Synonymous Generalization in Sequence-to-Sequence Recurrent Networks

Ning Shi
Georgia Institute of Technology
ning.shi@gatech.edu

Abstract

When learning a language, people can quickly expand their understanding of the unknown content by using compositional skills, such as from two words "go" and "fast" to a new phrase "go fast." In recent work of Lake and Baroni (2017), modern Sequence-to-Sequence (seq2seq) Recurrent Neural Networks (RNNs) can make powerful zero-shot generalizations in specifically controlled experiments. However, there is a missing regarding the property of such strong generalization and its precise requirements. This paper explores this positive result in detail and defines this pattern as the synonymous generalization, an ability to recognize an unknown sequence by decomposing the difference between it and a known sequence as corresponding existing synonyms. To better investigate it, I introduce a new environment called Colorful Extended Cleanup World (CECW), which consists of complex commands paired with logical expressions. While demonstrating that sequential RNNs can perform synonymous generalizations on foreign commands, I conclude their prerequisites for success. I also propose a data augmentation method, which is successfully verified on the Geoquery (GEO) dataset, as a novel application of synonymous generalization for real cases.

1 Introduction

In general, it is easier to learn new words with familiar synonyms than to learn new words without synonyms (Webb, 2007; Nation, 2001). A person who knows the verb "go" can understand the meaning of a new verb "move" if he has already noticed that they are synonymous.[1] In addition, as part of human cognition, systematic compositionality or algebraic compositionality helps people to understand and generate novel utterances from acquired primitives (Chomsky, 1957; Montague, 1970). Therefore, the person's understanding of the new verb "move" can suddenly generalize to most structurally related sentences, for instance, from "go fast," "go downstairs," and "ready to go" to "move fast," "move downstairs," and "ready to move."

Recent achievements in a wide range of tasks have benefited from the impressive generalization capability of Deep Neural Networks (DNNs) (Yann et al., 2015). However, what supports these models is the dependence on a large number of training data, rather than systematic generalization (Lake et al., 2017). Weak systematic compositionality has been considered as a primary obstacle to the expression of language and thought in connectionist networks for a long time (Fodor and Pylyshyn, 1988; Hadley, 1994; Marcus, 1998; Fodor and Lepore, 2002; Frank et al., 2009; Brakel and Frank, 2009; Marcus, 2018). The research of Lake and Baroni (2017) proposed to use the Simplified version of the CommAI Navigation (SCAN) dataset a new benchmark for compositional learning.[2] SCAN is a simple language-driven navigation environment where an agent learns to translate commands (e.g., "jump left") into actions (e.g., "LTURN JUMP"). According to previous research, modern RNNs fail in cases that systematic compositionality is required, but they can perform an excellent zero-shot generalization in controlled experiments (Lake and Baroni, 2017; Loula et al., 2018; Bastings et al., 2018). How do such generalization mechanisms apply to subtend the zero-shot learning in these experiments? This is crucial to understand the black box of DNNs but remains an open problem.

In this paper, I revisit this particular kind of compositional learning and define it as the synonymous

[1] I consider two verbs "go" and "move" a pair of synonyms because they express similar behavior semantically.

[2] SCAN at: https://github.com/brendenlake/SCAN.

