In Proceedings of Symposium on Machine Learning in Speech and Language Processing (MLSLP 2011), June 2011.

PANNING FOR GOLD: FINDING RELEVANT SEMANTIC CONTENT FOR GROUNDED LANGUAGE LEARNING

David L. Chen and Raymond J. Mooney

The University of Texas at Austin
Department of Computer Science
Austin, TX 78712, USA
dlcc@cs.utexas.edu and mooney@cs.utexas.edu

ABSTRACT

One of the key challenges in grounded language acquisition is resolving the intentions of the expressions. Typically the task involves identifying a subset of records from a list of candidates as the correct meaning of a sentence. While most current work assume complete or partial independence between the records, we examine a scenario in which they are strongly related. By representing the set of potential meanings as a graph, we explicitly encode the relationships between the candidate meanings. We introduce a refinement algorithm that first learns a lexicon which is then used to remove parts of the graphs that are irrelevant. Experiments in a navigation domain shows that the algorithm successfully recovered over three quarters of the correct semantic content.

Index Terms— ambiguously supervised learning, grounded language acquisition

1. INTRODUCTION

The area of *grounded language acquisition* studies how language can be learned by observing its use in some naturally occurring perceptual context. Unlike most work in statistical natural language processing which requires annotating large corpora with detailed syntactic and/or semantic information, this approach tries to learn language without explicit supervision in a manner more analogous to how children acquire language. This approach also grounds the meaning of words and sentences in perceptions and actions instead of arbitrary semantic tokens.

There are many sources of data from which language grounding can be done. Captions accompanying pictures or videos can provide information about names of people [1, 2], events and actions that occur in sports [3, 4], as well as the general content depicted by pictures [5]. Video games and virtual worlds are convenient for grounding names and attributes of objects as well as spatial prepositions since all the physical entities in the world are known and do not have to be identified [6, 7, 8, 9]. Descriptions of events such as sportscasts [10], football game summaries [11], and weather forecasts [12] can be grounded to event logs, database of game statistics, and weather data respectively. Finally, instructions can be mapped to concrete actions such as performing computer-related tasks [13, 14] or navigating to a destination [15, 16, 17, 18].

Central to the language grounding task is the problem of identifying relevant parts of the context that the language is referring to. This is an example of a broader class of learning problems in which the supervision is ambiguous. Unlike traditional supervised learning where the correct labels are provided, ambiguous supervision provides a superset of labels that contain the correct label. In some cases, the correct label might not even be in the provided set (although this is usually assumed to be infrequent, otherwise learning would be impossible.) The learner must then solve the additional task of predicting the correct labels in the training data.

One of the earlier work on solving the ambiguity problem was done by Siskind [19]. He presented an algorithm for learning meanings of words in cross-situational contexts but did not solve the ambiguity problem at the sentence level. Several recent projects have looked at aligning the sentences directly to the relevant semantic contents without explicitly learning the meaning of words. Snyder and Barzilay [11] used supervised data to train a multi-label classifier that aligns novel sentences to their corresponding database entries. Another approach is to iteratively train semantic parsers on the ambiguous training data and use those parsers to predict which are the correct meanings in the candidate sets [20, 10]. There has also been work that use generative methods to model the process of selecting relevant semantic content from the context which are then used to generate the observed sentences [12, 21]. Finally, a ranking approach has been applied to rank everything in the associated candidate set higher than records observed in the rest of the data [22]

In this paper we examine a slightly different form of ambiguous supervision. Typically the set of potential meanings are represented as an enumerated list. The task then is to find a subset of that list that are referred to by the sentence. However, entities in the ambiguous context are often related. We

represent such relationships explicitly by creating edges between them, thus forming a graph instead of a list. In particular, we look at the task of learning navigation instructions in which the goal is to map the sentences to an unobserved navigation plan. Each step in the plan follows a strict temporal order and can refer to any number of entities in the environment. The space of possible plans is represented using a graph and our goal is to find the appropriate subgraph that corresponds to the actual plan specified by the instructions.

2. NAVIGATION PROBLEM AND EVALUATION DATA

For our navigation task, the ambiguous supervision we receive is in the form of observations of how humans behave when following navigation instructions. Defined more formally, the system is given training data in the form: $\{(e_1, a_1, w_1), (e_2, a_2, w_2), \ldots, (e_n, a_n, w_n)\}$, where e_i is a natural language instruction, a_i is an observed action sequence, and w_i is a description of the current state of the world including the patterns of the floors and walls and positions of any objects.

