Predicting the resolution of referring expressions from user behavior

Nikos Engonopoulos[1] **Martín Villalba**[1] **Ivan Titov**[2] **Alexander Koller**[1]
[1]University of Potsdam, Germany [2]University of Amsterdam, Netherlands
{nikolaos.engonopoulos, martin.villalba}@uni-potsdam.de
titov@uva.nl, koller@ling.uni-potsdam.de

Abstract

We present a statistical model for predicting how the user of an interactive, situated NLP system resolved a referring expression. The model makes an initial prediction based on the meaning of the utterance, and revises it continuously based on the user's behavior. The combined model outperforms its components in predicting reference resolution and when to give feedback.

1 Introduction

Speakers and listeners in natural communication are engaged in a highly interactive process. In order to achieve some communicative goal, the speaker will perform an utterance which they believe has a high chance of achieving that goal. They will then monitor the listener's behavior to see whether this goal is actually being achieved. This process is a core part of what is commonly called *grounding* in the dialogue literature (see e.g. (Clark, 1996; Traum, 1994; Paek and Horvitz, 1999; Hirst et al., 1994)). Interactive computer systems that are to carry out an effective and efficient conversation with a user must model this grounding process, and should ideally respond to the user's observed behavior in real time. For instance, if the user of a pedestrian navigation system takes a wrong turn, the system should interpret this as evidence of misunderstanding and bring the user back on track.

We focus here on the problem of predicting how the user has resolved a *referring expression (RE)* that was generated by the system, i.e. a noun phrase that is intended to identify some object uniquely to the listener. A number of authors have recently offered statistical models for parts of this problem. Golland et al. (2010) and Garoufi and Koller (2011) have presented log-linear models for predicting how the listener will resolve a given RE in a given scene; however, these models do not update the probability model based on observing the user's reactions. Nakano et al. (2007), Buschmeier and Kopp (2012), and Koller et al. (2012) all predict what the listener understood based on their behavior, but do not consider the RE itself in the model. The models of Frank and Goodman (2012) and Vogel et al. (2013) aim at explaining the effect of implicatures on the listener's RE resolution process in terms of hypothesized interactions, but do not actually support a real-time interaction between a system and a user.

In this paper, we show how to predict how the listener has resolved an RE by combining a statistical model of RE resolution based on the RE itself with a statistical model of RE resolution based on the listener's behavior. To our knowledge, this is the first approach to combine two such models explicitly. We consider the RE grounding problem in the context of interactive, situated natural language generation (NLG) for the GIVE Challenge (Koller et al., 2010a), where NLG systems must generate real-time instructions in virtual 3D environments. Our evaluation is based on interaction corpora from the GIVE-2 and GIVE-2.5 Challenges, which contain the systems' utterances along with the behavior of human hearers in response to these utterances. We find that the combined model predicts RE resolution more accurately than each of the two component models alone. We see this as a first step towards implementing an actual interactive system that performs human-like grounding based on our RE resolution model.

Figure 1: An example scene in the GIVE environment.

2 Problem definition

In the GIVE Challenge, an interactive NLG system faces the task of guiding a human instruction follower (IF) through a treasure-hunt game in a virtual 3D environment (see Fig. 1). To complete the task, the IF must press a number of buttons in the correct order; these buttons are the colored boxes in Fig. 1, and are scattered all over the virtual environment. The IF can move around freely in the virtual environment, but has no prior knowledge about the world. The NLG system's task is to guide the IF towards the successful completion of the treasure-hunt task. To this end, it is continuously being informed about the IF's movements and visual field, and can generate written utterances at any time. As a comparative evaluation effort, the GIVE Challenges connected NLG systems to thousands of users over the Internet (see e.g. Koller et al. (2010a) for details).

Many system utterances are *manipulation instructions*, such as "press the blue button", containing an RE in the form of a definite NP. We call a given part of an interaction between the system and the IF an *episode* of that interaction if it starts with a manipulation instruction, ends with the IF performing an *action* (i.e., pressing a button), and contains only IF movements and no further utterances in between. Not all manipulation instructions initiate an episode, because the system may decide to perform further utterances (not containing REs) before the IF performs their action. An NLG system will choose the RE for an instruction at runtime out of potentially many semantically valid alternatives ("the blue button", "the button next to the chair", "the button to the right of the red button", etc.). Ideally, it will predict which of these REs has the highest chance to be understood by the IF, given the current scene, and utter an instruction that uses this RE.

