Markov Logic Networks for Spatial Language in Reference Resolution

Casey Kennington

Dialogue Systems Group, CITEC
Faculty of Linguistics and Literary Studies
Universität Bielefeld, Bielefeld, Germany
ckennington@cit-ec.uni-bielefeld.de
www.caseyreddkennington.com

Abstract. This paper presents an approach for automatically learning reference resolution, which involves using natural language expressions, including spatial language descriptions, to refer to an object in a context. This is useful in conversational systems that need to understand the context of an utterance, like multi-modal or embodied dialogue systems. *Markov Logic Networks* are explored as a way of jointly inferring a reference object from an utterance with some simple utterance structure, and properties from the real world context. An introduction to MLNs, with a small example, is given. Reference resolution and the role of spatial language are introduced. Different aspects of combining an utterance with properties of a context are explored. It is concluded that MLNs are promising in resolving a contextual reference object.

Keywords: reference resolution, spatial language, markov logic networks

1 Introduction

When we speak in a dialogue setting, we are often co-located (situated) in space and refer to objects in that space. Besides using salient properties to refer to an object, like its color, shape, and size, the language which is often used to refer to those objects make up *spatial language*; that is, where the object is relative to some frame of reference, or as relative to other objects in that space. Understanding spatial language has application in fields like dialogue systems, multi-modal systems, and robotics, and makes up a part of natural language understanding (NLU) in situated environments.

Work done in reference resolution has focused mainly on word-level approaches (see [16]), without specific focus on spatial language. [20] construct a visual grammar for reference resolution with some degree of success. [18] looks at incremental reference resolution, but also only looks at words. In this paper, we extend the work of reference resolution by focusing on real world properties and bridge them with utterances that use spatial language, with a small amount of linguistic structure. We created a statistical model trained on generated data and apply Markov Logic Networks (MLNs, [14]) as the machine learning technique in the experiments. It is shown that these models are significantly better than the baseline and that spatial language can be learned and applied to reference resolution using MLNs.

R.K. Rendsvig and S. Katrenko (Eds.): *ESSLLI 2012 Student Session Proceedings*, CEUR Workshop Proceedings vol.: see http://ceur-ws.org/, 2012, pp. 54–64.

Plan of this paper: In the following section, MLNs are defined and a simple example is given. The last part of the section will define reference resolution, with particular attention to spatial language and how it pertains to the task of this paper. In section 2, the domain, task, data, and procedure are explained. The experiment section then shows how well the system performs in reference resolution, as well as some visual representations of how well the system learned about spatial language.

1.1 Markov Logic Networks

Markov Logic Networks have recently received attention in language processing fields like coreference resolution [3], semantic role labeling [13], and web information extraction [17]. As defined by [14], a Markov Logic Network is a first-order knowledge base with a weight attached to each formula (see also Markov Random Fields [10], which make up MLNs). This knowledge base is a declaration of predicates and typed arguments, along with weighted first-order logic (FOL) formulas which define the type of relations between those predicates. This becomes a *template* to a MLN network. *Evidence* in the form of instances of the predicates can also be given either for training the network, or for direct inference. MLNs are a type of graphical model, where the nodes are not directed, an example of which can be found in Figure 2.

As an example MLN, consider the following statement: *If you write something down, you are twice as likely to remember it.* We can represent this by the predicates $Write(person)$ and $Remember(person)$. This can be formulated as: 2 $Write(x) \Rightarrow Remember(x)$. This defines a relationship between *Write* and *Remember*. The argument in both *Write* and *Remember* have the type *person* and for this example, we will introduce only one constant: *Mary*. Building all possible formulas given this information requires all combinations of positive and negative predicates in the formula, resulting in four formulas, all with weight 2, which will be referred to as *worlds*. It is then necessary to determine which of those four formulas are satisfied by the original formula of $Write(x) \Rightarrow Remember(x)$. It is satisfied when $Write$ is negative or when *Remember* is positive, or both. The only time it is not satisfied is when $Write(Mary) \Rightarrow \neg Positive(Mary)$.