The main challenge of this problem is that the navigation plans described by the instructions are not directly observed. Usually several different plans can be used to navigate the same route. In other words, there is not always a direct correspondence between e_i and a_i. Rather, e_i corresponds to an unobserved plan p_i that when executed in w_i will produce a_i. Our goal then is to infer the correct p_i from the training data.

To train and test our system, we use the data and virtual environments assembled by MacMahon *et al.* [23]. The data was collected for three different virtual worlds consisting of interconnecting hallways. An overhead view of one of the worlds is shown in Fig. 1. Each world consists of several short concrete hallways and seven long hallways, each with a different floor pattern (grass, brick, wood, gravel, blue, flower, and yellow octagons). The worlds are divided into three areas, each with a different painting on the walls (butterfly, fish, and Eiffel Tower). There is also furniture placed at various intersections (hatrack, lamp, chair, sofa, barstool, and easel). The three worlds contain the same elements but in different configurations. Each world also has seven chosen locations labeled 1 thorough 7.

MacMahon *et al.* collected both human instructor data and human follower data. The instructors first familiarized themselves with the environment and the seven chosen locations. They were then asked to give a set of written instructions on how to get from a particular location to another. Since they did not have access to the overview map, they had to rely on their explorations of the environments. These instructions were then given to several human followers whose actions were recorded as they tried to follow the instructions. All the actions are discrete and consist of turning left, turning right, and moving from one intersection to another.

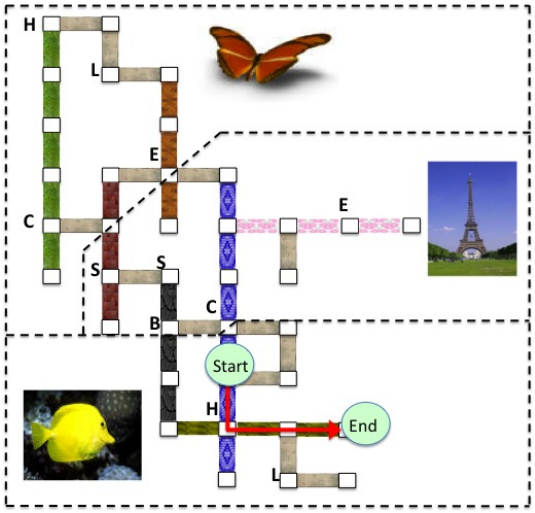

Fig. 1. This is an example of a route in our virtual world. The world consists of interconnecting hallways with varying floor tiles and paintings on the wall (butterfly, fish, or Eiffel Tower.) Letters indicate objects (e.g. 'C' is a chair) at an intersection.

Instruction: "Go away from the lamp to the intersection of the red brick and wood"

Basic:
Turn ( ) ,
Travel (steps: 1)

Landmarks:
Turn ( ) ,
Verify (left: WALL , back: LAMP , back: HATRACK , front: BRICK HALL) ,
Travel (steps: 1) ,
Verify (side: WOOD HALL)

Fig. 2. Examples of automatically generated plans.

3. ALGORITHM

Since the training data only provides the actions observed, we must first construct the possible plans that led to those actions. A simple way to generate such plans is to model the observed actions directly. In our case, this means forming plans that consist of only turning left and right, and walking forward a certain number of steps. This is often sufficient if the instruction directly refers to the specific action to be taken (e.g. *turn left*, *walk forward two steps*). We refer to these navigation plans which capture such direct instructions as *basic plans*.

To capture more complex instructions that refer to objects and places in the environment (e.g. *face the pink flower hallway*, *go to the sofa*), we simulate executing the given actions in the environment. We collect sensory data during the execution and form a *landmarks plan* that adds interleaving verification steps to the *basic plan*. The verification steps specify the landmarks that should be observed after executing each basic action. Examples of both a *basic plan* and a *landmarks plan* are shown in Fig. 2.

The *basic plan* generally underestimates what the true plan looks like while the *landmarks plan* overestimates.

