2015-08-26

Similarity Reasoning over Semantic Context-Graphs

Adrian Boteanu
Worcester Polytechnic Institute

SIMILARITY REASONING OVER SEMANTIC CONTEXT–GRAPHS

by

ADRIAN BOTEANU

A Dissertation
Submitted to the Faculty
of the

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

Degree of Doctor of Philosophy

in

Computer Science

August 2015

Approved as to style and content by:

Sonia Chernova, Major Advisor

Daniel Dougherty, Committee Member

Michael Littman, External Committee Member
Brown University, Computer Science Department

Candace Sidner, Committee Member

ABSTRACT

Similarity is a central cognitive mechanism for humans which enables a broad range of perceptual and abstraction processes, including recognizing and categorizing objects, drawing parallelism, and predicting outcomes. It has been studied computationally through models designed to replicate human judgment. The work presented in this dissertation leverages general purpose semantic networks to derive similarity measures in a problem-independent manner. We model both general and relational similarity using connectivity between concepts within semantic networks.

Our first contribution is to model general similarity using concept connectivity, which we use to partition vocabularies into topics without the need of document corpora. We apply this model to derive topics from unstructured dialog, specifically enabling an early literacy primer application to support parents in having better conversations with their young children, as they are using the primer together.

Second, we model relational similarity in proportional analogies. To do so, we derive relational parallelism by searching in semantic networks for similar path pairs that connect either side of this analogy statement. We then derive human readable explanations from the resulting similar path pair. We show that our model can answer broad-vocabulary analogy questions designed for human test takers with high confidence.

The third contribution is to enable symbolic plan repair in robot planning through object substitution. When a failure occurs due to unforeseen changes in the environment, such as missing objects, we enable the planning domain to be extended with a number of alternative objects such that the plan can be repaired and execution to continue. To evaluate this type of similarity, we use both general and relational similarity. We demonstrate that the task context is essential in establishing which objects are interchangeable.

ACKNOWLEDGMENTS

I would like to thank the professors that have advised me in ten years since I started higher education. First, I thank Ciprian Dobre and Valentin Cristea for conducting my bachelor's thesis at the Politehnica University of Bucharest (UPB), and for leading me on my first steps into research, which were instrumental in my decision to pursue a doctorate degree. I thank Adrian Petrescu, who was the first to entrust me with all the responsibilities of conducting research as a graduate student during the years of my master's degree at UPB. It was a privilege to be one of his students, and I dedicate this thesis to his memory. I cannot overstate the contribution that Sonia Chernova had to the completion of my doctorate and to my formation as a researcher. She encouraged and supported me to work on what I liked to work on, which I think is the best possible way to go through a PhD program. Over the course of this PhD, I received great advice from my thesis committee members, and also from Carolina Ruiz (WPI), Joseph Beck (WPI) and Kenneth Forbus (Northwestern University), advice for which I am grateful. Probably the hardest for me to concisely state, I want to thank my parents for giving me an upbringing that valued honesty, fairness, education, and knowledge, and for supporting me in countless ways throughout these years.

I was fortunate enough to be fully funded throughout my PhD, which allowed me to focus on this thesis. This work was partially supported by the National Science Foundation award number 1117584.

CONTENTS

1 Introduction **1**

1.1 Contributions and Chapter Outline 3

2 Background **5**

2.1 Semantic Networks and Language Resources 5

2.1.1 WordNet 6

2.1.2 Unified Verb Index 7

2.1.3 Research Cyc 8

2.1.4 ConceptNet 8

2.1.5 Discussion on Semantic Network Design 9

2.2 Context Representation 11

3 Topic Modeling **14**

3.1 Introduction 14

3.2 Related Work 18

3.2.1 Topic Modeling 19

3.3 Overview of System Architecture 21

3.4 Topic Modeling 22

3.4.1 Start Vocabulary Creation 23

3.4.2 Raw Topic Formation 24

3.4.3 Topic Refinement 25

3.4.4 Evaluation of Topic Quality 26

3.5 Generating Suggestions 29

3.5.1 Suggestion Quality 30

3.6 User Study on Suggestion Efficacy 32

3.7 Conclusion 36

4 Analogy Solving and Explaining **37**

4.1 Introduction 37

4.2 Related Work on Automated Analogical Reasoning 38

4.3 Semantic Similarity Engine . . . 40
4.3.1 Semantic Context Subgraph Extraction . . . 40
4.3.2 Sequence Similarity . . . 42
4.4 Modeling Analogies through SSE . . . 43
4.4.1 Answering Analogy Questions . . . 43
4.4.2 Explaining Analogies . . . 44
4.5 Evaluation Setup . . . 45
4.6 Conclusion . . . 50

5 Object Substitution in Robot Tasks **52**
5.1 Introduction . . . 52
5.2 Related Work . . . 53
5.2.1 Object Affordances . . . 54
5.2.2 Symbolic Plan Repair . . . 58
5.2.3 Learning from Demonstration . . . 59
5.3 Object Substitution within the Task Context . . . 60
5.3.1 Generating Candidates . . . 60
5.3.2 Deriving Context from the Task . . . 61
5.3.3 Concept Similarity . . . 61
5.4 Experimental Setup . . . 63
5.4.1 Labeling Valid Substitutions . . . 64
5.4.2 Performance Metrics and Classification Approach . . . 64
5.5 Substitution Prediction Results . . . 65
5.5.1 Evaluation of Context-creation Strategies . . . 65
5.5.2 Resource and Attribute Importance . . . 66
5.5.3 Context Vocabulary Sensitivity . . . 67
5.5.4 Generality of Substitution Models . . . 68
5.6 Inferring Substitution Characteristics . . . 70
5.7 Physical Robot Implementation . . . 72
5.8 Conclusion . . . 73

LIST OF FIGURES

2.1 A small example of nodes and relations in ConceptNet 9

3.1 Simulated in-story discussion suggestion via a dialog box. 15
3.2 System block diagram. 17
3.3 Annotated video frame from parent-child interaction during user study. 33

4.1 Overview of the Semantic Similarity Engine and its application to answering analogy questions and explaining the analogy relationship. 39
4.2 Example context surrounding *goose* and *flock*. The most meaningful sequece of relations is through an intermediate node, *bird*, and not the *Related To* edge which is directly connecting the nodes. 41
4.3 Unpruned (a) and pruned (b) context graphs. 41
4.4 Pie chart showing the proportion of start word pairs in the analogy dataset that have a geodesic distance of 1 (directly connected) through 7, or are unconnected. 42
4.5 Starting from HSSP, we select the common salient edge on each segment to produce human-readable explanations. *eq* and *ea* are then converted to English. . . . 44
4.6 Answer performance on four datasets, comparing the direct (D) and one-hop (1H) methods with the Divisi similarity baseline (S) both when using (top) and when ignoring (bottom) statistically derived relations. Questions are grouped by elementary grades (G1-G4), middle school grades (G5-G8), high school (G9-G12) grades and SAT. 46
4.7 Evaluation of explanation quality for direct (D) and one-hop paths (1H), comparing our system's output (C) with randomized (R) and edited (E) conditions. . . . 49

5.1 Screenshot showing the robot autonomously retrieving a roll of *tape* for the *pack schoolbag* task, after receiving user approval to use it instead of *glue*. 53
5.2 Portion of the *set table* task definition, which is one of the HTNs we used in our evaluation. 62

5.3 Relative classification performance of different classification methods. Following these results, we chose the Random Forest classifiers for our experiments, as it outperformed the others. 65
5.4 Relative classification performance for different methods of generating context vocabularies from the HTN and the task-agnostic baseline. 66
5.5 Mean and std. deviation of valid-only accuracy accuracy of full-context classifier for a varying percentage of words sampled from the context, compared to the 10-fold cross validation accuracy of the baseline. 68
5.6 Classification performance for single task learning, using 10-fold cross validation. 69
5.7 Classification performance when trained on all tasks but one and tested on the one left out. 70
5.8 Edge histograms for individual tasks, showing the differences in edge distribution between valid and invalid substitutions. 72