After a set of worlds is created, one can perform inference by calculating a probability:

$$P(X = x) = \frac{1}{Z}\, e^{\sum_j w_j f_j(x)} \tag{1}$$

Where Z, also known as the partition function, is the normalizing constant given by:

$$Z = \sum_{x\ in\ X} e^{\sum_j w_j f_j(x)} \tag{2}$$

Where j indexes over the formulas. Here w_j is defined as the weight of the corresponding formula (in our example, the weight was 2), and f_j is a function that returns 1 if the formula is satisfiable, and 0 if it is not satisfiable. After the worlds and their satisfiabilities are identified, the next step is to determine Z. If our evidence is $Write(Mary)$, Z can be computed by finding which of the four formulas

satisfy the evidence $Write(Mary)$, which results in one world that is satisfiable by the original formula, and one that is not. Given this set of worlds, the numerator of the probability is found by identifying the formulas which satisfy a query, for example, $Remember(Mary)$?, which results in only one satisfiable formula. In this case, the probability of $Remember(Mary)$ given the evidence $Write(Mary)$ is $\frac{e^2}{e^2+e^0} = 0.88$.

This simple example only considered a very small set of possible worlds. It doesn't take many more formulas, predicates, arguments, or constants to make computing MLNs intractable for large domains. In fact, inference alone is NP-Hard [15], and determining clause satisfiability is NP-Complete [21]. Most of the current research done in MLN attempts to find better ways to approach these problems of intractability, with a large degree of success. Furthermore, only inference was discussed. Inference is performed when the weights are given. There are also mechanisms for learning the weights given training data, but the details of how that is done will not be discussed here. For this paper, I use the discriminative learning approach [21] to automatically learn the formula weights. Methods for training MLNs can also be found in [5] and [12]. A book by Pedro Domingos and Daniel Lowd [6] offers a good introduction to MLN inference, weight learning, and contains several examples.

1.2 Reference Resolution

Reference resolution involves the linking of natural language expressions to contextually given entities [18]. Visual properties are often used to this end, properties such as color, size, and shape. However, we often use spatial relationships with other objects to aid in that reference. As spatial language plays a very important role in the reference resolution in this paper, spatial language will now be discussed.

Spatial Language

Learning the meaning of spatial relationships has been done in navigational direction tasks using reinforcement learning [22]. Spatial language has also been studied in psycholinguistic research, as well as practical applications like robotics (see *Spatial Language and Dialogue* [4] and *Language and Spatial Cognition* [8] by Annette Herskovit). When humans use spatial language, they use properties of the objects to which they are referring, such as color, shape, relative position to another salient object, or relative position with respect to some axis [23].

Spatial language involves the language humans use to describe space and the objects in that space. Humans require a common understanding of *absolute* and *relative* spatial descriptions in order to communicate effectively. Words such as *left*, *right*, *top*, *bottom*, and *middle* can represent absolute descriptions. Relative means that objects identified relative to another, perhaps more visually salient object. This is where prepositions are often used; *on top of*, *below*, *next to*, *to the left of*, *beside*. Even if we can gesture by pointing or looking at an object, we still usually need to be able to articulate the linguistic description of the object such that the hearer of the utterance can uniquely identify the object to which we are referring. This linguistic description becomes even more important when a human is interfaced with a computer and language is the only medium for communication from human to computer, which is the experimental setting for this paper.

It is essential to establish a frame of reference when using spatial language. It has been shown that alignment of the reference point is an ongoing process across utterances, depending on the context [24]. [11] and [22] use two main categories for

the frame of reference: *egocentric* where the speaker is the frame of reference, and *allocentric* were the coordinates are not based on the speaker. In this paper, we will assume a single frame of allocentric reference. Specifically, both the hearer and the speaker use the same frame of reference.

2 Two-Dimensional Spatial Learning

In this section, the domain, task, data, and procedure of the experiment which will use MLN to automatically learn reference resolution, particularly using spatial language, are defined. By giving accuracies of successfully referred objects, we show that MLNs can be used successfully in reference resolution, as well as some figures that show the extent that spatial language was learned.

2.1 Domain, Task, Data, and Procedure

Domain

In this study, we used the *Pentomino* puzzle piece domain. A Pentomino board is visually represented in rows and columns with pieces viewable by a human. On the computer side, pieces are identified with unique arbitrary identifiers. The properties of the pieces which are visually distinguishable by a human (color, shape, row, column, relation to other pieces) are accessible also to the computer. In this way, the board and pieces effectively become a shared visual context between the human and the computer, as described in [20]. An example Pentomino board can be seen in Figure 1.