Algorithm 1 LEXICON LEARNING

input Navigation instructions and the corresponding navigation plans $(e_1, p_1), \ldots, (e_n, p_n)$
output $Lexicon$, a set of phrase-meaning pairs

main
  for n-gram w that appears in $\mathbf{e} = (e_1, \ldots, e_n)$ **do**
    for instruction e_i that contains w **do**
      Add navigation plan p_i to $meanings(w)$
    end for
    repeat
      for every pair of meanings in $meanings(w)$ **do**
        Add intersections of the pair to $meanings(w)$
      end for
      Keep k highest-scoring entries of $meanings(w)$
    until $meanings(w)$ converges
    Add entries of $meanings(w)$ with scores higher than threshold t to $Lexicon$
  end for
end main

Thus, we distill the *landmarks plan* to recover the actual plan referred to by the instructions. We employ a lexicon learning algorithm to first learn the meanings of words and short phrases. The learned lexicon is then used to identify and remove extraneous components in the *landmarks plan*.

3.1. Learning a lexicon

We build a semantic lexicon by finding the common parts of the formal representations associated with different occurrences of the same word or phrase [19]. More specifically, we represent the navigation plans in graphical form and compute common parts by taking intersections of the two graphs [24]. Pseudo-code for the approach is shown in Algorithm 1. Initially, all navigation plans whose instruction contains a particular n-gram w are added to $meanings(w)$, the set of potential meanings of w. Then, the algorithm repeatedly computes the intersections of all pairs of potential meanings and adds them to $meanings(w)$ until further intersections do not produce any new entries. The intersection operation is performed by greedily removing the largest common subgraph from both graphs until the two graphs have no overlapping nodes. The output of the intersection process consists of all the removed subgraphs. An example of the intersection operation is shown in Fig. 3. Each potential word-meaning pair is given a *score* (described below) that evaluates its quality. After $meanings(w)$ converges, its members with scores higher than a given threshold are added as lexical entries for w. In our experiments, we consider only unigrams and bigrams, and use threshold $t = 0.4$ and maximum meaning set size $k = 100$.

We use the following scoring function to evaluate a pair of an n-gram w and a graph g:

$$Score(w, g) = p(g|w) - p(g|\neg w)$$

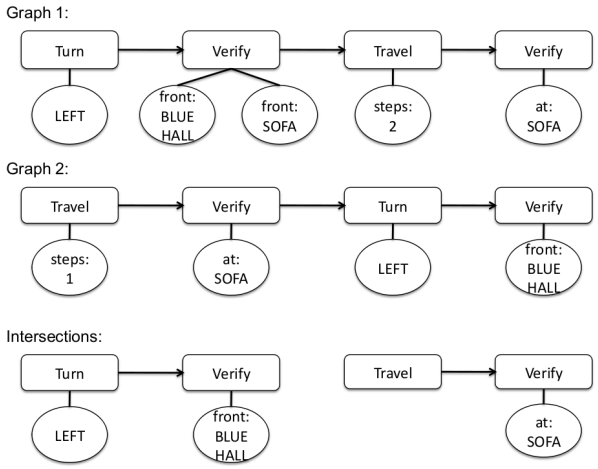

Fig. 3. An example of computing the intersections of two graph representations of navigation plans.

Intuitively, the score measures how much more likely a graph g appears when w is present compared to when it is not. A good (w, g) pair means that w should be indicative of g appearing (i.e. $p(g|w)$ should be close to 1), assuming w is monosemous[1]. However, the reverse is not true since an object or action may often be referred to using other expressions or omitted from an instruction altogether. Thus, the absence of a word w when g occurs, $p(\neg w|g)$, is not evidence against g being the meaning of w. To penalize g's that are ubiquitous, we subtract the probability of g occurring when w is not present. We estimate all the probability measures by counting how many examples contain the words or the graphs, ignoring multiple occurrences in a single example.

3.2. Refining navigation plans using the lexicon

The learned lexicon is then used to remove extraneous components from the *landmarks plans*. To refine (e_i, p_i), we first select the highest-scoring lexical entry (w, g) such that w and g appear in e_i and p_i, respectively. We then remove w from e_i and mark all occurrences of g in p_i, ignoring any redundant markings. This process is repeated until no words remain in e_i or no more lexical entries can be selected. Finally, we remove all nodes in p_i that were not marked and the remaining graph becomes the new *refined plan* p_i'.

4. EXPERIMENTAL EVALUATION

To evaluate our approach, we use the instructions and follower data collected by MacMahon *et al.* [23] to train and test our system. The data contains 706 non-trivial route instructions for the three virtual worlds. The instructions were produced

[1]Notice that the actual magnitude of $p(g|w)$ matters. Thus, using odds ratios as the scoring function did not work as well.