LIST OF TABLES

3.1 Example of the results after each processing stage. 22
3.2 Topic refinement results for increasing search depths. 27
3.3 Comparison between our approach and survey responses for refining raw topics. For the survey responses, the number of agreeing responses is shown as a fraction of the total responses for that particular question. If there is one, the predominant decision is in bold. 28
3.4 Crowdsourced evaluation of the qualitative features of the suggestions generated by our system. For each question, the table shows the proportion of Yes (or Agree) answers and the inter-user agreement. 32
3.5 Mean and standard deviation for vocabulary and dialog metrics for the three conditions. 35
3.6 Results of conducting a one-way Welch Anova test for statistical significance in vocabulary and dialog metrics between conditions. P-values in bold are significant with 95% confidence, the others show trends for a 90% confidence interval. . . . 35

4.1 Question examples from the SAT dataset and the answer results of our approach (correct answers shown in bold). ATT stands for Attempted, CORR stands for Correct. 46
4.2 Question examples from the SAT dataset and the answer results of our approach (correct answers in bold). 49

5.1 Vocabulary sizes for each task, the valid candidates counts are accoding to the expert annotation. 64
5.2 Classification performance for using similarity metrics from only WordNet, only ConceptNet, or both, for the full context classifier. Values reported for a random forest classifier over 10-fold cross-validation. 67
5.3 Rank scores for the similarity metrics. 67
5.4 Information ranker scores for attributes, including similarity metrics, task label and edge annotation. 73

5.5 Classification performance showing the changes in performance as attributes related to task and edge information are used or removed. 73

CHAPTER 1
INTRODUCTION

Perceiving similarity is one of the fundamental human cognitive mechanisms, part of the ensemble that enables learning. In visual perception, for example, similarity allows objects to be compared and recognized, leading to object permanence, a skill that is learned early in an infant's development [Baillargeon, 1991]. Before this transition, infants are not able to mentally model object occlusion or absence – if something is not within sight, it does not exist. Later on, children learn to classify instances into types by abstracting their characteristics. This important transition starts from recognizing and comparing specific objects or people to identifying the class an object belongs to (e.g. a chair as a type of furniture) [Baillargeon et al., 1985]. Categorizing objects relies on identifying key similarities that make a class of objects cohesive, which is of greater complexity than recognizing the same object. As the infant's cognitive abilities increase, so do the depth and complexity of evaluating similarity. Studies on human perception of similarity in adults have shown that it is an idiosyncratic process, difficult to model algorithmically [Goldstone and Son, 2005].

Several computational models of human similarity perception have been proposed. These models were derived from observational data aiming to capture the human behavior. An early yet still popular model, *geometric similarity* relies on representing concepts as n-dimensional space by decomposing them by multiple numeric attributes. This decomposition enables comparisons via a metric which measures distances between these points; commonly, this metric is the Euclidean distance. The choice of which dimensions are used to represent concepts is crucial. For example, one automatic method of selection is fitting pair-wise similarity evaluations into a reconstructed space [Shepard, 1962]. In this approach, human respondents rate the similarity of pairs of concepts. A similarity space is constructed algorithmically using these comparisons and the known characteristics of the concepts being compared.

Geometric similarity models assume a metric over the similarity space, which implies that the metric respects the triangle inequality. There exists empirical evidence that contradicts the geometric model, specifically its characteristics resulting from the metric assumption. Studies have shown that some entity pairs were recognized to be identical faster than others, implying that humans do not perceive identity uniformly across all concepts [Podgorny and Garner, 1979]. Despite these inaccuracies, geometric similarity remains an influential model. However, these limitations

prompted the development of the *featural similarity* model, which has been proposed as an alternative to geometric similarity in order to address the aforementioned shortcomings [Tversky, 1977]. In this model, concepts are represented as bags of features. Comparison criteria may take features into account differently, instead of assuming that concepts can be embedded into an Euclidean space. Both geometric and featural similarity assume that comparisons can be conducted in a non-hierachical manner, in which features are compared separately from others. A mismatch on a specific set of features does not affect how the rest of the features are evaluated. *Alignment-based* models were developed to model correspondence at a structural level. This contrasts geometric or featural models, which both represent concepts in a flat scheme and do not take into account the structure of complex comparisons. In order to strengthen similarity, features or properties need not only to match, but to also correspond to the same part of the larger structure. Through extension, not only physical attributes can be modeled in this fashion. Functional relations between parts of a system can represent comparison criteria themselves, as they are similar because they perform similarly [Gentner, 1983].

Regardless of their nature, in order to derive similarity judgments, background information is required to derive the features that the comparison will use. Each of these models assumes a certain conceptual representation that is directly mapped to the similarity space in which they operate. Whenever new information is to be incorporated into the system, it needs to be encoded in a suitable format, usually by a human annotator. Collecting and maintaining a large and coherent dataset is difficult. The problem is compounded by the fact that many of these similarity models do not implicitly allow for errors to be mitigated algorithmically or resolved interactively by the end user. In this work, we propose using general purpose semantic networks as a source of background knowledge. Semantic networks represent information as a graph in which nodes hold concepts and edges relations between the the concepts. Semantic networks have been used for a long time, not the least to represent semantics derived from natural language, but also spatial and affordance information. Since a semantic network allows for multiple edge types, it can develop and incorporate new types of information that the original structure was not provisioning for. As the data grows, either acquired from human input or automatically, semantic networks are susceptible to the same problems of inconsistency and noise. These characteristics mitigate some of the above issues and are the reasons for our choice of using semantic networks. We give an in-depth review of semantic networks in Section 2.1.

In this work, we propose a unified approach for using semantic networks in multiple types of similarity reasoning. The core idea is that semantic networks can be used to generate the frame of interpretation, or *context-graph*, for a similarity problem. There are at least 66 definitions of "context" across a variety of sciences, which leads to a large number of partially overlapping interpretations [Bazire and Brézillon, 2005]. Thus, we define a context-graph, or context, as the structured expansion of an unstructured context, using semantic information derived from a source that may be separate from the original source of the unstructured context. In our work,

the unstructured context is a vocabulary derived from the problem statement. For example, in the application we demonstrate in Chapter 3, the unstructured context is the vocabulary of words used in a discussion, and the context-graph that set of words linked via relations from ConceptNet. As an approach, context-graphs are common to all models presented in our work. Through context-graphs, we enable similarity reasoning that uses criteria different from what is available in the problem representation. This allows existing problems to be expanded and enriched by deriving context-graphs for the words contained in the problem. Relevant concepts that are not explicit in the problem representation may be used in generating solutions. Furthermore, providing data representation independent similarity algorithms enables greater flexibility with respect to which semantic network is used as a data source. At the same time, our approach allows for some of the noise present in the semantic network to be mitigated as part of each model, either built-in to the algorithm or through incorporating user feedback. We will now enumerate the contributions of this dissertation.

1.1 Contributions and Chapter Outline

- **Chapter 2: Background and Related Work.** This chapter covers work in the fields that this work relates to as a whole, summarizing relevant concepts, algorithms and methods. We use semantic networks as the source of background information in creating context-graphs. Thus, we review existing semantic networks in detail, taking into account their design and generation process. In relation to our proposed paradigm of context-graphs, we review methods of representing context in various computer science areas such as language processing and robot perception.

