Automatic Annotation of Referring Expression in Situated Dialogues

Niels Schütte
Technological University Dublin

John D. Kelleher
Technological University Dublin, john.d.kelleher@tudublin.ie

Brian Mac Namee
Technological University Dublin, brian.macnamee@tudublin.ie

Recommended Citation

Schutte, N., Kelleher, J. D. and Mac Namee, B. Automatic Annotation of Referring Expression in Situated Dialogues. International journal of computational linguistics and applications Vol. 2, No. 1-2, Jan-dec 2011, PP. 175-190. doi:10.21427/D7HW5Q

Automatic Annotation of Referring Expression in Situated Dialogues

Abstract. To apply machine learning techniques to the production and interpretation of natural language, we need large amounts of annotated language data. Manual annotation, however, is an expensive and time consuming process since it involves human annotators looking at the data and explicitly adding information that is implicitly contained in the data, based on their judgment. This work presents an approach to automatically annotating referring expressions in situated dialogues by exploiting the interpretation of language by the participants in the dialogue. We associate instructions concerning objects in the environment with automatically detected events involving these objects and predict the referents of referring expressions in the instructions on the basis of the objects affected by the events. We judge the reliability of these predictions based on the temporal and textual distance between instruction and event. We apply our approach to an annotated corpus and evaluate the results against human annotation. The evaluation shows that the approach can be used to accurately annotate a large proportion of the utterances in the corpus dialogues and highlight those utterances for which human annotation is required, thus reducing the amount of human annotation required.

Keywords: reference resolution, situated dialogue

1 Introduction

We present an approach to automatically annotating *referring expressions* in *situated dialogues*. A referring expression [1, Ch. 18] is an expression that occurs in natural language that is used to denote some kind of object that is discussed. For example in the sentence "Bob ate an apple", "Bob" is a referring expression that denotes some person named Bob, and "an apple" is a referring expression that denotes some apple.

The object that is being referred to is called the *referent* of the referring expression. An *anaphoric* referring expression is a referring expression that refers back to an object that has already been mentioned in the dialogue and is therefore in the linguistic context of the dialogue. An *exophoric* referring expression is a referring expression that refers to an object that has not previously been mentioned in the dialogue but that exists in some other context of the dialogue (e.g. the visual context). The process of *referring expression resolution* is the process of identifying the referents of referring expressions.

A situated dialogue is a conversation between at least two participants that takes places in an environment that is actively discussed as part of the dialogue. A typical example a of situated dialogue is a navigation task where one participant has to give instructions to a second participant to move through the environment the dialogue is situated in. Exophoric referring expressions are particularly common in this domain.

A computer system that participates in situated dialogues has to be able to resolve and produce exophoric referring expressions. There exist a number of approaches to this problem that can broadly be categorized as rule-based approaches and machine-learning (ML) based approaches. Rule-based approaches use a number of (generally hand crafted) rules to to perform the task. Salmon-Alt and Romary [6] describe a rule-based approach to resolving reference in purely linguistic domains. They also present a rule-based approach to resolving reference in a multimodal domain [3].

ML based approaches on the other hand do not rely on prefabricated rules but set out to learn behaviour that is presented in the form of examples. Using ML is particularly attractive for dealing with referring expressions because using ML opens up the possibility to learn and discover strategies used by humans directly from data, which may be difficult to identify by introspection or manual analysis.

Supervised ML is a form of ML where algorithms learn a function that maps from inputs to outputs. Such algorithms require as training data a set of examples in which inputs are associated with the expected output. Consequently, in order to train a ML algorithm to interpret or produce referring expressions, the algorithm requires a training set of examples that link spoken references to their intended referents in the world and that, furthermore, describe the conditions under which the reference was produced. These conditions may for example include the set of visible objects, the spatial relation of the speaker towards those objects and a records of previous references made by the speaker.

These training sets often have to be created manually by taking a set of inputs and annotating the expected outputs based on human judgment. This process is expensive and time consuming because it requires one or more human annotators to screen all of the examples and make a decision for each case. It is therefore desirable to find methods that can automatically perform at least parts of this process. This problem can be understood as a problem of information retrieval since the reference information must be (implicitly) contained in the data if human annotators are able to reproduce it.