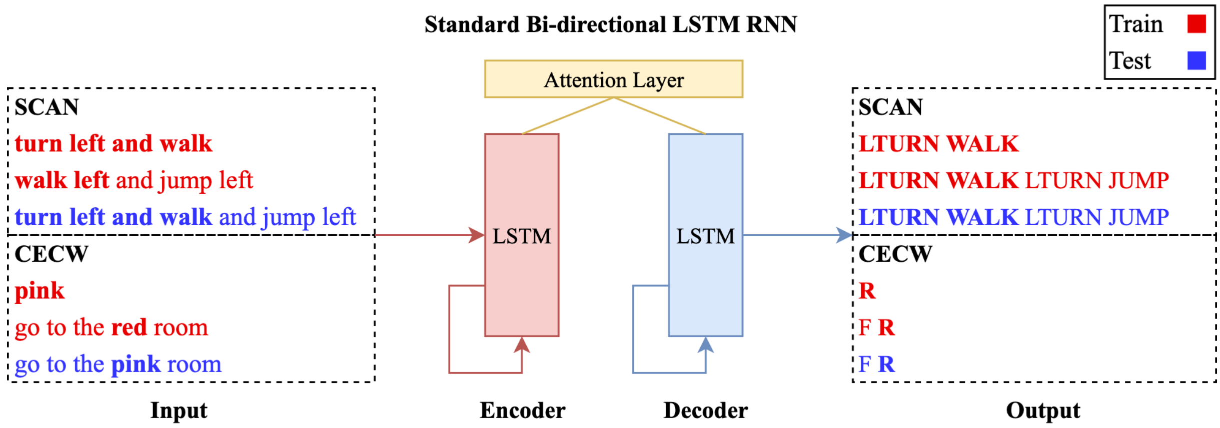

Figure 1: Modern sequential RNNs with LSTM units and attention mechanismsin SCAN and CECW. Sequences in red and blue represent training and test samples, respectively. The model should learn to translate an input command to its output representation. In each domain, the sequence that exists alone in bold is the primitive, and corresponding decomposition in bold is a synonym of that primitive. A model can do a synonymous generalization by predicting the blue command given two red commands.

generalization. Specifically, in Figure 1, an agent can make a synonymous generalization by correctly translating the foreign command "**turn left and walk** and jump left" after exposed to "**walk left** and jump left" and "**turn left and walk**" (See Section 2 for details). To make a better empirical study of synonymous generalization, I introduce a mobile-manipulation domain, Colorful Extended Cleanup World (CECW), where a robot can interact with distinctly colored rooms and movable objects according to commands received. With its diverse primitive colors, clear structures, and rich logical connections, CECW is appropriate to study the synonymous generalization (See Section 3 for details). By incrementing the primitives in the training set, the experiment proves that carrying primitives is the fundamental reason why a stranger can be correctly identified. At the same time, a data augmentation method is proposed and used as a practical application of synonymous generalization in real situations. Through experiments on Geoquery (GEO), it can be found that adding primitives in the training set can increase the model's awareness of corresponding compositions in test queries, so as to obtain a better outcome.

2 Synonymous Generalization

Following the work of Lake and Baroni (2017), our emphasis is placed on a particular case of the compositional skills, namely the synonymous generalization. Two sequences are considered synonymous when both of them point to the same semantic representation. In SCAN, "turn left and walk" and "walk left" are synonymous as they both map to "LTURN WALK." Later, I am able to define the synonymous generalization as the ability to accurately predict an unknown sequence by replacing decompositions of a known sequence with given synonyms. In Figure 1, a robot successfully generalize its existing knowledge to an unknown command "**turn left and walk** and jump left" by replacing the decomposition "**walk left**" of "**walk left** and jump left" with a given synonym "**turn left and walk**." In the whole process, synonyms that stand alone in the training set are referred to primitives. Their sequence pairs are primitive rules.

3 CECW

As the name suggests, the Colorful Extended Cleanup World (CECW) dataset[3]is a color-extended version of the Cleanup World (CW) borrowed from the mobile-manipulation robot domain (MacGlashan et al., 2015). CW refers to a world equipped with a movable object as well as four rooms in four colors, including "blue," "green," "red," and "yellow," which is designed as a simulation environment where the agent can act based on the instructions received (Gopalan et al., 2018). CW obeys a particular Geometric Linear Temporal Logic (GLTL) to parse commands by grammatical syntax[4], resulting in a total of 3382 commands

[3]CECW at: https://github.com/MrShininnnnn/CECW.