- **Chapter 3: Topic Modeling.** The first contribution addresses general similarity and its application in topic modeling. We show that topological distance and graph connectivity in a semantic network correspond to similarity as described by geometric similarity models. Closely grouped concepts in the network tend to be regarded as belonging to the same topic in human evaluation, irrespective of the type of relations that connect them. We demonstrate this model by applying it to deriving topics from dialog for an early-literacy primer. The development of early literacy skills has been critically linked to a child's later academic success. In particular, repeated studies have shown that reading aloud to children and providing opportunities for them to discuss the stories that they hear is of utmost importance to later academic success. CloudPrimer is a tablet-based interactive reading primer that aims to foster early literacy skills by supporting parents in shared reading with their children through user-targeted discussion topic suggestions. The tablet application records discussions between parents and children as they read a story and, in combination with a common sense knowledge base, leverages this information to produce suggestions. Because of the

unique challenges presented by our application, the suggestion generation method relies on a novel topic modeling method that is based on semantic graph topology. We conducted a user study in which we compared how delivering suggestions generated by our approach compares to expert-crafted suggestions. Our results show that our system can successfully improve engagement and parent-child reading practices in the absence of a literacy expert's tutoring [Boteanu and Chernova, 2013a].

- **Chapter 4: Analogy Solving and Explaining.** The second contribution addresses the opposite end of the similarity spectrum, which is relational similarity, studied in alignment-based theory. We model alignment-based analogy by evaluating relational parallelism along paths connecting respective word pairs in an A:B::C:D analogy. Using similar methods for extracting the context graph for either side of the analogy, we take a complementary approach by focusing only on paths and their constituent relations instead of generic graph connectivity. We use relational parallelism to simultaneously answer and explain proportional analogies, which are of the form *A is to B as C is to D*. By using context-graphs, we can identify relational parallelism between the two sides of analogy, and use this parallelism to generate human readable justifications of the answers. Our implementation, the Semantic Similarity Engine (SSE), related to alignment-based similarity models in that it assumes a one-to-one correspondence between the nodes in the similarity path pair. Our results show that SSE answers analogy questions with high confidence, a characteristic essential to enabling automated learning. Through surveys we show that human respondents agree with the explanations SSE generates.

- **Chapter 5: Object Substitution in Robot Tasks.** For the third contribution, we investigate similarity in a physical environment: substitutions for robot task execution. Robots executing plans in changing or new environments currently lack the flexibility to use novel objects in a context-aware manner. We present object substitution as a solution to repairing plans in open-world robotic applications. The key insight of our work is that considering the task context is important when performing a substitution. We relate the original, unavailable object, with a number of substitution candidates and evaluate their equivalence within the task context using a number of similarity metrics, including SSE analogy scores. Additionally, we further develop the explanatory capabilities of SSE to infer which relations are important for evaluating substitution within a task. In our evaluation we show that our approach models valid substitutions accurately and that the learned models are resilient to task variations. We further demonstrate the viability of our work by implementing and performing autonomous substitutions on a physical robot[1].

[1]We would like to thank David Kent for the contribution of implementing the tasks on the robot and creating the demonstration video.

CHAPTER 2
BACKGROUND

We focus on those aspects critical to understanding all contributions of this thesis, reviewing current semantic networks published as open resources for research (Section 2.1), and existing methods for representing context in artificial intelligence applications (Section 2.2). In addition to this summary, each contribution chapter (i.e. Chapters 3, 4, and 5) contains a review of works related to the topic discussed.

2.1 Semantic Networks and Language Resources

We use semantic networks throughout this thesis, either directly or via measures derived from semantic networks. Although commonly semantic networks are used to represent knowledge about language, as a representational structure semantic networks are not limited to encoding linguistic information. Other relational data, such as relative spatial positions of physical objects [Pronobis and Jensfelt, 2012, Kuipers and Byun, 1991] or inter-personal relations [Jung and Euzenat, 2007, Gloor et al., 2009] can be expressed using semantic representations. Because of the nature of the contribution this thesis brings, however, we will focus on linguistic semantic networks, and use the term to imply data related to language.

We will note as well that, while semantic networks are one of the most complex language resources, there are other types of resources available. Thus we will also review other language resources that either form the foundation of, or complement, semantic networks. These sources of language data can be grouped into the following categories [Navigli, 2009]:

- Structured
  - **Thesauri**, which represent relations between words from a linguistic point of view (synonyms, antonyms, etc.);
  - **Machine readable dictionaries**, which resemble printed dictionaries;
  - **Ontologies**, usually including a taxonomy and semantic relations;
- Unstructured

- **Corpora**, including raw corpora for resources solely comprised of text documents and sense-annotated corpora which provide word senses along with the texts;
- **Collocation resources**, providing statistics on word co-occurrence;
- **Lists of words**, among which are stop-words and domain labels.

Semantic networks build upon these resources. While similar to ontologies in their use a directed graph model, semantic networks have greater complexity and are generally larger. A semantic network represents concepts as vertices of a graph and relations between these concepts as edges in the graph. This general definition allows for a large variety of implementations with respect to the what type of information is encoded. Many applications have required the development of purpose-built semantic networks, specifically tailored to represent relations that are meaningful in the context of that application. For example, SenticNet [Cambria et al., 2010] was developed to assess if users express positive or negative feelings when writing comments or reviews online.

Other semantic networks have been created to represent broader variety of information. ConceptNet [Havasi et al., 2007] is one such example, which aims at representing commonsense information about the world: objects' properties, characteristics, uses, information about actions and verbs, and other data. Manually encoding such diverse information is not always practical or effective. In the case of ConceptNet, the data is corroborated from a variety of sources (see Section 2.1.4). Regardless of their nature, semantic networks can provide information from which we can derive a frame of interpretation, a process which depends on the types and variety of relations and concepts represented in the semantic network. It is therefore necessary to have access to a broad coverage database if general purpose similarity evaluations are to be attempted. We will now review four different semantic networks, all of which focus on different aspects of natural language.

2.1.1 WordNet

WordNet reflects a small number semantic relations between words, focusing on lexical relations. It uses synonym sets (synsets) to represent semantically equivalent concepts. Through its multitude of concepts, each synset designates a meaning of a word. For example, {*board, plank*} and {*board, committee*} represent two meanings of the word *board*. All words within a synset are interchangeable given an appropriate context because they are synonyms [Miller, 1995, Miller et al., 1990].

In addition to synonymy, other relations are represented in WordNet:

- **antonymy**, for example *good–bad*;
- **hyponymy**, for example *maple–tree*;

- **hypernymy** is the opposite of hyponymy (*tree–maple*);
- **meronymy**, for example *knob–door*, reflecting that a knob is part of a door;
- **morphological** relations, such as verb tenses.

As a result of the strong focus on lexical knowledge, WordNet does not represent verbs in detail. It also lacks expressive relations, such as the *UsedFor* or *Desires* relation types found in ConceptNet. These imply actions or requirements that are dynamic in nature and dependent on the context. However, the lack of such highly interpretable knowledge along with it being entirely contributed by experts makes WordNet a reliable resource.

2.1.2 Unified Verb Index

Of all parts of speech, verbs represent the hardest class to disambiguate [Chen and Palmer, 2009]. The Unified Verb Index (UVI) is an effort to merge the knowledge contained in previous collections of verb semantics [Kipper et al., 2008]: VerbNet[Schuler, 2005], PropBank[Kingsbury and Palmer, 2003], FrameNet [Baker et al., 1998], and OntoNotes [Hovy et al., 2006], containing over 8500 verbs in total. It fuses knowledge from these resources per verb, aiming to offer as much variety of information as possible. For example, the verb *to dance* has the following entries:

- Modes of being in motion (from VerbNet);
- Performance (from VerbNet);
- Waltz (from PropBank);
- A type of *self motion*;

We will provide further information for two of the data sources used in UVI:

- **VerbNet** is the largest available single online English verb lexicon. VerbNet was created to complement WordNet, as the latter mostly focuses on nouns; likewise, it follows a hierarchical organization. It provides a syntax example along with a demonstrating phrase for each sense (called frame) of the verb. For example, the verb *to destroy* has the following frames in VerbNet, in which NP stands for Noun Phrase, V for Verb and PP stands for preposition:
  - **NP V NP** : "The Romans destroyed the city."
  - **NP V NP PP.instrument** : "The builders destroyed the warehouse with explosives."
  - **NP.instrument V NP** : "The explosives destroyed the warehouse."