Contribution: In this work we present an approach to automatically generating annotations for exophoric referring expressions in a situated task-based dialogue. We focus on identifying the referent of a referring expression, as this is a task that (unlike the determination of the set of visible objects for example), cannot be performed automatically in a straightforward manner and generally requires the attention of a human annotator. We predict the referent of a referring expression based on the interpretation of that expression in the dialogue. This is only possible if the referring expression can be related to some detectable action. We therefore only consider referring expressions in utterances that instruct the hearer to perform some specific task. In the experiments described in this work we focus on one specific kind of instruction, namely instructions to pass through a door.

Overview: In Section 2 we discuss corpora that are possible fields of application for our approach and introduce the corpus that is used in the example presented in this work. In Section 3 we present our approach to detecting the referents of referring expressions. In Section 4 we present the evaluation of the application of our approach to test data. Finally, in Section 5 we discuss these results and possible extensions of this work.

2 Data

For this experiment we were interested in corpora featuring situated dialogue. In addition to information immediately related to the dialogue, such as transcriptions and annotations, we were also interested in additional data related to the environment, such as maps and recordings of the actions of the participants.

There exist a number of freely available situated dialogue corpora. The TRAINS corpus [7], which contains dialogues between two participants planning train routes on a map, is an example of a corpus that incorporates the visual modality, and has transcriptions, but does not feature reference annotations. In addition to that, the corpus works with a static map, which is not dynamically updated, which makes it difficult to annotate referring expressions, because participants frequently talk about hypothetical scenarios. In addition to this, it also lacks a record of the planned routes.

Another visually situated corpus is the MAPTASK corpus [2]. This corpus is based on an experiment where one participant describes a route in a map to a second participant, who has access to a slightly different map. Navigation takes place at an abstract level which makes it hard to identify events at a level that would be relevant to this experiment. In addition, both of these corpora are based on tasks that feature a setup where the participants are looking onto a scene that is displayed on a flat surface on a desk. As such, the participants are not inside the scene, but have an external perspective on it.

The corpus considered in this work is the SCARE corpus [4]. This corpus consists of dialogues between two participants in a navigation task where the environment is perceived from a first person perspective. It contains transcriptions and reference annotations and is therefore a good example for learning referring expression resolution. Moreover, unlike the TRAINS and MAPTASK corpus, the SCARE corpus features recordings of all navigation steps, thereby enabling us to reconstruct actions performed by the player.

What differentiates the SCARE corpus and makes it particularly interesting to us, is that it does not take a remote approach with an external perspective, but is very situated, by putting the participant inside the environment. The corpus was created in an experiment focusing on situated task-based dialogues. In this experiment one participant, the direction follower (DF), had to navigate through an environment simulated in a game engine, while the second participant, the direction giver (DG), had to give directions to the first participant to help them to fulfil a given task. The details of the task and the layout of the world were known only the the DG. The DF navigated through the environment in a first

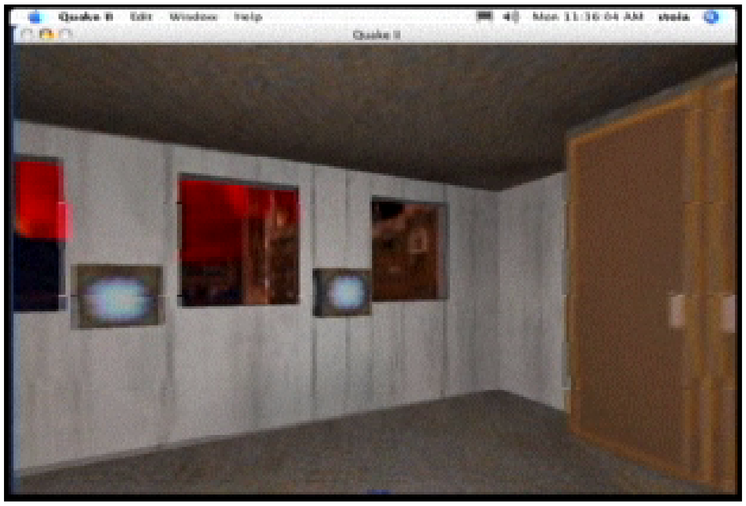

Fig. 1. Screenshot of a video recording from the SCARE corpus.

person perspective, of which a live video feed was shown to the DG. Therefore both participants had the same perspective on the environment. The participants communicated through a voice connection.