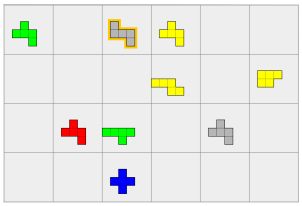

Fig. 1. Pentomino Board Example

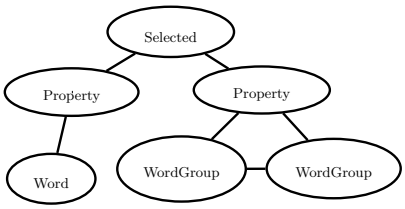

Fig. 2. MLN Relations

Perspective is not being studied in this paper, so assumptions need to be made about the frame of reference. The Pentomino board defines a fixed frame of reference which is a board on a computer screen. Also, given the nature of the utterances used in this paper (both generated from a corpus), egocentric descriptions such as *my left* were not used. The utterances used involve three components to each spatial description: *target*, a possible *reference object*, and *spatial term(s)* [2]. The goal is to see if these can be learned automatically to identify the reference object.

Task

The Pentomino board as implemented in the InPro toolkit [1, 19] was used.[1] The larger goal of NLU in Pentomino is to understand what action which is to be performed on the board (rotate, delete, move, mirror, or select a piece), which results in a change of the board's state (e.g., rotating a piece means visually making it appear with a different orientation). This experiment will focus only on identifying the piece to which the user is referring, not the action or change of board state.

Data

The training and evaluation data were automatically generated. This was a natural choice because one of the goals of this paper is to determine how well MLNs perform on language data. We can infer that if MLNs can't be made to work well on simplistic, automatically generated data with simple relational structures, then they will not work on real speech data with more complex structures. Further, the use of automatically generated data was motivated by the fact that the perspective is fixed, making the utterances of a finite type. Also, the task is reference resolution so the verb can be assumed to be a command-form *select* (as in "Select the piece ..."), thus reducing the possible syntactic structures. This puts focus on the spatial language and words that describe other properties of a piece, which can be generated to be representative of typical real-world language use. Finally, we don't generate actual utterances, but a simple semantic representation that defines relationships between words and word groups (phrases), though we will refer to them as the utterance. This kind of semantic representation can be obtained from most syntactic or semantic formalisms, though no syntactic or semantic parsing is done here.

The training data were generated by creating one thousand boards. Each board was randomly assigned 3-5 rows and 3-5 columns (number of rows and columns need not be the same), creating a small variability in the dimensions of the boards which allowed for generality. The number of pieces was also randomly assigned, between 4 and 6 pieces for training where the maximum number of pieces must be no larger than the number fields in the smallest possible board. Each piece was assigned a random color, shape, row, and column, and of course multiple pieces could not occupy the same space. After each piece was in place, one was chosen randomly to be the *selected* piece, the piece that was referred. A pseudo-utterance that described that piece was then generated in the form of a simple semantic representation. The pseudo-utterance contained words that described the shape, color, absolute, and relative descriptions. There were a total of 5 colors and 7 shapes. I used the following absolute spatial descriptions: *left*, *right*, *top*, *bottom*, *middle*, *corner*, and the following relative spatial descriptions: *above*, *higher than*, *on top of*, *next to*, *beside*, *below*, *under*, *beneath*, *on the left of*, and *on the right of*. The shape, color, and all absolute spatial descriptions were treated as a bag of words. The words in a relative spatial description were treated as a word group, which corresponds to a small amount of linguistic structure (grouped, for example, as prepositional phrases).

An example of this is given in Figure 3. Here, the selected piece is a red cross somewhere near the middle, below a gray cross which is in the middle of the top row. The generated pseudo-utterance would be something like *select the middle red cross below the gray cross in the top middle*, where *red*, *middle*, and *cross* are treated as simple words, and *below the gray cross in the top middle* is treated as a word group.

[1] `http://sourceforge.net/projects/inprotk/`

The evaluation set was similarly generated, but used 100 boards, with row and column lengths between 3 and 7, and the number of pieces was between 4 and 9. The range of rows and columns, and the range of possible number of pieces was larger than in training. For each evaluation experiment, the evaluation set was randomly generated, so each one was different.