  We can observe that the senses are finely grained, each providing highly specific definitions and contexts under which the sense is to be used. In this example, "the Romans" are different than "the explosives" because the former refers to humans directly and the latter is a tool that was used, and this induces two different senses for the verb *to destroy*.

- **PropBank** stands for "proposition bank." It is a corpus of semantically annotated propositions focusing on verbs. It contains general textual descriptions of the context in which the sense is used. For example, the verb *to deal* has the proper meanings of *handling* and *dealing cards*, and the metaphorical sense of dealing with an issue. Phrasal verbs (*deal in*, *deal with*) are also included.

2.1.3 Research Cyc

The Cyc project was designed as a response to purpose-built semantic networks developed in the early 90s, which were too brittle despite their high performance [Lenat, 1995, Reed et al., 2002, Matuszek et al., 2006]. Similar to WordNet and UVI, Cyc is a human-contributed semantic network, developed however as a commercial product. Research Cyc is the version of Cyc licensed for academic use. For a hand-annotated semantic netowrk, Cyc has a relatively high number of terms (250,000), and contains numerous assertions (2.2 million).

Unlike the other semantic networks we review in this section, these assertions are expressed using a large number of predicates (i.e. edge types): 15,000. Although there are numerous predicates, a large proportion of them is not populated and the meaning of these terms is not axiomatized. This is the main reason for our choice of using ConceptNet over Cyc as the primary semantic network used in this thesis. Although ConceptNet does not provide an axiomatic edge description either, having a much smaller number of relation types results in more easily interpretable results.

2.1.4 ConceptNet

ConceptNet forms a large graph of concepts connected through relations, aggregating both authored and mined linguistic sources, including DBPedia, WordNet, ReVerb, VerbNet, the English Wikitionary, in addition to crowdsourced data [Havasi et al., 2007, Havasi et al., 2009, Speer and Havasi, 2013]. We chose to use ConceptNet over other semantic networks because of its breadth of concepts and relations. ConceptNet represents 48 relation types covering a broad spectrum while remaining interpretable by humans. For example, relation types include type hierarchies (*IsA*), properties (*HasProperty*), uses (*UsedFor*), abilities (*CapableOf*) and intents (*Desires*). Figure 2.1 shows a small portion from ConceptNet, exemplifying some of these relations connecting nodes related to the interactive application we use.

From the ConceptNet project we also use the Divisi2 toolkit [Speer et al., 2010], a sparse singular value decomposition matrix of the graph relations, to measure the pairwise similarity distance between words. The toolkit produces a square symmetrix matrix, in which each column and row corresponds to a concept index. The values in the matrix are real numbers in the $[-1, 1]$ interval indicating the similarity for the respective pair of row-column indices, where 1 is identical and -1 entirely dissimilar.

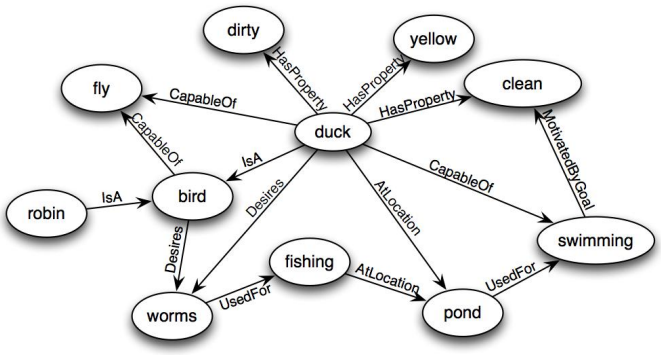


Figure 2.1: A small example of nodes and relations in ConceptNet

The distinguishing features of ConceptNet are its broad spectrum of data and relations; however, these come at the cost of precision. As opposed to a hand-crafted knowledge bases such as WordNet, ConceptNet contains inconsistencies such as different spellings of the same word (e.g. color and colour) or incorrect edges. Deeper evidence of noise can be found in the way nouns resembling verbs are handled: the concept *even* is linked to *day* by a *PartOf* relation, with the supporting phrase *evening is a part of the day*. This erroneous association results from the lemmatization process, in which the original word *evening* is lemmatized to *even*. To mitigate this noise, which would undermine a strictly greedy approach we incorporated the following design elements into the models presented in this work: (1) we separate topic modeling into two steps, which generate and refine topics using two different criteria (Divisi and connected components); (2) for solving analogies we search for wide similar relational paths, which contain multiple matching edges per segment; (3) we take into account the task context as a whole when evaluating object substitutions, which leads to a high resilience to changes in the task's vocabulary.

2.1.5 Discussion on Semantic Network Design

As the review above indicates, there is a broad variety of design goals that determine what a semantic networks should represent. Choosing which types of concepts and edges should be represented is an essential step that dictates the evolution and purpose of the resource. For example, WordNet is specifically designed to capture hyponym-hypernym and other lexical relations, which results in the tree-like structure of its synsets. Although it might share a significant proportion of its concepts with ConcetpNet or Research Cyc, the choice of relations produces a very different structure. Similarly, even though ConceptNet and Research Cyc both were created with the intention of representing commonsense knowledge about the world, the design of their sets of relations, as well as the data collection method, resulted in drastically different graphs. On the one hand, ConceptNet's restricted relation set allowed the crowd to contribute information, while the

highly-specific relations that Cyc represents require expert input.

We can interpret the choice of relation types as a partition of the concept space: the more specific and finely-grained relation types are, the smaller the set of concepts that they are applicable to becomes. Highly specific relations only would apply to a small subset of concepts, and these would make high-level processes such as explaining analogies have less cross-domain coverage. However, a more detailed relation set is likely to lead to stronger domain-specific performance. There exist numerous approaches that focus on semantics applied within a specific scientific domain, such as geosciences [Malik et al., 2007, Parekh et al., 2004] and molecular biology [Cho et al., 2007, Pesquita et al., 2009].

These design choices have a direct impact on the applications that a semantic network can be used for. Choosing a set of edges can be seen as taking a stance on how concepts should be represented and what type of information should be considered. In the case of ConceptNet, allowing for relatively imprecise relations restricts its immediate applicability in the real world, but gains a higher degree of interpretability when presenting information to humans. It would likely be impractical to reliably collect information from a broad range of resources, such as ConceptNet does, if the set of relations would be as diverse as Cyc's; as our crowdsourced surveys on topic quality and analogy explanations show, it is not uncommon to find little agreement between crowdsourced workers on difficult word problems. By relaxing the precision of representing information, ConceptNet allows for more diverse data sources to be integrated.

However, simplifying the relational space comes at a cost in precision. Specifically, we note two drawbacks: the loss of precise word senses, and the inclusion of information open to interpretation (or arguably incorrect). These drawbacks are distinct from the noise inherent to automatic relation extraction methods, and in our opinion would be present even in the case of a noise free network. In the case of explicitly representing word senses in the semantic network, there are a number of solutions representations. One approach is to implicitly use word senses by means of the other words presented (e.g. in the topic model, analogy explanation, object substitution suggestion): it is left up to the user to decide the word meaning and reject unlikely interpretations. Representing possibly conflicting information stems from the open-world approach to collecting data. An impartial resource should represent alternate points of view, even though it is detrimental to efficiently reaching decisions. Conflicts are much more common for contentious concepts, than for example for common household tasks or concepts. We would also like to point out that even if the semantic network would make clear decisions to represent on a single point of view, this would not be necessarily in agreement with the end-user.