[4]In the grammar, $\wedge$, $\vee$, and $\neg$ denote logical conjunctions of "and", "or", "not"; $\square$ denotes "always"; $\diamond$ denotes "eventually." In the textual expression, symbols "&", "|", and "!" indicate logical connections of "and", "or", and "not"; "F" indicates "eventually"; "G" indicates "always"; "X" and "Z"

CW&CECW Command	GLTL Formula	Textual Expression
go to the red room	$\Diamond R$	F R
go to the red room and then go to the blue room	$\Diamond(R \wedge \Diamond B)$	F & R F B
go to red room but do not enter yellow room	$\Diamond R \wedge \neg \Box Y$	& F R G ! Y
go through the red or blue room to the yellow room	$\Diamond((R \vee B) \wedge \Diamond Y)$	F & | R B F Y
push the chair from the red room into the blue room	$\Diamond(R \wedge \Diamond X)$	F & R F X
go to the red room move chair to the green room	$\Diamond(R \wedge \Diamond Z)$	F & R F Z
GEO Request	Query	
how big is alaska ?	A,(size(B,A),const(B,stateid(alaska)))	
how many citizens in boulder ?	A,(population(B,A),const(B,cityid(boulder)))	
which state is the smallest ?	A,smallest(A,state(A))	

Table 1: Example sequence pairs in the CW, CECW, and GEO dataset. CW and CECW share the same domain. Source or input sequences are in left column, and target or output sequences are in right column.

reflecting 39 GLTL expressions. In addition, commands can be represented in textual expressions as shown in Table 1. For example, a robot should eventually stay in the red room ("$\Diamond$ R" or "F R") if it hears the command "go to the red room."

In this work, the problem in CW is formatted as a supervised semantic parsing task to translate commands (e.g., "go to the red room") to their textual expressions (e.g., "F R"). For the generation of the CECW, duplicated samples are deleted, resulting in 2130 unique sequence pairs. For the existing four colors, each color is added two synonyms ("purple" and "navy" for "blue," "olive" and "lime" for "green," "pink" and "orange" for "red," and "brown" and "tan" for "yellow"), agreeing to four target entities ("B," "C," "R," and "Y") respectively. On this basis, the color words in the commands are replaced with their two synonyms and keep them paired with the same textual expressions. In the case of "go to the **red** room," there are now two new commands, namely "go to the **pink** room" and "go to the **orange** room," which map to the same expression "F **R**." I call "pink" and "orange" two primitives, and "pink" $\rightarrow$ "R" and "orange" $\rightarrow$ "R" two primitive rules. As a result, CECW contains 188 different tokens from the source side and 11 from the target side. The overall data size increases from 2130 to 11153.

The CECW dataset are then randomly into a training set (80%) and a test set(20%). All the commands with primitives are removed from the training set. By adding primitives back according to the order of their target entities "B," "C," "R," and "Y," I create six subsets including ***Colorless*** for none synonymous primitives, ***B*** for "purple" and "navy", ***BC*** for "olive" and "lime", ***BCR*** for "pink" and "orange", ***BCRY*** for "brown" and "tan", and ***ALL*** for the original train set. On the one hand, CECW is similar to SCAN, with a clear grammatical structure, which makes it straightforward to verify compositional learning. On the other hand, the addition of three logical structures (and, or, not) leads CECW more comparable to the real-life context, as well as more challenging. Furthermore, various colors in CECW are used as the natural primitives to test for synonymous generalization. Therefore, CECW is a more appropriate locale for incremental experimentation here.

indicate the interaction with movable objects.

4 Experiments

This paper focuses on the model with the same structure as the model determined by Lake and Baroni (2017), which is a standard bi-directional long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) RNNs with attention mechanism (Bahdanau et al., 2014) under the seq2seq framework (Sutskever et al., 2014). The goal is to learn a function mapping source sequences $q = x_1...x_{|q|}$ to target sequence $a = y_1...y_{|a|}$. The conditional probability $p(a|q)$ is defined as:

$$p(a|q) = \prod_{|a|}^{t=1} p(y_t|y_{<t}, q)$$

,where $y_{<t} = y_1...y_{t-1}$. As shown in Figure 1, the model mainly consists of an encoder to extract features from q as a hidden state, a decoder to generate $y_1, ..., y_{|a|}$ conditioned on the encoding matrix, and an attention layer as the intermediate step. I implement the model in TensorFlow (Abadi et al., 2016). Training configurations and model hyperparameters are reported in Table 2.