Part of the effort in using MLNs is determining what should be specified as predicates (e.g. Color, Shape, Row, etc) and what should be learned automatically. In this paper, only one abstract predicate, *Property*, that implicitly does the typing into different features was used. The example in Figure 3 shows a single *Piece*, and multiple *Property* predicates for that piece. Where numbers would cause confusion, they can simply be annotated with a unique property type (e.g. column 1 = 1C). The properties used as represented in Figure 3 were type, color, % horizontal from center, % vertical from center, row, and column. The predicates *Word* and *WordGroup* represent the words, and group of words, respectively, as previously explained. The final argument for all the predicates is the board identifier, which separated states of the board and corresponding selected pieces and utterances from other boards. The second argument in *WordGroup*, groupID, is a unique identifier that is the same across a group of words. In Figure 3, the identifier is simply 1, but if more *WordGroup*s existed, then they would be represented by a different number. A typical representation of a board would have many pieces and more word groups for the utterance.

Figure 3 also shows the actual MLN template that was used for training. Lines 1-5 are the predicate definitions and argument types and 6-7 define the relations via FOL formulas. The + before an argument in a formula tells the MLN to learn from each constant in the training data, which is necessary when dealing with natural language because it must learn how to interpret individual words as individual symbolic features. The result is that words and word groups are mapped to properties.

MLN template

1 $Piece(piece, boardID)$
2 $Property(piece, prop, boardID)$
3 $Word(word, boardID)$
4 $WordGroup(gWord, groupID, boardID)$
5 $Selected(piece, boardID)$
6 $Word(+w, b) \wedge$
$Property(p, +p1, b) \Rightarrow$
$Selected(p, b)$
7 $WordGroup(+w1, e, b) \wedge$
$WordGroup(+w2, e, b) \wedge$
$Property(p, +p1, b) \Rightarrow Selected(p, b)$

example

```
Piece(tile-7,-1)
Property(tile-7,X,-1)
Property(tile-7,red,-1)
Property(tile-7,0H,-1)
Property(tile-7,-67V,-1)
Property(tile-7,2R,-1)
Property(tile-7,1C,-1)
Word(red,-1)
Word(cross,-1)
Word(center,-1)
WordGroup(below,1,-1)
WordGroup(gray,1,-1)
WordGroup(cross,1,-1)
WordGroup(top,1,-1)
WordGroup(middle,1,-1)
```

Fig. 3. MLN template and example for *select the middle red cross below the gray cross in the top middle*

Procedure

The *Alchemy* system [5] for MLN learning and inference was used for these experiments.[2] The board and pseudo-utterances were represented as evidence to the MLN, as well as which piece was selected. Discriminative training with `Select` as the predicate to query was used. As MLN is a way of defining a relation between various predicates, those predicates need to represent meaningful information from the Pentomino board, as well as the utterance and how the utterance relates to the board, as shown in Figure 3.

To test, a board is similarly randomly generated as in training, and a piece is again randomly selected. However, the selected piece is what was being inferred about by the MLN, so it was kept hidden. The pseudo-utterance was generated in the same way as in training. Using the state of the board and the utterance, the MLN system then inferred which piece was being described by that utterance. If the piece with the highest probability matched the selected piece, it was marked as correct. If there was a tie, then the first one returned by the query was chosen. This is therefore a simple measure of accuracy. In this scenario, the majority class baseline would be 25%, where 4 is the minimum number of possible pieces during evaluation.

2.2 Experiment: Absolute and Relative Spatial Understanding

Table 1 shows various training and evaluation settings that were used. Each row represents the parts of the utterance and board properties that were represented. The **Full** column was trained on one set of boards with all possible descriptions (shape, color, relative spatial, absolute spatial, row, column) and that trained model was used to evaluate for both rows in the column. Even though, words like *row* and *column* are often used to distinguish pieces in this kind of setting, those words are left out of evaluation unless specifically stated because they are easy features to learn as they map easily to their corresponding piece properties. The **Ind** column (short for Individual) in the table shows the results when each row's setting was trained individually, and subsequently evaluated.

Description	**Full**%	**Ind**%
shape, color	70	92
absolute spatial with rows and columns	87	97
absolute spatial	65	69
relative spatial	35	59
shape, color, absolute spatial	**92**	**100**
absolute and relative spatial	34	82
all	58	88

Table 1. Various `Select` Accuracy Results; **Full** represents a single trained model with all features and evaluated only on the features of the row, **Ind** represents a different model trained and evaluated on the features of the corresponding row.