Thus, we consider that the key aspects of choosing a semantic resource for the scope of our work are breadth and richness. This is because our approach focuses on producing results that are meaningful to and interpretable by the end user. Even though having finer granularity representations would lead to better domain-specific performance, it is more beneficial to allow the user to correct the system and close the feedback loop. While the performance of our methods would

likely be affected by using different semantic networks, it is important to observe that a given problem cannot be elucidated if the semantic network does not contain information relevant to it. In this case, the conservative nature of our methods is an advantage, preventing the system from producing results with insufficient support. For example, the proportion of analogy questions that SSE does not attempt to answer could be used as a measure of the semantic network's density for the question's domain.

While ConceptNet offers breadth and richness, it represents information only at the word (conceptual) level. This makes it a flexible resource that can be expanded and improved using heterogeneous data sources, but at the same time it limits how specific its information can be. Current semantic network designs attempt to improve performance by either explicitly representing word senses or by using more specific relation types. The main drawbacks of this approaches are that the more specific these representations become, the more limited they are. Highly specific representations make it particularly difficult to aggregate information from multiple sources, a problem related to ontology matching in relational databases [Euzenat et al., 2007, Doan et al., 2004, Otero-Cerdeira et al., 2015]. Instead of attempting to increase performance by using finer-grained relations, multi-modal representations might offer better solutions for physical-world applications, such as object substitution. Recent efforts in knowledge representation that combine language with visual information include databases for image recognition training such as COCO [Lin et al., 2014]. COCO contains labeled and segmented images of common objects as they are found in their common environments, totaling 80 object classes. These classes could be mapped to ConceptNet nodes to enable multi-modal representations. Multi-modal representations have also been used in video analysis [Chang et al., 2007].

In addition to enabling a robotic system to identify proposed substitutions in the environment, adding visual data to the nodes represented in the semantic network would also have the benefits of offering further information on what data is common in contemporary environments, and of allowing for estimates of the size and usual context of an object. While such information may be derived from text analysis or encoded manually, image information could be used as a direct source of evidence. Ideally, having a visual reference could enable embodied agents to refine these models by directly observing their surrounding world. These benefits would likely further improve the performance of object substitution, which at the moment does not distinguish constraints stemming from how common or likely available objects are (for example, when attempting to replace a pen with a quill).

2.2 Context Representation

Defining and using context has been identified as a core problem that needs to be addressed in artificial intelligence [Brézillon, 1999, Brézillon, 2014]. The term context generally implies that a program does not simply take information at face value in a rigid manner and instead allows for

variable behavior depending on the global data that is available. Context is defined with respect to a given application, standing for any data that is or should be available to the application for consideration. Examples of context range from a window in a graphical user interface to the knowledge available to an application [Brézillon, 1999].

In the field of artificial intelligence, the word context has been particularly used to describe an intelligent agent using information at its disposal holistically. The distinction is made between systems that use information directly in the attempt to satisfy a goal, and systems that only proceed towards the goal depending on the current state and (recent) history of state transitions. For example, if an intelligent collaborative agent has the goal of discussing a certain matter with a human, it will attempt to generate discourse that makes fluent transitions between discussion topics instead of directly prompting the user with a question that satisfies its goal [Rich and Sidner, 1998a]. In this example, the agent uses the context of the relationship between itself and the human to exhibit a certain behavior.

In addition to enabling agents to exhibit better and more predictable behavior, context information is powerful in disambiguating what the agent is perceiving. When directly applied to perception, context information can identify and trim the set of likely observations in applications such as object recognition in computer vision [Lee and Grauman, 2012]. Conversely, context can detect what object in a scene is the least expected i.e. what object does not fit in the image's context [Choi et al., 2012]. When interacting with humans in a reinforcement learning or learning from demonstration setting, agents need to correctly interpret the human's intention. This is not always straightforward, since the human may have assumptions that the robot needs to know and use correctly for the robot to learn the intended behavior [Aihe and Gonzalez, 2015].

The greater the ambiguity present in the learning domain, the more important the role context plays in disambiguating input that the agent receives. Natural language is inherently ambiguous, and it is in this domain that some of the currently most common context-aware approaches applied to intelligent agents originated.

The simplest representation of context for text data assumes that it is not necessary to encapsulate structure directly in the context representation itself. Therefore, the context can be represented as a set of words or features. The bag-of-words model is a popular model of representing a context in an unstructured manner. Initially used in information retrieval [Croft et al., 2010], bag-of-words models have been used in physical-domain applications such as computer vision [Bolovinou et al., 2013], scene recognition [Botterill et al., 2008], or robot navigation [Nicosevici and Garcia, 2012]. The main assumption of these models is that the order of the words in the document (placement of features in the image, etc) is not as important as the words themselves, therefore the order can be ignored.

However, there are limitations for using bag-of-words models as they do not represent word association in greater detail. One alternative is to use n-grams, which are word tuples of length n that frequently occur consecutively in the text. Because n-grams reflect more complex con-

cepts (n-grams can represent concepts that are described through multiple words, such as the bi-gram *washing machine*), they have been shown to outperform bag-of-words approaches [Wallach, 2006]. N-gram models have usually been limited to bi-grams or tri-grams for practical applications, and still represent a naive view of the relations between words in a sentence. In the case of structured data, such as XML, specific query methods have been developed, but these scenarios are far from the open world applications investigated in the present work because such structured descriptions are not readily available [Arvola et al., 2011].

Latent Semantic Analysis (LSA) is a well established method that creates unstructured word associations in which the words need not to be consecutive, instead relying on co-occurrence at the document level over a corpus [Landauer et al., 1998, Wiemer-Hastings et al., 2004, Dumais, 2004]. The result, known as topic models, offer a compact representation of documents in which full text is reduced to a collection to sequences of topics that also allows for efficient search in the form of Latent Semantic Indexing (LSI) [Deerwester et al., 1990, Hofmann, 1999]. While richer than bag-of-words models, topic representations do not take into account sentence structure more than at the level of n-grams.

Of the methods that represent text structurally, we will mention syntactic parse trees (SPT). Long standing work in natural language processing focuses on decomposing sentences into parse trees, which represent the structure of the sentence from a grammatical perspective. In SPTs, the leaf nodes are occupied by the sentences as they are encountered in the sentence, and on higher levels reside the corresponding syntactic units. There are numerous approaches to deriving SPTs from text [Abney, 1992, Gildea and Palmer, 2002]. One core application of SPTs is extracting relations from text [Fundel et al., 2007, Rusu et al., 2007], which produces data that may be used to populate semantic networks automatically. Difficult to parse sentences are often ambiguous, with recent methods leveraging external semantic information in conjunction with statistical knowledge to obtain a solution [Shi and Mihalcea, 2005].

The contributions presented in this work are founded on the related idea that connectivity in semantic networks can be used to enable similarity reasoning in a flexible and detailed manner. For this work, we represent context as a subgraph extracted from the semantic network, which includes the concept sets given as input. Therefore, the input to our context creation methods are bags-of-words, but the underlying working models are structured – the structure is derived from the semantic network.

CHAPTER 3
TOPIC MODELING

3.1 Introduction

Early literacy is a term used to describe the stage of literacy development that occurs before children are able to read and write. During this stage, important abilities develop, including vocabulary development, phonemic awareness and letter knowledge, all of which ultimately influence general cognitive skills. The development of early literacy skills through early experiences with books and stories has been critically linked to a child's reading and academic success. Repeated studies have shown that reading aloud to children and providing opportunities for them to discuss the stories that they hear is of utmost importance to later academic success [Wells, 1985, Bus et al., 1995, Burns et al., 1999, Duursma et al., 2008].