Batch Size	32	Training Epoch	64 or 128	Dropout Rate	0.5
Embedding Size	256	Hidden Size	256	Initial Parameters	±0.08
#Encoder Layer	2	#Decoder Layer	1	Adam Learning Rate	0.0005

Table 2: Training Configurations and Model Hyperparameters. I froze the training epoch as 64 for CECW and 128 for GEO. All the experiments share the same model structure and hyperparameters. Instead of employing pre-trained embeddings, I generate all parameters through Gaussian initialization to prevent prior knowledge from contaminating our experiments.

4.1 Experiment 1: CECW

In this incremental experiment involving six CECW subsets, I aim to verify the synonymous generalization and confirm that the addition of primitive is the key. As shown in Figure 1, a good learner should generalize its understanding of "go to the **red** room" to "go to the **pink** room" with the assistance of the given primitive "pink" $\rightarrow$ "R."

Results As elaborated in Table 3, the participation of two primitive rules in each data augmentation does help to steadily improve prediction, especially in sequence accuracy (from 50% on ***Colorless*** to 96% on ***BCRY***). The model trained on ***BCRY*** achieved a competitive result using only a quarter of the data used by the baseline on ***ALL***.

Verification Apparently, the performance of the consistent model goes up with the addition of primitive from ***Colorless*** to ***ALL***. This incremental setting suggests that the attendance of primitives is the only explanation for the impressive zero-shot learning, and, consequently, I state that such type of learning skills is exactly the synonymous generalization defined earlier.

Logical Processing Meanwhile, the good processing of multiple logical structures proves the potential of synonymous generalization in practical natural language applications. In contrast to sequential actions in SCAN, textual expressions in CECW retain three main logical conjunctions, that is, "and," "or," and "not."[5] The high zero-shot accuracy, to a certain extent, ensures the possibility of transferring this magical generalization ability to more real cases.

Requirements An interesting fact is that making synonymous generalization does not require an explicitly expressed synonymous association between the primitive and the decomposition of the training sequence. There is no advance step to tell the model, for instance in Figure 1, "red" and "pink" are a pair of synonyms. Thus, I conclude that a model is able to perform a successful synonymous generalization when the difference between a training sequence pair and a test sequence pair is a given primitive, which also means that both sequences should have the same target output.

Independence It is observed that models trained on ***BC*** and ***BCRY*** recognize "go to the **navy** room and then the **lime** room" with the assistance of two primitive rules "**navy**" $\rightarrow$ "B" and "**lime**" $\rightarrow$ "C." A total of 55 (2.4%) test commands are related to two primitive rules. I find commands that need one or two primitive rules are well learned, but it becomes tough when involving more. It might be the result of a more complex logical structure and a longer sequence length as well. Thus, I report that synonymous generalizations have a certain degree of independence.

4.2 Experiment 2: GEO

The goal of the second experiment is to take advantage of the synonymous generalization. Previous results recommend decomposing training samples as primitive rules to join the training as well. In this experiment, I show how to treat entity names as primitive to augment the training data for reaching higher accuracy.

Data GEO consists of user requests paired with database queries (see Table 1). It contains 880 instances which are randomly grouped into 704 samples for training and 176 samples for testing (Zettlemoyer and Collins, 2012).The entity names of the city, state, country, and river are considered as primitives such as "new york", "ohio", and "rio grande." Given that, a primitive rule can be "new york" $\rightarrow$ "new york." There are two subsets for comparison. ***GEO Normal*** is exactly the training set, while ***GEO Augmented*** is ***GEO Normal*** in addition to 101 primitive rules for entity names.

[5] The construction of some commands such as "*go through the red or blue room to the yellow room*" needs the assistance of multiple logical connections (see Table 1).