[2] `http://alchemy.cs.washington.edu/`

Though the main goal of this paper is to show that a MLN can effectively learn spatial language reference resolution, it is also useful to see how a MLN performs in different settings, as shown in Table 1. Overall, alchemy and MLN perform above baseline in all areas. Most interesting to note is the fact that when all types were used (shape, color, absolute, and relative spatial), the accuracy was only 58%. This is possibly due to the fact that it was trained on a system which included the rows and columns (information that was not used at evaluation). This is evidenced by the fact that the *Ind* column for the same row gave 88% and was trained on similar settings, only minus rows and columns, forcing the model to learn how to resolve without rows and columns as piece properties. It is further interesting to note that shape, color, and absolute spatial received high accuracies for both systems. However, a system with relative spatial knowledge will be more robust to real language input in a real-world environment, despite the somewhat lower probability.

2.3 Board Distributions

Another way of showing how well the MLN learned about spatial language is to look at a graphical representation of the probabilities of each piece in the board as a gradient, where the darker in color, the higher the probability. Some of these gradient boards (using a 5x5 board) are displayed, where all fields are taken by a piece with the same color and shape, thus nullifying them as distinguishing features. First, a look at absolute spatial language. An example of absolute *top-left*, and absolute *centre-right* are represented in Figures 4 and 5 respectively. These show that the absolute language was well learned, especially for a specific point like *top-left* in Figure 5.

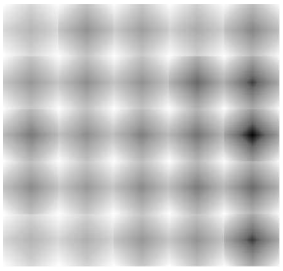

Fig. 4. Center Right

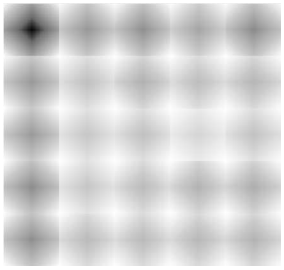

Fig. 5. Top Left

Relative spatial language is a little more difficult to visualize in a gradient, but certain relative relations can be isolated by looking at a gradient map with its corresponding board, where the board is not completely filled. The board and gradient map for the relation *above* can be found in Figures 6 and 7. For a piece to be *above* another piece, it simply needed to be in a higher row than that piece, regardless of the column. The notion of *above* is learned, but the distribution over the pieces is somewhat unexpected. Any piece that is above another piece should have some probability, with the darker gradients starting at the higher pieces, decreasing with the rows. The concept of *above* here is generally learned, though there is need for improving the model when it comes to relative spatial relationships.

After the submission of this paper, we used the principles that were learned here and applied them to real, non-generated Pentomino data collected in a Wizard-of-Oz study [7, 18] in a situated setting of natural language understanding. That work resulted in a paper [9] in which we used MLNS not only for reference resolution, but to predict a semantic frame which also included the action to take on the referred piece, and the desired state of the board after the action was complete (for example, the utterance *rotate the blue cross clockwise* would have *rotate* as the action, *the blue cross* as the referred piece, and the resulting state of the board would be that it appears 90 degrees to the right). Experiments were performed on hand-annotated speech, as well as automatically transcribed speech, evaluated incrementally (word-level increments), and on full utterances. We concluded that information from the visual context (pentomino board), the utterance (which included a syntactic context-free parser and corresponding semantic representation), and previous discourse context perform well, in terms of frame accuracy, in the Pentomino domain.

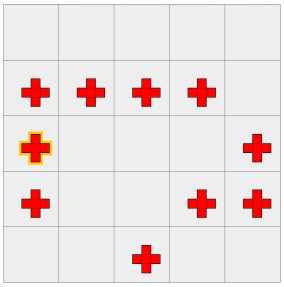

Fig. 6. above Board

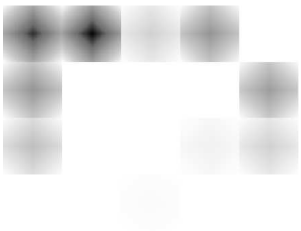

Fig. 7. above Map

3 Conclusions and Future Work

Markov Logic Networks are effective in reference resolution in a two-dimensional domain like Pentomino. They can effectively learn a mapping between an utterance and real world object properties, including utterances that contain spatial language descriptions.