This area has been studied extensively from an educational standpoint, with theories such as dialogic reading suggesting that, in order for the child to develop good language skills, it is important for parents to engage with their child in focused conversation that is driven by shared reading [Arnold and Whitehurst, 1994]. Having enough exposure to language, both in terms of hearing words spoken by adults and learning new words, has been indicated as crucial for future academic success [Hilbert and Eis, 2014]. Small children go through a rapid learning period of acquiring language skills, during which it is particularly important for them to have verbal interactions with adults, especially their parents, from which they can learn. Therefore the learning process is partially conditioned by the parents' ability to steer these interactions toward learning goals [Whitehurst and Lonigan, 1998]. There exists evidence that parents receiving professional coaching on dialogic reading are more capable to lead joint reading sessions from which the child gains literacy skills [Pillinger and Wood, 2014]. However, such training may not be available for all families due to additional expenses and unawareness of its necessity.

To the best of our knowledge, this area has not received any contributions towards helping parents have better conversations with their children through algorithmic means. Existing adaptive tutoring systems, which track students' progress and provide customized feedback, have focused on teaching and verifying mathematical or scientific knowledge, with the aim to enhance and supplement classes taught in school [Feng et al., 2006, Weld et al., 2012, Wenger, 2014]. Such systems are designed for students that are in school and that already have a set of language and

reading skills. Instead, we target a younger age group, of children that either have yet to learn how to read and write, or are very early in the process of learning to do so.

The main goal of this work is to provide an interactive parent-child reading experience which would help parents talk more with their children. Our work leverages the fact that electronic books and tablet readers have become increasingly prevalent in recent years. In many cases, these devices seek to promote early literacy and increase child engagement by including animation and sound effects in the stories. Scientific evaluations of these technologies have found that, although engaging, such devices do not effectively achieve educational goals when used alone [Korat and Shamir, 2008]. Instead, recent studies highlight the importance of joint parent-child reading, showing that learning gains are achieved by combining the use of digital media with adult interaction [Segal-Drori et al., 2010]. Interactive stories tend to distract readers with multimedia features without increasing the dialog between parents and children during reading, which results in lower learning rates [Parish-Morris et al., 2013].

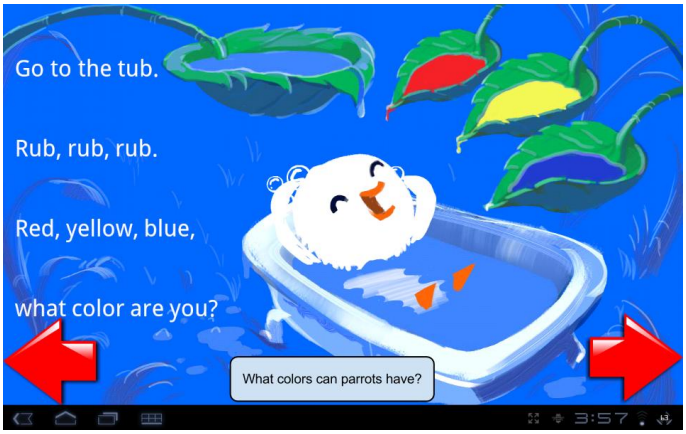


Figure 3.1: Simulated in-story discussion suggestion via a dialog box.

Taking these findings into account, we have developed CloudPrimer, an interactive reading primer to support parents in these discussions by offering suggestions for broadening the dialog and enriching the verbal interaction. The CloudPrimer application 1) records discussions of parent-child pairs engaged in a reading activity, 2) builds discussion topic models based on data gathered from across the community of readers, 3) generates English suggestion phrases using the topic models, and 4) delivers the suggestions at appropriate times during new interactions. The primer is based on an existing multimedia tablet application, in which a narrative is delivered through text, images, animations and sound, shown in Figure 3.1 [Chang and Breazeal, 2011]. The tablet application blends in simple tasks such as color mixing in order to provide learning opportunities for children. Our system delivers prompts to support parents in starting and conducting discussions that revolve around elements from the story. In this work, we introduce a

semi-supervised method for generating prompts without expert input. The traditional method for obtaining these prompts is for them to be authored by a literacy expert, which we consider to be the gold standard in evaluating our method.

Our approach consists of leveraging an initial community of readers to derive the topics of discussion parents and children approach while using the story. We then use these topics to generate suggestion prompts. We choose to derive topics from the dialog transcriptions instead of using only the text contained in the application because, compared to more traditional printed forms, interactive stories are rich in visual media and contain few words, both printed and spoken. Otherwise the potential discussion topic suggestions would be limited by the relatively rigid and simple story structure. Our main assumption is that, within this body of initial readers, some parents will have more developed and engaging discussions with their children, which our system can use to generate suggestions that will benefit future readers. It is important to note that our goal, to get parents to talk more with their children, is not equivalent to a dialogic reading strategy, since the suggestions generated through our method are formed by single statements that do not closely follow the development of the story.

Our goal is to add breadth to the interaction and to expand on what the story provides. Automatically generating suggestions for dialogic reading would require expert in-depth analysis and a more thorough strategy than what is possible to model automatically from noisy speech, as designing suggestions specifically for dialogic reading would imply not only a deep understanding of the characters and sequence of events in the story, but also of the readers' discussions from the training corpus. Furthermore, focusing on dialogic reading only would restrict the target age of our application for 3 to 5 year old children, because this method has been shown to be effective only for this interval.

We take a topic modeling approach instead of attempting to gain a deeper understanding of the discussions because it is unfeasible to automatically derive complex models for this use setting. The unstructured nature of parent-child dialog, the noise introduced in the audio recording from either ambient noise or from manipulating the table computer, together with the unreliable nature of transcribing speech from children, does not provide sufficient sentence integrity in order to build more complex models. Instead, we leverage commonsense knowledge to build topics that not only consist of word groupings, but also provide additional information about the relations between words in each topic.

Modeling dialog topics in informal discussion presents a particular set of challenges. In free form discussions, the goals are not agreed upon in advance. Since the participants do not announce detailed intentions on how they expect the interaction to evolve, abrupt changes of subject are frequent. In this relaxed form, expanding the discussion so that there is more parent-child discussion takes precedence over debating a well defined topic. The main drivers of the conversation are the participants' interests, their relationship and any external inputs, such as a book or movie that they might be discussing. The context and development of such interactions contrast with

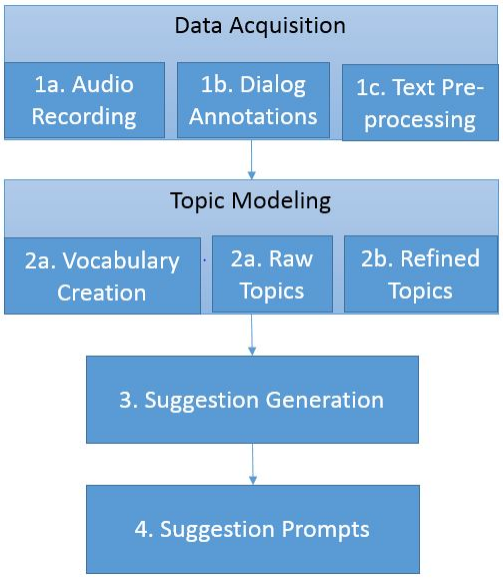


Figure 3.2: System block diagram.

formal settings, such as written articles and news broadcasts. The latter have a distinct, professional, approach in handling a subject and describing it. Speakers in this context try to meet the expectations of a large public who does not offer direct feedback, so stating opinions and facts accurately is important. This is accomplished through a crisp discourse which uses names and jargon to anchor the readers' or viewers' focus of attention. In contrast, a casual conversation uses improvised references that are mostly relevant only for the other participants [Linell, 1998].

To evaluate the relative impact of prompting readers with suggestions, we present an end-to-end user study conducted in a lab setting that evaluates the impact our suggestions have on parent-child dialog in comparison to two other conditions. This study was done in a controlled lab environment. Throughout data collection and evaluation we used the same story narrative and design. The results of the study indicate that the overall improvement in parent-child communication resulting from delivering suggestions generated by our method is comparable to exposing users to prompts created by child literacy experts.

