for the presumed requirement of eating out that are more or less equally probable.

The examples given so far suggest that requirements are directly tied to the attributes mentioned in the consequent (e.g., enjoying eating out – mentioning a neighbored restaurant), but the distance between the apartment under discussion and the target the client asks for results in an interesting order of alternatives to the target: *There’s an Italian restaurant in the neighboring quarter, but there’s also a food court less than 1 mile away* communicates that although the food court is closer to the apartment than the Italian restaurant, the restaurant is a better fit to the user’s requirement of eating out because otherwise it would not be worth mentioning the restaurant at all.

In sum, 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 for the presumed requirement.

3 The model

Our model is rooted in probability theory and generates SACs by strategic reasoning about possible requirements of the user. It follows 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; Qing et al., 2016; Zeevat and Schmitz, 2015), but it differs from these models in focusing on the generation task of determining the most probable content for solving the decision problem of the inquirer and realizing that content by a suitable answer. Our model constitutes the basis of a Q/A system where a client is looking for an apartment to rent and the system answers the user's questions about desirable attributes, either directly or indirectly. We presume that each question is motivated by an underlying requirement of the client. 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 client has. The client 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 client 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 agent's goal is helping the user to find an optimal object efficiently by anticipating the requirements r that are relevant to the user. Modified numerals should be generated when the anticipated requirements involve distance, for instance.

A discourse-sensitive and category-dependent parameter κ_c measures the amount of common ground concerning the requirements of category c. If κ_c exceeds some threshold, the generation of an SAC for category c is blocked since mentioning the assumed requirement would not be informative anymore.

3.1 The model in a nutshell

Suppose we are inferring requirements r which are at least ρ relevant to a question Q asking for attribute q. $\mathcal{M}$ is the set of requirements r which are more than ρ likely for a question attribute q.

$$\mathcal{M} = \{r|P(r|q) > \rho\} \tag{1}$$

In our database, a garden serves several requirements, among them are:

$$\begin{aligned} P(\text{enjoy greenery}\,|\,\text{garden}) &= 0.89 \\ P(\text{gardening}\,|\,\text{garden}) &= 0.85 \\ P(\text{dog walking}\,|\,\text{garden}) &= 0.54 \\ P(\text{smoking}\,|\,\text{garden}) &= 0.35 \end{aligned}$$

The requirements and their probabilities have been determined by experimental studies that will be described in the next section. With $\rho > .5$ we have $\mathcal{M} = \{\text{enjoy greenery, gardening, dog walking}\}$. The set $\mathcal{S}$ contains all pairs of attributes a and requirements r with $r \in \mathcal{M}$, and they are more than v useful to choose between alternatives a.

$$\mathcal{S} = \{(a,r)|r \in \mathcal{M} \wedge P(a|r)U(a,r) > v\} \tag{2}$$

$$
\begin{tikzcd}
 & r_1 \arrow[r] & a_1 \\
q \arrow[ur] \arrow[r] \arrow[dr] & r_2 \arrow[r] \arrow[ur] & a_2 \arrow[d, dotted, no head] \\
 & r_3 \arrow[uur] \arrow[d, dotted, no head] & a_m \\
 & r_n &
\end{tikzcd}
$$

Figure 1: Example inference

$P(a|r)$ is the probability that attribute a fulfills requirement r. The aim is to determine a utility of attribute a for requirement r. $U(a,r)$ might determine how useful a is for requirement r. For example, a balcony is less useful for gardening than a garden, but when a user asks for a garden for an apartment which has no garden, a balcony is still a good alternative because it can be used for gardening, too. If a requirement hinges on a numerical property such as distance d, $U \propto \frac{1}{1+d}$. In our example, $\mathcal{S}$ is the set $\{$(garden, enjoy greenery), (garden, gardening), (garden, dog walking), (balcony, gardening)$\}$.

Hence, the first task is to determine the set of requirements for question q. Being an empirical task, we performed studies via Mechanical Turk to determine the requirements that have subsequently been represented in the database of our Q/A system. In a second step, we infer the attributes true of the apartment (does it have a balcony, a garden, what public transports are in the vicinity etc.) which best fit the requirements. Question q and attribute a mentioned in the answer are thus linked indirectly via relevant requirements r.