After uttering the manipulation instruction, the system needs to ascertain whether the IF understood the RE correctly, i.e. it must engage in grounding. A naive grounding mechanism might wait until the IF actually presses a button and check whether it was the right one. This is what many NLG systems in the GIVE Challenges actually did. However, this can make the communication ineffective (IF performs many useless actions) and risky (IF may press the wrong button and lose). Thus, it is important that the system updates its prediction of how the IF resolved the RE continuously by observing the IF's behavior, *before* the actual button press. For instance, if the IF walks towards the target, this might reinforce the system's belief in a correct understanding; turning away or exiting the room could be strong evidence of the opposite. The system can then exploit the updated prediction to give the IF feedback ("no, the *blue* button") to prevent costly mistakes.

We address these challenges by estimating the probability distribution over the possible objects to which the IF may resolve the RE. We then update this distribution in real time by observing the IF's movements. More specifically, assume that a system tries to refer to some object a^* among some set A of available objects. Given an RE r generated for a^* at time t_0, the state of the world s at t_0, and the observed behavior $\sigma(t)$ of the user at $t \geq t_0$, we estimate the probability $p(a|r, s, \sigma(t))$ that the user resolved r to an object $a \in A$. When generating the instruction, an optimal NLG system will use the RE r that maximizes $p(a^*|r, s, \sigma(t_0))$. It can then track $p(a|r, s, \sigma(t))$ for time points $t > t_0$ throughout the episode, and generate feedback when $p(a'|r, s, \sigma(t))$ exceeds $p(a^*|r, s, \sigma(t))$ for some $a' \neq a^*$; that is, when the updated probability distribution predicts that the IF resolved r to an incorrect button.

3 A model of RE resolution

In order to model the distribution over possible objects, we assume the following generative story: when receiving an instruction containing an RE r at a given world state s, the IF resolves it to an object a; depending on the object a, the IF then moves towards it, exhibiting behavior σ. These assumptions correspond to the following factorization:

$$p(a, \sigma|r, s) = p(\sigma|a)p(a|r, s)$$

The posterior probability distribution over objects a can be obtained by applying the Bayes rule and using the above assumptions:

$$p(a|r,s,\sigma) \propto p(a|r,s)p(a|\sigma)/p(a)$$

For simplicity, we assume a uniform $p(a)$ over all objects in a world. We can thus represent $p(a|r,s,\sigma)$ as the normalized product of a *semantic* model $p_{sem}(a|r,s)$ and an *observational* model $p_{obs}(a|\sigma)$. We use log-linear models for both, and train them separately. The feature functions we use only consider general properties of objects (such as color and distance), and not the identity of the objects themselves. This means that we can train a model on one virtual environment (containing a certain set of objects), and then apply the model to another virtual environment, containing a different set of objects.

Semantic model The semantic model estimates for each object a in the environment the initial probability $p_{sem}(a|r,s)$ that the IF will understand a given RE r uttered in a scene s as referring to a. It represents the meaning of r, contextualized to s, and is only ever evaluated at the time t_0 of the utterance. The features used by this model are:

- **Semantic features** aim to encode whether r is a good description of a. *IsColorModifying* evaluates to 1 if a's color appears as an adjective modifying the head noun of r, e.g. "the blue button". *IsRelPosModifying* evaluates to 1 if a's relative position to the IF is mentioned as an adjective in r, e.g. "the left button".
- **Confusion features** capture the hypothesis that the IF may be confused by the description of a landmark when resolving the RE; e.g. an RE like "the button next to the red button" might confuse the IF into pressing a red button, rather than the one meant by the system. These are the same features as in the Semantic case, but looking for modifier keywords in the entire RE, including the head.
- **Salience features** account for the fact that an IF is more likely to resolve r to a if a was visually salient in s. *IsVisible* evaluates to 1 if a is visible to the IF in s. *IsInRoom* evaluates to 1 if the IF and a are in the same room. *IsTargetInFront* evaluates to 1 if the *angular distance* towards a, i.e. the absolute angle between the camera direction and the straight line from the IF to a, is less than $\frac{\pi}{4}$. *VisualSalience* approximates the visual salience of Kelleher and van Genabith (2004), a weighted count of the number of pixels on which a is rendered (pixels near the center of the screen have higher weights).