Condition	Acc. ± s.d.	Seq Acc. ± s.d.	Top 5 Seq Acc. ± s.d.	Train Size
		Experiment 1		
Colorless	$91.72 \pm 0.44\%$	$48.62 \pm 2.85\%$	$97.59 \pm 1.47\%$	2118
B	$92.97 \pm 0.16\%$	$56.61 \pm 1.04\%$	$97.55 \pm 1.03\%$	2120
BC	$95.48 \pm 0.20\%$	$72.44 \pm 1.42\%$	$98.95 \pm 0.31\%$	2122
BCR	$97.16 \pm 0.10\%$	$83.33 \pm 0.74\%$	$99.51 \pm 0.11\%$	2124
BCRY	$\mathbf{99.15 \pm 0.14\%}$	$\mathbf{96.42 \pm 0.93\%}$	$\mathbf{99.91 \pm 0.03\%}$	**2126**
ALL	$99.99 \pm 0.00\%$	$99.96 \pm 0.00\%$	$100.00 \pm 0.00\%$	8922
		Experiment 3		
GEO Normal	$97.58 \pm 0.13\%$	$69.20 \pm 3.20\%$	$78.07 \pm 1.11\%$	704
GEO Augmented	$\mathbf{97.81 \pm 0.05\%}$	$\mathbf{75.68 \pm 1.58\%}$	$\mathbf{82.16 \pm 0.65\%}$	**805**

Table 3: Evaluation matrix of synonymous generalization verification on CECW in Section 4.1, as well as an application on GEO in Section 4.2. The evaluation covers 4 indicators including accuracy, sequence accuracy, top-5 sequence accuracy and the number of used samples in training. Specifically, the accuracy marks the correct predictions on token-level as a percentage of the target sequence length. The sequence accuracy stands for the right predictions on sequence-level as a percentage of the test size. For the top-5 accuracy, any of the five highest probability predictions matching the expected answer is an accurate one.

Results It is obvious in Table 3 that including 101 primitive rules did result in a significant improvement from 69% to 75% in sequence accuracy, as well as from 78% to 82% in top-5 accuracy.

Rare Entity Compared with the model trained on ***GEO Normal***, the one trained on ***GEO Augmented*** earns more credits by correctly detecting the rare entities such as “new york,” “new mexico,” and “new jersey.” I find repeating these state names as primitives in the training phase helps the model structurally treat these entities rather than predict the next token for “new.”

Data Augmentation Many data augmentation methods aim to seek cases of poor handling and to supplement the training data with complete patterns on purpose. However, the additional training samples in ***GEO Augmented*** are simply name entities pointing to themselves. On the basis of this finding, I would like to propose that parsing and decomposing the training set into primitive rules to augment existing training data might improve the performance of the model directly.

5 Related Work

It is said that humans are highly systematic in language learning, and that cognitive ability is generally applicable to structure-related contexts (Chomsky, 1957). The algebraic compositionality explains the powerful generalization derived from limited knowledge (Montague, 1970). Whether neural networks or connectionist approaches can simulate and display systematicity has been controversial for a long time (Fodor and Pylyshyn, 1988; Hadley, 1994; Marcus, 1998; Phillips, 1998; Wong and Wang, 2007; Brakel and Frank, 2009). RNNs were later considered as one of the major candidates for reproducing the generalization ability of human in language learning (Frank et al., 2014; Bowman et al., 2015; Russin et al., 2019). Partial recurrent networks have demonstrated their potential for compositional skills in zero-shot learning (Liška et al., 2018). Lake and Baroni (2017) introduced SCAN to encourage studying compositionality of recurrent seq2seq models. As a flipped version of SCAN, NACS was later proposed as a supplement to benchmarks (Bastings et al., 2018). However, both SCAN and NACS fall into the straightforward language-driven navigation environment, which is far from the real-word language scenarios. Besides, to the best of our knowledge, empirical research on the strong zero-shot learning of RNNs in controlled experiments is still lacking.

6 Conclusion

In this paper, I defined the powerful compositional learning of seq2seq models in previous studies as the synonymous generalization. I introduced a new dataset, CECW, and showed how the addition of primitives assisted the RNNs to generalize their knowledge to make strong zero-shot learning in incremental experimentation. I also showed that the synonymous generalization based data augmentation method improved the model’s performance in the GEO dataset.