# instructions	3236
Vocabulary size	629
Avg. # words	7.8 (5.1)
Avg. # actions	2.1 (2.4)

Table 1. Statistics for the segmented version of the corpus collected by MacMahon et al. The average statistics for each instruction are shown with standard deviations in parentheses.

	Precision	Recall	F1
Basic plans	81.47	56.04	66.40
Landmarks plans	45.39	85.56	59.31
Refined landmarks plans	80.59	77.49	79.01
Refined landmarks plans (no temporal links)	80.54	68.87	74.25

Table 2. Partial matching accuracy of the plans

by six instructors for 126 unique starting and ending location pairs spread evenly across the three worlds. There were 1 to 15 human followers for each instruction.

Since this data was originally collected only for testing purposes and not for learning, each instruction is quite long with an average of 5 sentences. However, for learning, it is more natural to observe the instructors interact with the followers as they progress. Thus, to create our training data, we first segmented the instructions into individual sentences and aligned them with the corresponding action sequences. Statistics for the segmented data can be seen in Table 1.

4.1. Methodology

To examine how well our system infers the correct navigation plans from the observed actions, we hand-annotated each instruction with the correct navigation plan and compared the inferred plans to these gold-standard plans. We used a partial correctness metric to measure the precision and recall of the inferred plans. To calculate precision, each step in the inferred plan receives one point if it matches the type of the corresponding step in the gold-standard plan. An additional point is then awarded for each matching argument. Precision is then computed as the sum of the points divided by the total number of possible points. Since the two plans may contain different number of steps, we use a dynamic programming algorithm to find a order-preserving mapping of steps from one plan to the other such that precision is maximized. Recall is computed similarly with the roles of the inferred and gold-standard plans swapped. Finally, we also compute F1 scores, the harmonic mean of precision and recall.

To evaluate the *basic* and *landmarks plans*, we simply computed their average accuracies. For the *refined landmarks plans*, we built a lexicon from all the examples and then used it to refine the *landmarks plans*.

4.2. Results

The results are shown in Table 2. As mentioned previously, the *basic plans* tend to underestimate the true plans, so they have high precision but low recall. On the other hand, the *landmarks plans* include a lot of extraneous details so they have high recall but low precision. Using our method for extracting the relevant parts of the proposed plans, we achieve both high precision and recall, recovering over three quarters of the gold-standard plans.

To examine the utility of modeling the temporal constraints of the actions, we also performed an ablation experiment where we removed the directed edges between the actions during lexicon learning. This is similar to how ambiguous context are usually modeled. As can be seen from the results, precision stays about the same while recall drops. This is because the lexicon can no longer handle phrases that map to multiple actions.

5. DISCUSSIONS AND FUTURE WORK

Resolving ambiguous supervision is only part of the larger task of learning the semantics of language. We have also used the learning approach described here to implement an end-to-end navigation system that learns to interpret novel navigation instructions and execute the parsed plans [25].

Other than our refinement method, other learning techniques can also be adapted to handle the relationships between entities. For example, a generative model could be restricted to produce only sequence of actions that follow the correct temporal ordering.

6. CONCLUSIONS

Solving the ambiguous supervision problem is central to the language grounding problem. We have presented a different approach to modeling this problem by representing the space of possible meanings as a graph rather than a list. This allows us to include information about the relationships between the entities we are grounding to. We presented an algorithm for finding the relevant subgraphs given the natural language sentences by first learning a lexicon and then using the lexicon to remove irrelevant parts of the graph. While this approach yielded promising results in recovering the correct semantic content, there remains a lot of room for improvement with the use of more sophisticated machine learning techniques.

Acknowledgments

This work was funded by the NSF grants IIS-0712097 and IIS-1016312.

7. REFERENCES

[1] Shinichi Satoh, Yuichi Nakamura, and Takeo Kanade, "Name-it: Naming and detecting faces in video by the integration of image and natural language processing," in *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97)*, 1997.

[2] Tamara L. Berg, Alexander C. Berg, Jaety Edwards, and D. A. Forsyth, "Who's in the picture," in *Advances in Neural Information Processing Systems 17 (NIPS 2004)*, 2004.

[3] Michael Fleischman and Deb Roy, "Situated models of meaning for sports video retrieval," in *Proceedings of Human Language Technologies: The Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT-07)*, Rochester, NY, 2007.