Future work involves using what was learned via this task and implementing it into a larger natural language understanding framework. We will continue to use MLNs, and further incorporate other real-world information, such as eye gaze and gestural information from the human. We will extend this to domains beyond Pentomino, domains which use real world spatial representations, and apply it in interactive settings.

References

1. Baumann, T., Schlangen, D.: The InproTK 2012 Release. In: Proceedings of Human Language Technologies: The 2012 Annual Conference of the North American Chapter of the Association for Computational Linguistics on - NAACL '12 (2012)

2. Carlson, L.A., Hill, P.L.: Formulating Spatial Descriptions across Various Dialogue Contexts. In: Spatial Language and Dialogue, pp. 89–103 (2009)
3. Chen, F.: Coreference Resolution with Markov Logic. Association for the Advancement of Artificial Intelligence (2009)
4. Coventry, K.R., Tenbrink, T., Bateman, J. (eds.): Spatial language and dialogue, vol. 3. Oxford University Press (2009)
5. Domingos, P., Kok, S., Poon, H., Richardson, M.: Unifying logical and statistical AI. American Association of Artificial Intelligence (2006)
6. Domingos, P., Lowd, D.: Markov Logic An Interface Layer for Artificial Intelligence. Morgan & Claypool (2009)
7. Fernández, R., Lucht, T., Schlangen, D.: Referring under restricted interactivity conditions. In: Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue. pp. 136–139 (2007)
8. Herskovits, A.: Language and Spatial Cognition. Cambridge University Press (2009)
9. Kennington, C., Schlangen, D.: Markov Logic Networks for Situated Incremental Natural Language Understanding. In: Proceedings of SIGdial 2012. Association for Computational Linguistics, Seoul, Korea (2012)
10. Kindermann, R., Snell, J.L.: Markov random fields and their applications (1980)
11. Levinson, S.C.: Space in Language and Cognition. Explorations in Cognitive Diversity, vol. 5. Cambridge University Press (2003)
12. Lowd, D., Domingos, P.: Efficient weight learning for Markov logic networks. Knowledge Discovery in Databases: PKDD 2007 pp. 200–211 (2007)
13. Meza-Ruiz, I., Riedel, S.: Jointly identifying predicates, arguments and senses using Markov logic. In: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics on - NAACL '09. p. 155. No. June, Association for Computational Linguistics, Morristown, NJ, USA (2009)
14. Richardson, M., Domingos, P.: Markov logic networks. Machine Learning 62(1-2), 107–136 (2006)
15. Roth, D.: On the hardness of approximate reasoning. Artificial Intelligence 82(1-2), 273–302 (1996)
16. Roy, D.: Grounding words in perception and action: computational insights. Trends in Cognitive Sciences 9(8), 389–396 (Aug 2005)
17. Satpal, S., Bhadra, S., Rajeev, S.S., Prithviraj, R.: Web Information Extraction Using Markov Logic Networks. Learning pp. 1406–1414 (2011)
18. Schlangen, D., Baumann, T., Atterer, M.: Incremental Reference Resolution: The Task, Metrics for Evaluation, and a Bayesian Filtering Model that is Sensitive to Disfluencies. In: Proceedings of SIGdial 2009 the 10th Annual SIGDIAL Meeting on Discourse and Dialogue. pp. 30–37. No. September (2009)
19. Schlangen, D., Skantze, G.: A general, abstract model of incremental dialogue processing. Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics on - EACL '09 (April), 710–718 (2009)
20. Siebert, A., Schlangen, D.: A Simple Method for Resolution of Definite Reference in a Shared Visual Context. In: Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue. pp. 84–87. No. June, Association for Computational Linguistics (2008)
21. Singla, P., Domingos, P.: Discriminative Training of Markov Logic Networks. Computing 20(2), 868–873 (2005)

22. Vogel, A., Jurafsky, D.: Learning to Follow Navigational Directions. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistic. pp. 806–814 (2010)
23. Vorwerg, C.: Consistency in Successive Spatial Utterances. In: Coventry, K.R., Tenbrink, T., Bateman, J.A. (eds.) Spatial Language and Dialogue, chap. 4, pp. 40–55. Oxford University Press (2009)
24. Watson, M.E., Pickering, M.J., Branigan, H.P.: Alignment of Reference Frames in Dialogue. In: Cognitive Science Society (2004)