Observational model The observational model estimates for each object a the probability $p_{obs}(a|\sigma)$ that the IF will interact with a, given the IF's recent behavior $\sigma(t) = (\sigma_1, \ldots, \sigma_n)$, where σ_i is the state of the world at time $t - (i-1) \cdot 500\text{ms}$, and $n \geq 1$ is the length of the observed behavior. p_{obs} is constantly re-evaluated for times $t > t_0$ as the IF moves around. p_{obs} uses the following features:

- **Linear distance features** assume that the closest button is also the one the IF understood. *InRoom* returns the number of frames σ_i in σ in which the IF and a are in the same room. *ButtonDistance* returns the distance between the IF and a at σ_1 divided by a constant such that the result never exceeds 1. If a is neither in the same room nor visible, the feature returns 1.
- **Angular distance features** analyze the direction in which the IF looks. *TargetInFront* returns the angular distance towards a at σ_1. *AngleToTarget* returns *TargetInFront* divided by π, or 1 if a is neither in the same room nor visible. *LinearRegAngleTo* applies linear regression to a list of observed angular distances towards a over all frames σ_i, and returns the slope of the regression as a measure of variation. Negative values indicate that the IF turned towards a, while positive values mean the opposite. If a is neither visible nor in the same room as the IF at σ_i, the angle is set to π.
- **Combined distance feature**: a weighted sum of linear and angular distance towards a, called *overall distance* in Koller et al. (2012).
- **Salience features** capture visual salience and its change over time. Defining VS_i as the result of applying the p_{sem} feature *VisualSalience* to σ_i and a, *LastVisualSalience* returns VS_n. *LinearRegVisualSalience* applies linear regression to all values VS_i and returns the slope as a measure of change in salience. *VisualSalienceSum* returns $(\Sigma_{i=1}^{n}\text{VS}_i) * \text{VS}_1$. This emphasizes the contribution of VS_1, which we assume is the

most reliable predictor of the IF's intentions.

- **Binary features** aim to detect concrete behavior patterns: *LastIsVisible* applies the p_{sem} feature *IsVisible* to σ_1, and *IsClose* evaluates to 1 if the IF is close enough and correctly oriented to manipulate a in the GIVE environment at σ_1.

4 Evaluation

Data We evaluated our model using data from the GIVE-2 (Koller et al., 2010b) and the GIVE-2.5 Challenges (Striegnitz et al., 2011), obtained from GIVE Organizers (2012). These datasets constitute *interaction corpora*, in which the IF's activities in the virtual environment were recorded along with the utterances automatically generated by the participating NLG systems. The data consists of 1833 games for GIVE-2 and 687 games for GIVE-2.5.

To extract training data for our model from the GIVE-2.5 data, we first identified moments in the recorded data where the IF pressed a button. From these, we discarded all instances from the tutorial phase of the GIVE game and those that happened within 200 ms after the previous utterance, as these clearly didn't happen in response to it. This yielded 6478 training instances for p_{obs}, each consisting of σ at 1 second before the action, and the button a which the IF pressed. We chose $n = 4$ for representing σ, except to ensure that the features only considered IF behavior that happened in response to an utterance. We achieved this by reducing n for the first few frames after each utterance, such that the time of σ_n was always after the time of the utterance. Finally, we selected those instances which are episodes in the sense of Section 2, i.e. those in which the last utterance before the action contained an RE r. This gave us 3414 training instances for p_{sem}, each consisting of a, r, the time t_0 of the utterance, and the world state s at time t_0.

We obtained test instances from the GIVE-2 data in the same way. This yielded 5028 instances, each representing an episode. We chose GIVE-2 for testing because the mean episode length is higher (3.3s, vs. 2.0s in GIVE-2.5), thus making the evaluation more challenging. Feature selection was done using the training data and a similar dataset from Koller et al. (2012). Note that the test data and training data are based on distinct sets of three virtual environments each, and were obtained with different NLG systems and users. This demonstrates the ability of our model to generalize to unseen environments.