[4] Sonal Gupta and Raymond Mooney, "Using closed captions to train activity recognizers that improve video retrieval," in *Proceedings of the CVPR-09 Workshop on Visual and Contextual Learning from Annotated Images and Videos (VCL)*, Miami, FL, June 2009.

[5] Yansong Feng and Mirella Lapata, "How many words is a picture worth? automatic caption generation for news images," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL-10)*, 2010.

[6] P. Gorniak and D. Roy, "Speaking with your sidekick: Understanding situated speech in computer role playing games," in *Proceedings of the 4th Conference on Artificial Intelligence and Interactive Digital Entertainment*, Stanford, CA, 2005.

[7] Wesley Kerr, Paul R. Cohen, and Yu-Han Chang, "Learning and playing in Wubble World," in *Proceedings of the Fourth Artificial Intelligence for Interactive Digital Entertainment Conference (AIIDE)*, Palo Alto, CA, October 2008.

[8] Shaolin Qu and Joyce Y. Chai, "Context-based word acquisition for situated dialogue in a virtual world," *Journal of Artificial Intelligence Research*, vol. 37, pp. 247–277, 2010.

[9] Hilke Reckman, Jeff Orkin, and Deb Roy, "Learning meanings of words and constructions, grounded in a virtual game," in *10th Conference on Natural Language Processing (KONVENS)*, 2010.

[10] David L. Chen, Joohyun Kim, and Raymond J. Mooney, "Training a multilingual sportscaster: Using perceptual context to learn language," *Journal of Artificial Intelligence Research*, vol. 37, pp. 397–435, 2010.

[11] Benjamin Snyder and Regina Barzilay, "Database-text alignment via structured multilabel classification," in *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI-2007)*, 2007.

[12] Percy Liang, Michael I. Jordan, and Dan Klein, "Learning semantic correspondences with less supervision," in *Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL-IJCNLP)*, 2009.

[13] S.R.K. Branavan, Harr Chen, Luke S. Zettlemoyer, and Regina Barzilay, "Reinforcement learning for mapping instructions to actions," in *Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL-IJCNLP)*, 2009.

[14] Tessa Lau, Clemens Drews, and Jeffrey Nichols, "Interpreting written how-to instructions," in *Proceedings of the Twenty-first International Joint Conference on Artificial Intelligence (IJCAI-2009)*, 2009.

[15] Nobuyuki Shimizu and Andrew Haas, "Learning to follow navigational route instructions," in *Proceedings of the Twenty-first International Joint Conference on Artificial Intelligence (IJCAI-2009)*, 2009.

[16] Cynthia Matuszek, Dieter Fox, and Karl Koscher, "Following directions using statistical machine translation," in *Proceedings of the 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2010.

[17] Thomas Kollar, Stefanie Tellex, Deb Roy, and Nicholas Roy, "Toward understanding natural language directions," in *Proceedings of the 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2010.

[18] Adam Vogel and Dan Jurafsky, "Learning to follow navigational directions," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL-10)*, 2010.

[19] Jeffrey M. Siskind, "A computational study of cross-situational techniques for learning word-to-meaning mappings," *Cognition*, vol. 61, no. 1, pp. 39–91, Oct. 1996.

[20] Rohit J. Kate and Raymond J. Mooney, "Learning language semantics from ambiguous supervision," in *Proceedings of the Twenty-Second Conference on Artificial Intelligence (AAAI-07)*, Vancouver, Canada, July 2007, pp. 895–900.

[21] Joohyun Kim and Raymond J. Mooney, "Generative alignment and semantic parsing for learning from ambiguous supervision," in *Proceedings of the 23rd International Conference on Computational Linguistics (COLING-10)*, 2010.

[22] Antoine Bordes, Nicolas Usunier, and Jason Weston, "Label ranking under ambiguous supervision for learning semantic correspondences," in *Proceedings of 27th International Conference on Machine Learning (ICML-2010)*, 2010.

[23] Matt MacMahon, Brian Stankiewicz, and Benjamin Kuipers, "Walk the talk: Connecting language, knowledge, and action in route instructions," in *Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI-06)*, 2006.

[24] Cynthia A. Thompson and Raymond J. Mooney, "Acquiring word-meaning mappings for natural language interfaces," *Journal of Artificial Intelligence Research*, vol. 18, pp. 1–44, 2003.

[25] David L. Chen and Raymond J. Mooney, "Learning to interpret natural language navigation instructions from observations," in *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI-11)*, 2011.