In our example, a question about a garden triggers three requirements which are more than $\rho = 0.5$ likely: r_1 = enjoy greenery, r_2 = gardening, and r_3 = dog walking. There are three pairs in $\mathcal{S}$ where attribute a_1 = garden meets all three requirements and only one pair where the attribute balcony meets requirement r_2. We can represent the competition between which attribute should be mentioned in an answer to a question q as in Figure 1.

For example, if an apartment has both a garden and a balcony, a garden meets all potential requirements better than the balcony. So a sales agent who is ignorant about what a user's true intentions for asking about a garden are, should not mention the balcony. Only if the apartment has a balcony but no garden, is the balcony a valid alternative.

The database contains numerical properties of objects o such as distance to the apartment. The numerical properties contribute to an object's expected utility through a quality coefficient. For instance, the greater the distance between o and the apartment, the lower the expected utility. The system generates modified numerals by considering the distance of all o from the apartment, rounding the distance estimates to a contextually appropriate level of precision, preferring fractal reference points observed by Jansen and Pollmann (2001) and Dehaene and Mehler (1992), modifying them by the comparative quantifier "more than". The system then translates this information into natural language by using simple sentence templates like "There's a(n) X [more than n [unit]] away."

3.2 The model in detail

The general inferential task outlined in the previous subsection will now be described in more detail to explain how the Q/A system infers the necessary information for generating our three types of SACs as indirect answers.

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 the most relevant question for expressing a requirement:

$$P(r|q) = \frac{P(q|r) \times P(r)}{\sum_{r' \in R^q} P(q|r') \times P(r')} \tag{3}$$

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.

In general, the system has to anticipate the underlying decision problem that induces the client 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:

$$B(r|q) = 1,\ \text{if}\ r \in R^q;\ \text{else}\ 0 \tag{4}$$

Questions q, as well as attributes a, are associated with a set of requirements R^q. Furthermore, questions are about attributes of some subdomain c of the overall domain of apartment attributes, for example interiors or transportation connections.

Since the requirement of the client 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 possible requirements R^a of the attribute a_c:

$$EB(r|a_c, R^q) = \sum_{r \in R^q} P(r|a_c) \times B(r|a_c) \tag{5}$$

Attribute a_c can be the attribute the user is asking for (i.e., $a_c = q_c$). 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_c of category c with question attribute q_c, and q_c is not true of the apartment, we consider only the requirements R^q for the original question q_c.

The expected utility of r and q of category c can be determined by:

$$EU(r, q_c) = EB(r|q_c, R^q) - \kappa_c \tag{6}$$

κ_c is a dialogue-sensitive cost for realizing the category-dependent 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.

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 results in $EU < 0$, which blocks the generation of an SAC. An SAC is only generated if $EU > 0$, 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 for generating the speech act conditional. The pseudocode of the decision tree for the generation of direct answers and SACs as indirect answers is given in Table 1.

If attribute q_c is true of a flat f, we determine whether there is some requirement r in the set of possible requirements R^q which triggers the expected utility of r and q_c to be positive (> 0). If this is not the case, none of the requirements are 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.

If attribute q_c is false of flat f, the model checks whether there is some alternative attribute a_c satisfying requirements r such that the expected utility $EU(r, a_c)$ is positive. If $EU(r, a_c)$ is negative, the decision

Algorithm 1 An algorithm for determining 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 $P(r|p)$, a user question providing the attribute q the user is asking for, threshold τ

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

while user response ≠ 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
  $\kappa_c := \kappa_c + \sum_{i=1}^{n} P(r_i^c|a_i)$ (update of κ_c values)

Table 1: Content determination for SACs

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 that represents the average of all $P(r_i|a)$:

$$\tau = \frac{\sum_i P(r_i|a)}{|(r,a)|} \tag{7}$$

with $|(r,a)|$ the number of all requirement-answer combinations. This value determines whether a requirement is probable enough to be worth the effort made to utter it. 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.

4 Empirical grounding

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. The studies were designed to test the acceptability of SACs as indirect answers by users of the system. The first study was performed to determine the input probabilities for the 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 whether there is a possibility of talking about them during a conversation in a sales setting. In order to receive the probabilities of both interlocutors in a dialogue, we divided the participants into two groups to judge as a customer (54) or a real estate agent (59).