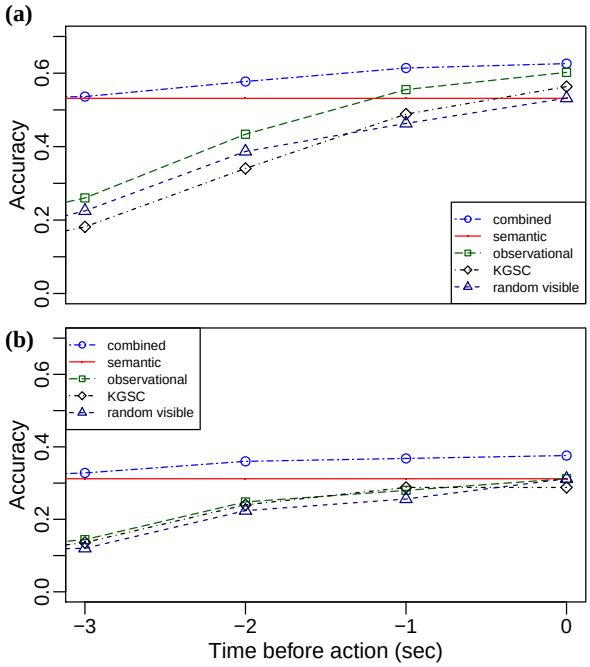

Figure 2: Prediction accuracy for (a) all episodes, (b) unsuccessful episodes as a function of time.

An example video showing our models' predictions on some training episodes can be found at `http://tinyurl.com/re-demo-v`.

Prediction accuracy We first evaluated the ability of our model to predict the button to which the IF resolved each RE. For each test instance $\langle r, s, \sigma, a \rangle$, we compare the object returned by $\arg\max_a p(a|r, s, \sigma(t))$ to the one manipulated by the IF. We call the proportion of correctly classified instances the *prediction accuracy*.

Fig. 2a compares our model's prediction accuracy to that of several baselines. We plot prediction accuracy as a function of the time at which the model is queried for a prediction, by evaluating at 3s, 2s, 1s, and 0s before the button press. The graph is based on the 2094 test instances with an episode length of at least three seconds, to ensure that results for different prediction times are comparable. As expected, prediction accuracy increases as we approach the time of the action. Furthermore, the combined model outperforms both p_{sem} and p_{obs} reliably. This indicates that the component models pro-

vide complementary useful information. Our model also outperforms two more baselines: *KGSC* predicts that the IF will press the button with the minimal *overall distance*, which is the distance metric used by the "movement-based system" of Koller et al. (2012); *random visible* selects a random button from the ones that are currently visible to the IF.

The fact that this last baseline does not approach 1 at action time suggests that multiple buttons tend to be visible when the IF presses one, confirming that the prediction task is not trivial.

Correctly predicting the button that the IF will press is especially useful, and challenging, in those cases where the IF pressed a different button than the one the NLG system intended. Fig. 2b shows a closer look at the 125 unsuccessful episodes of at least three seconds in the test data. These tend to be hard instances, and thus as expected, prediction accuracy drops for all systems. However, by integrating semantic and observational information, the combined model compensates better for this than all other systems, with an accuracy of 37.6% against 31.2% for each individual component.

Feedback appropriateness Second, we evaluated the ability of our model to predict whether the user misunderstood the RE and requires feedback. For all the above models, we assumed a simple feedback mechanism which predicts that the user misunderstood the RE if $p(a') - p(a^*) > \theta$ for some object $a' \neq a^*$, where θ is a confidence threshold; we used $\theta = 0.1$ here. We can thus test on recorded data in which no actual feedback can be given anymore.

We evaluated the models on the 848 test episodes of at least 3s in which the NLG systems logged the button they tried to refer to. The results are shown in Fig. 3 in terms of F1 measure. Here precision is the proportion of instances in which the IF pressed the wrong button (i.e., where feedback should have been given) among the instances where the model actually suggested feedback. Recall is the proportion of instances in which the model suggested feedback among the instances where the IF pressed the wrong button. Again, the combined model outperforms its components and the baselines, primarily due to increased recall. The difference is particularly pronounced early on, which would be useful in giving timely feedback in an actual real-time system.

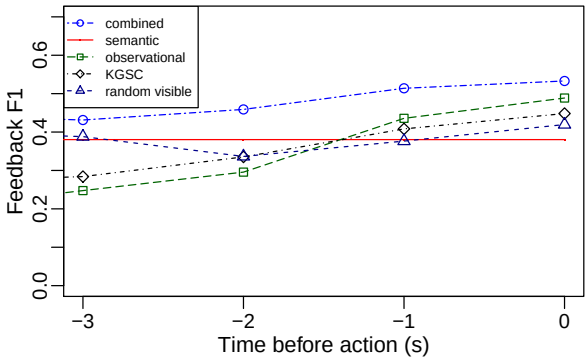

Figure 3: Feedback F1-measure as a function of time.

5 Conclusion and future work

We presented a statistical model for predicting how a user will resolve the REs generated by an interactive, situated NLG system. The model continuously updates an initial estimate based on the meaning of the RE with a model of the user's behavior. It outperforms its components and two baselines on prediction and feedback accuracy.