The corpus comprised video and audio recordings of the dialogues, as well as transcriptions of the audio files that were annotated for reference, i.e. referring expressions that referred to objects in the environment were annotated which object the expression referred to. In addition to that, demo files were provided that could be replayed in the game engine, thereby recreating the navigation movements in each dialogue.

3 Extending the SCARE Corpus

As noted in Section 2, the SCARE corpus contains annotated dialogue transcriptions and a record of the player movement. In order to automatically annotate referring expressions, we needed to create new data from the corpus. In particular, we had to identify a set of referring expressions and then determine the referent for each expression. We did this by establishing a correspondence between instructions that contain a referring expression in the dialogue and events in the world that could be caused by these instructions. The events we wanted to consider were not explicitly contained in the data, so we had to reconstruct them. Consequently, establishing a correspondence between instructions in the dialogue and events in the world involved 3 steps:

1. We detected a set of instructions.
2. We detected a set of events.
3. We established a correspondence between instructions and events and recorded values for different distance metrics between instructions and events.

Each of these steps is described in detail below. We then evaluated the correspondence against gold standard manual annotations. This evaluation is described in Section 4.

3.1 Detecting the Instructions

In this experiment we were interested in referring expression that caused events we could detect by looking at the movement of the player in the environment. One class of such events is passing through doors. We therefore detected instances of the DG telling the DF to go through a door. We did this using a regular expression of this form[1]:

```
[go|pass]through.*[door|one|that]
```

This expression fit instructions such as "go through the right door" or "pass through the next one". We collected instructions up to a length of seven words. The regular expression was defined by examining a small number of the dialogues in the SCARE corpus. In total we detected 135 referring expressions using this regular expression. This approach probably did not capture all instructions, but served as a good starting point.

3.2 Detecting Events

Once we had detected the set of instructions that we would use in our experiment we then had to detect the set of relevant events to match against the instructions. We did this by replaying the demo files in the game engine and recorded the position and orientation of the player and aligned this information with time. By comparing this information with geometric information about the layout of the rooms, we were able to detect the moments when the player left a room and entered another room. This in turn enabled us to determine which door the player had passed at what point in time. Each passing of a door formed an event.

3.3 Establishing the Correspondence

In this step we determined a correspondence between instructions and events for our example corpus. We aimed to identify for each instruction the event that occurred when the DF fulfilled the instruction. Events naturally occur slightly after the instruction has been produced because the DF needs time to interpret the instruction and to navigate into a position where it is possible to perform the required action. However, not every instruction is immediately succeeded by an event that fulfils the instruction. We see two main reasons for this:

1. The DF may misunderstand the instruction and perform a different action.

[1] $.*$ matches any sequence of characters, $[x|y]$ matches the sequence x or y.

2. The DF may not understand an instruction or find it ambiguous and ask the DG to clarify. In this case, the next event may follow after a longer delay, during which the participants come to an agreement about the next action, and may actually not fulfil the original instruction because the participants decided on a different course of action.

Also, a number of other events may occur between an instruction and the corresponding event because the DF has to fulfil a number of subgoals in order to be able to fulfil the instruction.

At first glance, two approaches are apparent: we can either start with the events and search for an instruction to match each event; or we can start out with the instructions, and determine which event was caused by each instruction. The first approach immediately appeared less favourable because in the example dialogues, a great number of events are not directly caused by instructions. This happens when the DF is exploring the map on their own, or if the DG gives high level goals, such as returning to a previously visited room, which the DF can fulfil without being instructed in every step. We therefore decided to use the approach where we start with the instructions and then search for events that match these instructions.

We processed each dialogue incrementally by going through it from the beginning, picking up instructions and events as they occurred. An incoming instruction was processed by storing it on a FIFO queue. An incoming event was processed by removing the oldest instruction from the queue and associating it with the new event, and storing the resulting pair for later evaluation. This was based on the assumption that events were preceded by instructions. Roughly speaking this approach associates each instruction with the next event occurring after it.

We collected instructions in a queue because that enabled us to correctly interpret concatenated instructions (e.g. "go through this door and then go through the next one") as two instructions that had to be executed sequentially.