The second study tested the acceptability of the different types of SACs as indirect answers. Participants took on the role of either customer or realtor. 241 out of 250 subjects (119 as customers and 122 as realtors) successfully participated in the experiment. Participants were shown 5 questions such as *Are there any restaurants near the apartment?* and for each question they were shown 5 possible answers (direct yes/no, and the 3 SACs) and had to rate the acceptability of each answer on a scale from

0 to 100 (Figure 2, left panel). One-way ANOVA found significant variation among the 5 types of answers ($F(4, 2405) = 217.3, p < 0.001$) and Tukey HSD revealed that the significance was due to the low acceptability of NSAC while PSAC and ASAC received similar ratings to direct answers.

The third study investigated the acceptability of NSACs by eliciting how well an object in the database fulfills users' requirements r. Our assumption was that an object a should only be presented in an NSAC as an alternative to the object the user asks q if it fulfills the requirements better ($P(a|r) > P(q|r)$) or at least as good as q ($P(a|r) = P(q|r)$). 49 participants were recruited and found NSACs less acceptable when a was worse at fulfilling r than q ($P(a|r) < P(q|r)$, Figure 2, right panel.

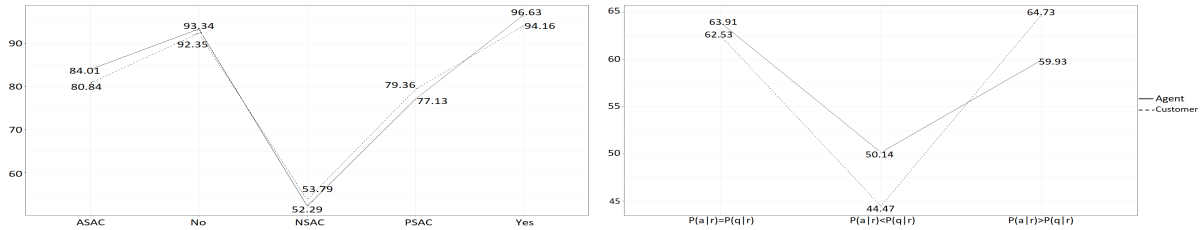


Figure 2: ***Left:*** *Acceptability of answer types.* ***Right:*** *NSAC meeting requirement.*

5 Overall evaluation of the system

The Q/A system described in this paper and used for the experimental studies is available at `https://www.linguistics.rub.de/app/pragsales/biscuit`. We compared this Q/A system that is able to generate SACs dynamically with a baseline system that generates direct answers only. This baseline system is our original system with high κ_c 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 prompted to ask questions about a flat for her/his friend. The participants were informed about requirements for their friend. By means of their questions, they have 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 questions, which is obviously not sufficient for determining language efficiency. The questions were answered with SACs and direct *yes/no* answers. At the end of the experiment participants answered 10 questions 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.

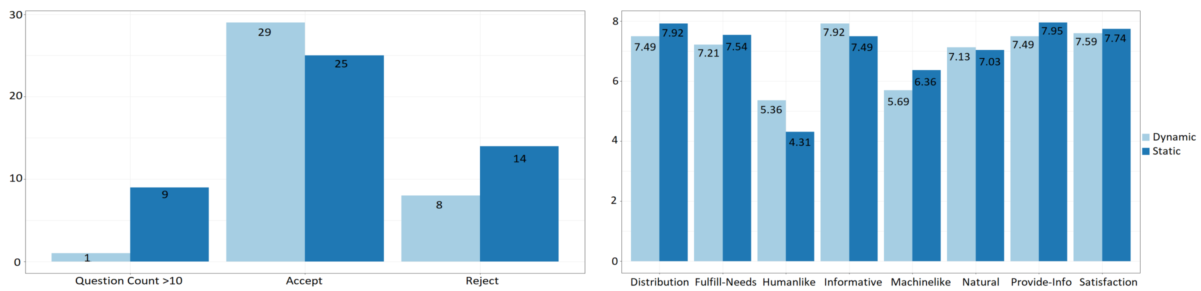


Figure 3: *Comparison of dynamic and static system.*

Figure 3 (left panel) 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 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 participant ratings did not show a significant difference between both systems (see Figure 3, right panel). 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 scored better on human-like 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.