Our model captures a real-time grounding process on the part of the interactive system. We thus believe that it provides a solid foundation for detecting misunderstandings and generating suitable feedback in an end-to-end dialogue system. We have presented our model in terms of a situated dialogue setting, where clues about what the hearer understood can be observed directly. However, we believe that the fundamental mechanism should apply to other domains as well. This would amount to finding observable linguistic and non-linguistic clues of hearer understanding that can be used as features of p_{obs}.

The immediate next step for future research is to extend our model to an implemented end-to-end situated NLG system for the GIVE Challenge, and evaluate whether this actually improves task performance. This requires, in particular, to compute the RE that is optimal with respect to p_{sem}. We will furthermore improve p_{obs} by switching to a more temporally dynamic probability model.

Acknowledgments. We thank Konstantina Garoufi and the anonymous reviewers for their insightful comments and suggestions. The first two authors were supported by the SFB 632 "Information Structure"; Titov's work was supported by the Cluster of Excellence at Saarland University.

References

Hendrik Buschmeier and Stefan Kopp. 2012. Adapting language production to listener feedback behaviour. In *Proceedings of the Interdisciplinary Workshop on Feedback Behaviors in Dialog*.

Herbert C. Clark. 1996. *Using Language*. Cambridge University Press.

Michael C. Frank and Noah D. Goodman. 2012. Predicting pragmatic reasoning in language games. *Science*, 336(6084):998.

Konstantina Garoufi and Alexander Koller. 2011. Combining symbolic and corpus-based approaches for the generation of successful referring expressions. In *Proceedings of the 13th European Workshop on Natural Language Generation (ENLG)*.

GIVE Organizers. 2012. Give challenge website: Corpora. `http://give-challenge.org/research/page.php?id=corpora`.

Dave Golland, Percy Liang, and Dan Klein. 2010. A game-theoretic approach to generating spatial descriptions. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Graeme Hirst, Susan McRoy, Peter Heeman, Philip Edmonds, and Diane Horton. 1994. Repairing conversational misunderstandings and non-understandings. *Speech Communications*, 15:213–229.

J. D. Kelleher and J. van Genabith. 2004. Visual salience and reference resolution in simulated 3-D environments. *Artificial Intelligence Review*, 21(3).

Alexander Koller, Kristina Striegnitz, Donna Byron, Justine Cassell, Robert Dale, Johanna Moore, and Jon Oberlander. 2010a. The First Challenge on Generating Instructions in Virtual Environments. In E. Krahmer and M. Theune, editors, *Empirical Methods in Natural Language Generation*, number 5790 in LNCS, pages 337–361. Springer.

Alexander Koller, Kristina Striegnitz, Andrew Gargett, Donna Byron, Justine Cassell, Robert Dale, Johanna Moore, and Jon Oberlander. 2010b. Report on the Second NLG Challenge on Generating Instructions in Virtual Environments (GIVE-2). In *Proceedings of the 6th International Natural Language Generation Conference (INLG)*.

Alexander Koller, Konstantina Garoufi, Maria Staudte, and Matthew Crocker. 2012. Enhancing referential success by tracking hearer gaze. In *Proceedings of the 13th Annual SIGdial Meeting on Discourse and Dialogue (SIGDIAL)*, Seoul.

Yukiko Nakano, Kazuyoshi Murata, Mika Enomoto, Yoshiko Arimoto, Yasuhiro Asa, and Hirohiko Sagawa. 2007. Predicting evidence of understanding by monitoring user's task manipulation in multimodal conversations. In *Proceedings of the ACL 2007 Demo and Poster Sessions*.

Tim Paek and Eric Horvitz. 1999. Uncertainty, utility, and misunderstanding: A decision-theoretic perspective on grounding in conversational systems. In *AAAI Fall Symposium on Psychological Models of Communication in Collaborative Systems*.

Kristina Striegnitz, Alexandre Denis, Andrew Gargett, Konstantina Garoufi, Alexander Koller, and Mariet Theune. 2011. Report on the Second Second Challenge on Generating Instructions in Virtual Environments (GIVE-2.5). In *Proceedings of the 13th European Workshop on Natural Language Generation (ENLG)*.

David Traum. 1994. *A computational theory of grounding in natural language conversation*. Ph.D. thesis, University of Rochester.

Adam Vogel, Christopher Potts, and Dan Jurafsky. 2013. Implicatures and nested beliefs in approximate Decentralized-POMDPs. In *Proceedings of ACL*.