Events that occurred while the instruction queue was empty were discarded as events that occurred without explicit instruction. Such events occurred often in the dialogues when the DF was asked to move to a location that had previously been visited. In this case, the DF often could find the way on their own without having to be explicitly instructed for each step.

The matching process resulted in a set of pairs of instructions and events which was the basis for our further work. The algorithm we used for this process is presented as Algorithm 1.

Every time an instruction and an event were associated, we recorded the distance in time between instruction and event, and the number of words spoken between them to facilitate evaluation. We derived these values from the time aligned dialogue transcriptions.

The output of the algorithm consists of a list of associated instructions and events. Each pair represents a possible causal relationship between an instruction and an event, which will be examined more closely in the next step.

Algorithm 1 The algorithm for associating instructions and events.

Input:	**dialogue:** temporally ordered set of events and instructions.
Data Structures:	**instructionQueue:** A queue for storing incoming instructions, initially empty.
	correspondences: A list of instructions-event pairs.
Output:	A list of instructions-event pairs which may be related.

```
FOR the length of the dialogue
   e := select the next event or instruction
   IF e is an instruction
      push(e, instructionQueue)
   ELSE
      IF e is an event
         IF empty(instructionQueue)
            discard e
         ELSE
            i := pop(instructionQueue)
            a := associate(i,e)
            append(correspondences,a)
         ENDIF
      ENDIF
   ENDIF
RETURN correspondences
```

Figure 2 illustrates the approach. Intervals of speech are represented as blocks below the time axis. Dark blocks represent instructions, while bright block represent speech that is not an instruction. Stars on the time axis represent events. The horizontal brackets delineate the intervals between the end of an instruction and the next event. The dashed vertical lines cut out intervals on the the time axis and pieces of the speech blocks, which form the distance values.

4 Evaluation

As mentioned in Section 2, referring expressions in the original corpus were annotated for reference. We therefore knew for each referring expression to which object it actually referred. This information formed the gold standard for the evaluation of our approach to reference resolution.

Once we had processed all the dialogues in the corpus we used the unmodified set of instruction/event pairs we had identified as the baseline for our experiment. Using this set of data as a prediction function for causal relation corresponds to assuming that the referring expressions in each instruction referred to the object that was concerned in the next event that occurred. Conversely, this would assume that each instruction was perfectly interpreted and fulfilled directly after the instruction.

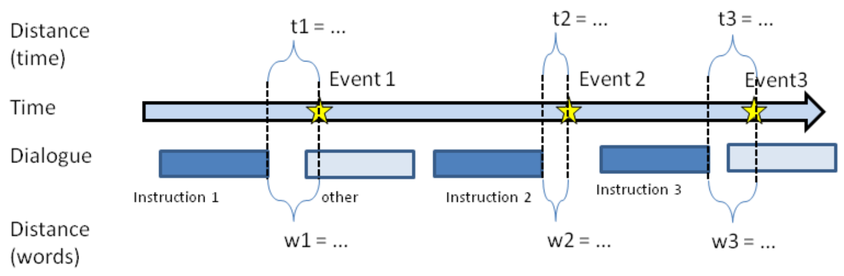

Fig. 2. Illustration of the instruction-event association and distance measuring process. Blocks represent intervals of speech, stars represent events.

This appeared a very strong assumption, since misunderstandings between human communicators frequently occur. We therefore utilized different ways of measuring the distance between an instruction and an event and used this distance measure to judge the reliability of the original pairing. We measured the distance in time (i.e. the number of seconds that passed between instruction and event), and words (i.e. the number of words that were spoken between the instruction and the event). We experimented with a number of different distance values as cut-off points. If the distance between an instruction and the following event was too large, we would refuse to rate it, leaving the decision up to a human annotator.

We ran the association algorithm (Algorithm 1) to create a set of instruction-event pairs. We subsequently judged the results by a number of different distances. The time distances we used were 5, 7.5, 10, 15 and 20 seconds, the word distances were 5, 10, 20, 40, 50 and 60 words. The results for the time distances are presented in Table 4, the results for word distances in Table 4. They show:

- the total number of cases (Column 1)
- the number of cases that were removed because the distance between instruction and event exceeded the boundary of this instance (Column 2). This row is illustrated in Figure 3 for time distances and in Figure 4 for word distances.
- the percentage of removed cases (Column 3)
- the number of remaining cases (Column 4)
- the number of cases where the association between instruction and event was correct according to the baseline (Column 5). This row is illustrated in Figure 5 for time distances and Figure 6 for word distances.
- the percentage of correct cases among the cases that were not removed (Column 6)
- the overall percentage of the correctly associated cases among the number of total cases (Column 7)