We conducted a separate evaluation study for modified numerals in order to distinguish their contribution to indirect answers from the contribution of SACs. In evaluating the generation of numerals, we start from the assumption that a system which can communicate qualitative differences is one which can make things that are objectively speaking not different *seem* like they are. Participants are led to believe they will view five different properties for a friend who is looking to buy a house in Brooklyn, New York, but in actuality two of the houses are identical. Participants are misled by the realtor (our system) who generates a different numeral for the two identical houses with respect to one attribute—the distance to the nearest subway station—so as to make it seem like there is a qualitative difference between the two. For one, the system will generate a vague expression using a comparatively modified numeral ("more than 1 mile"), for the other, it will generate an exact unmodified numeral ("1.2 miles" or "1.7 miles"). The unmodified numeral is the objective distance rounded to one decimal. This way, we test whether "1.2 miles" or "1.7 miles" comes closer to participants' expected reading of "more than 1 mile," cf. the normative versus transgressive reading in the approach by Anscombre and Ducrot (1983).

We recruited 100 participants with U.S. IP addresses via Amazon's Mechanical Turk, 76 successfully completed the study. When participants ask about a subway station near house 4, they are told it is "more than 1 mile away." When they pose the same question for house 5, the agent will give them an exact distance. The 50 participants in the first version of the study are told the subway station is "1.2 miles away" from house 5; those in the second version are told the station is "1.7 miles away."

Participants are asked to select their favorite house. The left graph in Figure 4 shows that the majority of participants shortlist house 4 and house 5, the two identical properties, but they favor house 4 to house 5 at a ratio of 2:1 when told the subway is "1.2 miles away" and 3:1 when told it is "1.7 miles away." After submitting the shortlist to their friend, the true distance of the subway station qualified as "more than 1 mile" was revealed. When participants learned the true distance, they indicated on a 7-point Likert scale whether they felt they had been misled (-3) or whether felt an imprecise numeral was appropriate (+3). The graph on the right in Figure 4 shows that, on average, participants who favored house 4 and learn that "more than 1 mile away" really meant "1.7 miles away" (red) give ratings which are 1.561 lower then participants who found out it meant "1.2 miles" (blue).

We fitted a linear mixed effects model to the Likert ratings with participants' group membership as fixed effect and by-subject variation as a random effect with random intercepts and random slopes. According to this model, group membership predicts a significant difference in ratings of 1.561 (SE = 0.535, $t = -2.917$, $p = 0.0054$), the lowering actually observed. A null model without the fixed effect only accounted for a lowering of 0.06. We conclude that our system successfully deceived participants into perceiving a qualitative difference where there was none.

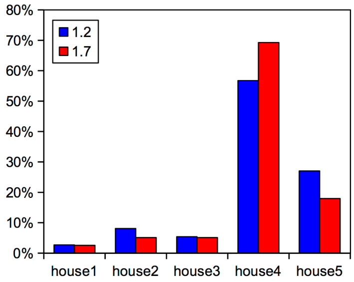
Acknowledgments

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

References

J-C. Anscombre and O. Ducrot. 1983. *L' argumentation dans la langue*. Mardaga, Bruxelles.

J.L. Austin, 1970. *Ifs and cans*, volume 2. Oxford University Press.



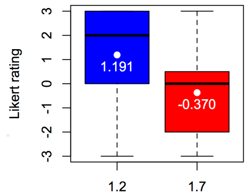


Figure 4: ***Left:** Participants' first choice (%).* ***Right:** Likert rating upon learning true distance.*

E. Csipak. 2015. *Free factive subjunctives in German*. Ph.D. thesis, Göttingen University.

Stanislas Dehaene and Jacques Mehler. 1992. Cross-linguistic regularities in the frequency of number words. *Cognition*, 43(1):1–29. DOI: 10.1016/0010-0277(92)90030-L.

M.C. Frank and N. Goodman. 2012. Predicting pragmatic reasoning in language games. *Science*, 336 (6084).

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

M. Franke. 2007. The pragmatics of biscuit conditionals. In Paul Dekker Maria Aloni and Floris Roelofsen, editors, *Proceedings of the 16th Amsterdam Colloquium*.

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.

Carel J. M. Jansen and M. M. W. Pollmann. 2001. On Round Numbers: Pragmatic Aspects of Numerical Expressions. *Journal of Quantitative Linguistics*, 8(3):187–201. DOI: 10.1076/jqul.8.3.187.4095.

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

C. Qing, N.D. Goodman, and D Lassiter. 2016. A rational speech-act model of projective content. In Anna Papafragou, Dan Grodner, Dan Mirman, and John C. Trueswell, editors, *Proceedings of the 38th annual meeting of the Cognitive Science Society (CogSci-2016)*.

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

J.S. Stevens, S. Reue, A. Benz, and R. Klabunde. 2015. A strategic reasoning model for generating alternative answers. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*.

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

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