We argue that our system offers an automated, semi-supervised method for generating suggestions. In order to build the models our approach requires and to generate suggestion prompts, separate dialog collection, annotation and processing is required for each new story. The core components, i.e. topic modeling and suggestion generation, are entirely unsupervised once the model parameters have been established. However, current automated speech transcription technology is not sufficiently robust to transcribe dialog. Likewise, the suggestions need to be filtered by the crowd for appropriateness, as the application is sensitive. However, such tasks can be automated and are arguably simpler to implement than contacting literacy experts for each story.

As the speech transcription technology matures, it may be feasible to replace the crowdsourced elements. Nonetheless, we maintain that the golden standard from a perspective solely focused on literacy benefits are the suggestions authored by experts. Our method represents an algorithmic alternative to this standard.

3.2 Related Work

Several intelligent systems which learn from users have been designed in numerous areas. Such solutions include automated office managers [Modi et al., 2005], personified assistants [Rich and Sidner, 1998b] and smart house applications [Bouchard et al., 2006]. These systems either interact directly with the user, through messages or an avatar, or can change their behavior without explicitly notifying the user. The common approach between all these numerous applications is to create a personalized model by observing an individual user. More broadly, recommender systems have been applied in a variety of domains, such restaurant recommendations [Boteanu and Chernova, 2013b], music suggestions [McFee et al., 2012], e-commerce [Schafer et al., 1999] or media websites [Bennett and Lanning, 2007]. The main types of recommender systems are using product ratings (collaborative filtering), the content of the reviews and item description, or hybrid methods [Ricci et al., 2011]. For these applications, the goal is to suggest an item, be it a movie or a restaurant, that the user is likely to enjoy given personal preference and how the item was reviewed by other users. Within this taxonomy of recommneder systems, we can be considered the discussion suggestion application of our topic modeling method to be a content-based recommender system, since it uses vocabularies collected from a large population of users to generate suggestions. The main difference between our work and recommender systems is that our system does not prompt suggestions based on user feedback, instead focusing on covering a variety of concepts that were discussed by previous users.

Starting from the words spoken by readers in the training corpus, we derive related words and relations using a separate semantic network. For this work we are using ConceptNet, a freely available commonsense knowledge base which we summarize in Chapter 2.

Significant work has been done in tracking topics in e-mail, news, scientific literature and meetings by using data mining methods and machine learning [Krause and Guestrin, 2006, Eisenstein and Barzilay, 2008]. Commonly, topics are modeled via latent semantic methods through analyzing word co-occurrence across large collections of documents [Deerwester et al., 1990]. Once topics are created, data mining methods are used to predict categories for new instances. The key difference between our topic modeling method and latent statistical approaches is that, because we are using a readily available semantic network, we do not require a large corpus. Instead, topics are constructed from a vocabulary using the semantic network. By doing so, we separate the process of relating words from the process of modeling the topics encountered in a single document. We argue that this reduces the training bias present in topics that are extracted

from a given corpus. This does not restrict our method to common words since semantic networks can incorporate information about entities (for example, ConceptNet has nodes about countries, geographical points, cities, etc). In addition, semantic networks can include information derived statistically from co-occurrence in documents, which in the case of ConceptNet is represented as *RelatedTo* edges.

The first stage of our algorithm for grouping words into topics also has some similarity with K-Means approaches for clustering words [Steinbach et al., 2000]. The raw topic algorithm can be viewed as clustering in a non-Euclidean space using Divisi similarity to measure distance between words. However, the key differences are the absence of a cluster center and having the same word belong to multiple clusters. Because of these factors, our approach of topic formation is not affected by the order in which words are processed.

3.2.1 Topic Modeling

Existing topic modeling groups words via statistical methods by observing word co-occurrence across large collections of documents represented as bag-of-words models [Deerwester et al., 1990]. Once topics are created, data mining methods are used to predict categories for new instances. The key difference between our topic modeling method and latent statistical approaches is that, because we are using a readily available semantic network, we do not require collecting a large corpus specific to the application. Instead, topics are constructed from a vocabulary. By using a semantic network, we separate the process of relating words from the process of modeling the topics encountered in a single document. We argue that this reduces the training bias present in topics that are extracted from a given corpus.

Now we will describe in more detail two commonly used statistical topic modeling methods. Latent Semantic Indexing (LSI) is an established method of associating words using their co-occurrence in documents, resulting in associations of related meanings within the context of the document [Deerwester et al., 1990]. The method was developed for mitigating problems stemming from word *synonymy* and *polysemy* in an information retrieval context. For this application, the task is to select the most relevant documents out of a very large collection using keywords provided by the user. The authors identify synonymy, polysemy, individual user word preference and only partial term overlap between documents as the main reasons for which words need to be associated statistically in order to enable effective search. LSI uses the singular value decomposition (SVD) of the word-document occurrence matrix to derive associations and to compactly represent documents as feature vectors. Another more recent approach, latent Dirichlet allocation (LDA), uses latent Bayesian models to evaluate word associations [Blei et al., 2003]. The context is similar to LSI i.e. a large collection of documents, therefore the resulting associations can be considered similar. Both LSI and LDA have been used extensively and there have been numerous variations developed from the initial approaches. For example, LSI has been extended

to probabilistic models [Hofmann, 1999], and Gibbs sampling has been used for LDA to increase computational performance [Porteous et al., 2008].

Statistical topic modeling methods require large corpora, which are not always available. One such application is that of grouping tags associated with other on-line content, such as user-submitted pictures, cooking recipes, or scientific articles. Along with the content, the users also submit a set of tags that describe the content so that it can be searched for; as a database-wide collection, these tag clouds are called folksonomies [Sinclair and Cardew-Hall, 2008]. Existing work has applied collaborative filtering methods to folksonomies to identify tag clusters (i.e. topics) [Simpson, 2008], enable information retrieval [Hotho et al., 2006a], identify trends in the content [Hotho et al., 2006b], and to make recommendations [Xu et al., 2008, Wang and Blei, 2011]. In addition to collaborative filtering methods, some approaches also incorporate some topic modeling method, for example when applied to social media messages [Zhao et al., 2011].

These methods produce what is commonly called *topic models*, which are sets of words in which words have possibly weighted membership. These topic models represent similarity from a generic standpoint. Although it can be assumed that it is likely that words in a topic model have compatible meanings, their relations are not explicitly modeled. These approaches are powerful methods of revealing words associations when assuming no previous knowledge about the data (or the words), but they do not offer better insight in why the words can be associated. However, language is immutable for practical time frames. Therefore, it is not necessary to recompute topic models for common words because the similarity relations between common words are static. This is because, except for words such as jargon or proper nouns, common words do not lose or gain new senses fast enough to require frequent re-evaluation. Since semantic networks offer pre-computed knowledge about words and their relations, they can be used to construct topics using only vocabularies instead of document collections. In addition, semantic networks can be constructed from a variety of sources and thus represent multiple of relations between concepts, unlike statistical topic models which only focus on whether words are part of the same topic or not.

Our algorithms for grouping words into topics also have some similarity with K-Means approaches for clustering words [Steinbach et al., 2000]. Both our raw topic formation and topic refinement techniques can be viewed as two clustering steps in a nonlinear space using two different distance measures (SVD distance and path length, respectively). However, the key differences are the absence of a cluster center and having the same word belong to multiple clusters. Because of these factors, our approach of topic formation is not affected by the order in which words are processed.

3.3 Overview of System Architecture

In this section, we give a high level overview of our system, which consists of the modular pipeline shown in Figure 3.2. Section 3.4 presents our two stage topic modeling approach. We then discuss multiple strategies for generating suggestions starting from topics in Section 3.5. In Table 3.1 we present a running example that illustrates how the data is processed at each step before being interactively delivered during reading sessions.