To give an intuition about the significance of the different columns: Column (3) tells us for what fraction of the cases the algorithm refused to make a judgment. The figure basically tells us what how much work is left for the human

	Total	Removed		Remaining	Correct		Total correct
	#	#	%	#	#	%	%
Col. Nr.	(1)	(2)	(3)	(4)	(5)	(6)	(7)
Baseline	135	0	0.0	135	93	68.9	68.9
Time 5s	135	88	65.2	47	34	72.3	25.2
Time 7.5s	135	61	45.2	74	61	82.4.	45.2
Time 10s	135	40	29.6	95	80	84.2	59.3
Time 15s	135	34	25.2	101	84	83.2	62.2
Time 20s	135	26	19.3	109	88	80.7	65.2

Table 1. Results for the different time distance values.

	Total	Removed		Remaining	Correct		Total correct
	#	#	%	#	#	%	%
Col. Nr.	(1)	(2)	(3)	(4)	(5)	(6)	(7)
Baseline	135	0	0.0	135	93	68.9	68.9
Words 5	135	122	90.4	13	4	30.8	2.9
Words 10	135	102	75.6	33	21	63.6	15.5
Words 20	135	62	45.9	73	59	80.8	43.7
Words 40	135	38	28.1	97	80	82.5	59.3
Words 50	135	31	23.0	104	85	81.7	63.0
Words 60	135	26	19.3	109	86	78.9	63.7

Table 2. Results for the different word distance values.

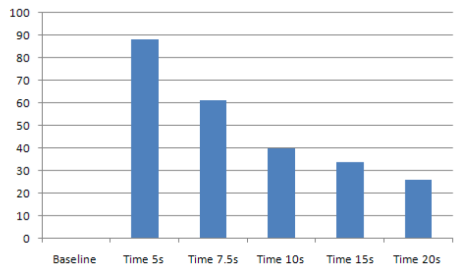

Fig. 3. Proportion of removed cases by time distance.

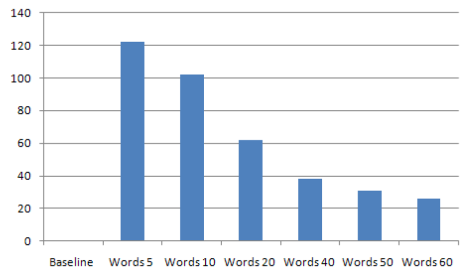

Fig. 4. Proportion of removed cases by word distance.

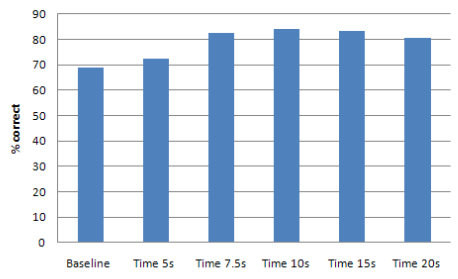

Fig. 5. Proportion of correct judgements by time distance.

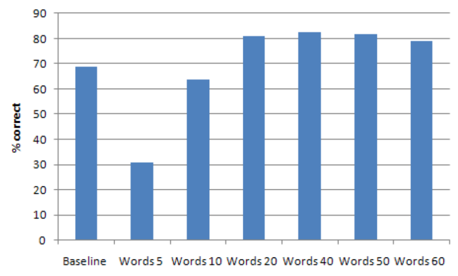

Fig. 6. Proportion of correct judgements by word distance.

annotator. Column (6) tells us how many of the cases that were not refused were judged correctly. This basically gives us a measure of the quality of the predictions made.

Column (7) tells us which fraction of the total number of cases was correctly annotated according to the manual annotations from the corpus.

As we can see, the baseline alone delivers somewhat acceptable results. However, if we were to use the baseline approach in an actual annotation task, we would end up with false results with no indication of which results were doubtful decisions.

Overall the figures show that taking the distance between instruction and event does increase the quality of the judgments. For the approach using time as a cut-off criterion, it appears that early cut-off points removes many cases while later cut-off points retain more. The quality of judgments peaks around 10 seconds. Nonetheless, it might be more economical to choose a later cut-off point to get more cases processed, while only incurring a small decrease in quality.