Acknowledgments I would like to express great appreciation to Haotian Zhang, Keran Zhao, Yewen Pu, and Ziheng Zeng for very helpful discussions.

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Joost Bastings, Marco Baroni, Jason Weston, Kyunghyun Cho, and Douwe Kiela. 2018. Jump to better conclusions: SCAN both left and right. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 47–55. Association for Computational Linguistics.

Samuel R Bowman, Christopher D Manning, and Christopher Potts. 2015. Tree-structured composition in neural networks without tree-structured architectures. *arXiv preprint arXiv:1506.04834*.

Philémon Brakel and Stefan Frank. 2009. Strong systematicity in sentence processing by simple recurrent networks. In *31th Annual Conference of the Cognitive Science Society (COGSCI-2009)*, pages 1599–1604. Cognitive Science Society.

Noam Chomsky. 1957. *Syntactic Structures*. Mouton, Berlin, Germany.

Jerry A Fodor and Ernest Lepore. 2002. *The compositionality papers*. Oxford University Press.

Jerry A Fodor and Zenon W Pylyshyn. 1988. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71.

Stefan L Frank, P Calvo, and J Symons. 2014. Getting real about systematicity. *The architecture of cognition: Rethinking Fodor and Pylyshyns systematicity challenge*, pages 147–164.

Stefan L Frank, Willem FG Haselager, and Iris van Rooij. 2009. Connectionist semantic systematicity. *Cognition*, 110(3):358–379.

Nakul Gopalan, Dilip Arumugam, LL Wong, and Stefanie Tellex. 2018. Sequence-to-sequence language grounding of non-markovian task specifications. In *Robotics: Science and Systems*.

Robert F Hadley. 1994. Systematicity in connectionist language learning. *Mind & Language*, 9(3):247–272.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Brenden M Lake and Marco Baroni. 2017. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. *the 35th International Conference on Machine Learning (ICML 2018)*.

Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. 2017. Building machines that learn and think like people. *Behavioral and brain sciences*, 40.

Adam Liška, Germán Kruszewski, and Marco Baroni. 2018. Memorize or generalize? searching for a compositional rnn in a haystack. *arXiv preprint arXiv:1802.06467*.

Joao Loula, Marco Baroni, and Brenden Lake. 2018. Rearranging the familiar: Testing compositional generalization in recurrent networks. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 108–114. Association for Computational Linguistics.

James MacGlashan, Monica Babes-Vroman, Marie desJardins, Michael L Littman, Smaranda Muresan, Shawn Squire, Stefanie Tellex, Dilip Arumugam, and Lei Yang. 2015. Grounding english commands to reward functions. In *Robotics: Science and Systems*.

Gary F Marcus. 1998. Rethinking eliminative connectionism. *Cognitive psychology*, 37(3):243–282.

Gary F Marcus. 2018. *The algebraic mind: Integrating connectionism and cognitive science*. MIT press.

Richard Montague. 1970. Universal grammar. *Theoria*, 36(3):373–398.

Ian SP Nation. 2001. *Learning vocabulary in another language*. Ernst Klett Sprachen.

Steven Phillips. 1998. Are feedforward and recurrent networks systematic? analysis and implications for a connectionist cognitive architecture. *Connection Science*, 10(2):137–160.

Jake Russin, Jason Jo, and Randall C O'Reilly. 2019. Compositional generalization in a deep seq2seq model by separating syntax and semantics. *arXiv preprint arXiv:1904.09708*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Stuart Webb. 2007. The effects of synonymy on second-language vocabulary learning. *Reading in a Foreign Language*, 19(2):120–136.

Francis CK Wong and William SY Wang. 2007. Generalisation towards combinatorial productivity in language acquisition by simple recurrent networks. In *2007 International Conference on Integration of Knowledge Intensive Multi-Agent Systems*, pages 139–144. IEEE.

LeCun Yann, Bengio Yoshua, and Hinton Geoffrey. 2015. Deep learning. *Nature*, 521:436–444.

Luke S Zettlemoyer and Michael Collins. 2012. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. *in Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence (UAI2005).*