The picture using words as a cut-off criterion appears similar. A peak occurs around 40 words. In this respect, both approaches show similar results, and it does not appear that one of them is clearly preferable.

5 Conclusions and Future Work

We presented an approach towards automatically generating referring expressions annotations for situated dialogues that exploits the interpretation of referring expressions by the participants of the dialogue. We demonstrated the approach for a specific type of references in a specific corpus. The approach can be generalized to other types of references in other corpora under two conditions: (1) The references must be contained in instructions that cause events involving the referents and (2) It must be possible to automatically detect these events.

On a conceptual level we can relate this approach to more general approaches that are based on intention recognition and perceived affordances [9]. In [8] Gorniak et. al. describe using intention recognition to improve reference resolution in the context of a game. In this work we somewhat reverse this approach: We take actions in the game as hypotheses about the intention of instructions (quasi hijacking the interpretation performed by the listener) and use the objects affected by the action as the referent of referring expressions in the instruction.

We explored different cut-off values that give an indication for which suggested linkings might be unreliable. Deciding on a particular cut-off point, allows the algorithm to decide which cases are easy and reliably judged, and which cases are hard to judge, and should rather be inspected by a human annotator. However, it is not immediately clear how to derive cut-off values for new domains. It may be possible to directly transfer values between sufficiently similar domains. Another approach would be to manually create a gold standard annotation for a small subset of the domain and to determine values for this subset and transfer them to the whole domain.

A different approach to judging reliability of associations would be to, instead of rejecting associations, to integrate the data about the reliability of the different cut-off points into a single confidence estimation function that directly signals the reliability of a judgment based on the different distance measures.

The evaluation shows that taking cut-off points into account makes it possible to automatically correctly annotate a nontrivial subset of the referring expressions that were considered, while separating out cases where a human annotator should be consulted. For future work it would be interesting to more closely analyze the structure of the discourse between instruction and event. It might be possible to discover confusion or discussion between the participants, and use this as an additional distance measure. Possible indicators might be the word rate or overlap between the utterances of the participants. Auditory features such as the prosody of the utterances might also be significant indicators.

The GIVE corpus [5] comprises a data set, that is very similar to the one we used, but is based on written instead of spoken language and features only monologue. In further work we may investigate how well our approach can applied to this corpus and in how far results are transferable.

References

1. Jurafsky, D., and Martin, J. H. (2008).: Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition. Prentice Hall, second edition.
2. Kievit, L.; Piwek, P.; Beun, R.-J.; and Bunt, H. (2001). Multimodal Cooperative Resolution of Referential Expressions in the DenK System. In: Cooperative Multimodal Communication, p. 197-214. Springer Berlin / Heidelberg
3. Salmon-Alt, S., and Romary, L. (2000). Reference Resolution Within the Framework of Cognitive Grammar. In: Proceedings of the International Colloqium on Cognitive Science, San Sebastian : Spain (2001). Volume: abs/0909.2626.
4. Stoia, L.; Shockley, D. M.; Byron, D. K.; and Fosler-Lussier, E. (2008). Scare: A Situated Corpus with Annotated Referring Expressions. In: Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008).
5. Gargett, A.; Garoufi, K.; Koller, A.; and Striegnitz, K. (2010). The Give-2 Corpus of Giving Instructions in Virtual Environments. In In Proceedings of the 7th Conference on International Language Resources and Evaluation (LREC), Valletta, Malta, 2010: European Language Resources Association (ELRA).
6. Grosz, B. J.; Weinstein, S.; and Joshi, A. K. (1995). Centering: A Framework for Modeling the Local Coherence of Discourse. Computational Linguistics 21(2):203225.
7. Heeman, P., and Allen, J. (1995). Trains 93 dialogues. Technical Report. University of Rochester, Rochester, NY, USA.
8. Gorniak, P.; Orkin, J.; and Roy, D. (2006) Speech, Space and Purpose: Situated Language Understanding in Computer Games. Twenty-eighth Annual Meeting of the Cognitive Science Society Workshop on Computer Games
9. Gorniak, P. (2005). The Affordance-Based Concept. PhD thesis, Massachusetts Institute of Technology.