1. Dialog Data Acquisition: this step includes recording and annotating the interactions from which we generate suggestions. We collected 40 reading sessions from a preschool in Worcester, MA, and from the Boston Museum of Science (5 and 35 final usable reading sessions after discarding unusable ones, respectively). The data collected through the preschool was obtained by giving parents the tablets to take home. The data collected at the Boston Museum of Science was from readers (parent-child pairs) recruited during their visit at the museum. For the latter, the interaction took place on the museum premises. All collected data included a full audio recording of the interaction, which was recorded using the on-board tablet microphone. These recordings were annotated by a professional text transcription service with one annotator per reading session. Each annotation included all utterances that were recorded, along with the corresponding timestamp and presumed speaker (either parent, child or speech produced by the tablet application).

   Recording the sessions in a casual setting resulted, we believe, in more natural interactions, but at the expense of noise being present in the recording. In addition to background noise, the main source of distraction for the readers were interactions with other people not actively involved in using the tablets. Furthermore, the transcription process itself introduced noise such as misheard words or typing mistakes. A short example of this transcription is presented on the first row of Table 3.1; To prepare the transcription texts for topic modeling, we pre-processed them by removing all stop words and other very common English words. We then converted all words into lemmas form using the NLTK WordNet Lemmatizer library [Loper and Bird, 2002].

2. Topic Modeling: we model topics from the start vocabulary in a two step process using semantic networks:

   (a) Start Vocabulary Creation: in this step, we corroborate all words from all discussions and filter out words that occur only in one session in order to filter out unrepresentative words. We name this set the *start vocabulary*, because it is the input for our topic modeling approach.

   (b) Raw Topic Formation: in the first stage of topic modeling, we construct *raw topics* based on the start vocabulary using the Divisi2 module of ConceptNet [Speer et al.,

Algorithm Stage	Data
Dialog Data Collection	Here you go, you have a purple duck. What's your favorite color? Um, purple. So baby duck is hungry, eat two beetles. The beetles? Yeah, that's the beetles. More, more. Those are ladybugs.
Preprocessing	purple duck favorite color purple baby duck hungry eat beetle beetle yeah beetle ladybug want beetle need beetle need beetle want feed
Start Vocab. Creation	{purple, tap, duck, beetle, yeah}
Raw Topic Formation	{blue purple, green, yellow, different} {ant, firefly, cricket, ladybug, beetle} {owl, bird, duck} {need let want}
Topic Refinement	{blue, purple, green, yellow} {ant, firefly, cricket, ladybug, beetle} {owl, bird, duck} {need, want}
Suggestion Generation	What are green and yellow? An owl is a bird, what other birds do you know?

Table 3.1: Example of the results after each processing stage.

2010]. The output of this module presents the starting point for refining topics.

(c) Topic Refinement: the raw topics produced by the previous step are generated using approximate similarity. In this step, we *refine* these topics by directly exploring ConceptNet's graph structure. This process involves directly traversing the graph to identify the connected components within a topic's set of words.

3. Suggestion Generation: we use topics and edge-information to generate question phrases. By using topics, we reduce the exploration space in which word tuples are tested against the template defined for each question type. In addition to the end-to-end study, we evaluate the output of this module separately in Section 3.5.1;

4. Suggestion Prompts: we deliver suggestions during conversations and evaluate their impact via metrics such as the number of words spoken by the participants.

3.4 Topic Modeling

In this section, we present the topic modeling method used for this work. The main assumption of this method is that topological distance between concepts in a semantic network represents similarity. As mentioned before, because the semantic network can incorporate data produced from a variety of mechanisms (expert authored, statistically derived, crowdsourced information – as in the case with ConceptNet), our topic modeling method is only limited by what is represented in the semantic network. For this work, we use ConceptNet, which has a broader set of concepts and relations compared to, for example, WordNet. Our method works at the graph level of the semantic network, which implies that any type of concept (node in the graph) and any type of relation (edge in the graph) can be potentially used to form topics – all parts of speech and proper names, which

ConceptNet represents. One limitation of our approach is that it can not directly assign unknown words to topics; such words are first related to known concepts through an external method (expert knowledge or co-occurrence in documents, for example) and then added through appropriate edges to the semantic network, allowing statistical information to be combined with other data sources. The section is divided into the following parts: obtaining the start vocabulary, generating raw topics, refining them into the final form, and evaluating topics via two crowdsourced studies.

Due to the nature of the dialog form that we target, we use a different approach than statistical topic modeling such example LSI [Deerwester et al., 1990]. Another important factor in this decision is that defining co-occurrence in temporal data such as our transcription corpus is not as rigorous as per-document co-occurrence. Per-document co-occurrence would be unsuitable because the majority of the interaction is similar between reading sessions, since we are developing suggestions specific to a story. Instead, we aim to identify secondary topics of discussion that are introduced by readers in addition to the ones imposed by the story. The topic modeling method used in this work builds on our previous work, incorporating common sense reasoning in evaluating topics occurring in free-form the discussions [Boteanu and Chernova, 2013a].

3.4.1 Start Vocabulary Creation

The first step in grouping words by topic is to identify which words are the most relevant to the discussion. Written text and speech are very different in terms of phrasing, word selection and connectors. In particular, parents talking with their children have other goals beside transmitting a message, such as teaching new words [Hausendorf and Quasthoff, 1992]. We observed that parents often have to restate the goals and important concepts to keep their children on track. Another characteristic of dialog is that the density of topic-relevant words, such as nouns, verbs and adjectives, is low compared to a written document.

In prior work, [Boteanu and Chernova, 2013a] introduced an interest metric heuristic to determine which words may be of interest to the readers at any point during the session. This method was designed to select words from a very small number of reading sessions, potentially producing user-specific vocabularies from single reading sessions. In that context, the limited available data may not constitute sufficient basis to eliminate noise via collaborative filtering. We consider noise any word spoken in the background by non-participants or words that the participants use in discussions unrelated to the reading session. Since we collected a relatively large corpus of 45 discussions, it enabled the use of a collaborative filtering approach. We include into the start vocabulary all words that occur in at least two reading sessions except for stop words. We compared the resulting vocabulary with the union of all vocabularies resulted from applying the interest metric introduced by [Boteanu and Chernova, 2013a] and found no significant difference. Therefore we use collaborative filtering to create the start vocabulary for all results presented in this chapter.

We process transcriptions by separating text into words and removing stop words[1]. To account for typing errors in the annotation, we then perform spellchecking using a large US English dictionary[2]. In addition, we attempt to lemmatize words using the WordNet lemmatizer included in NLTK. If a lemma is unobtainable (i.e. the word is unknown to the lemmatizer), we add the word to the dictionary as outputted by the spellchecker. The start vocabulary corresponding to our training corpus of 45 reading sessions contains 1009 words.

3.4.2 Raw Topic Formation

In this section, we describe the first stage of grouping the vocabulary of interest into discussion topics. This step produces rougher topics that are later refined. An example of a set of raw topics is given in Table 3.1. We define a *topic* as a set of words relating to a common theme, without any particular order. Topics do not have names themselves and are defined only by the words that belong to them. This allows us to model each topic through the common sense relations between its constituent words. One key characteristic is that a word can belong to multiple topics at once. We allow this since words usually have multiple meanings.

Algorithm 1 Pseudocode for raw topic creation.

```
topics = φ
v = readVocabulary()
for each w in v do
    for each t in topics do
        similar = 0
        for tw in t do
            if similarity(w, tw) > similarityThreshold then
                similar = similar + 1
            end if
        end for
        if similar > size(t) * minSimilarityVote then
            t = t ∪ w
        end if
    end for
    if w was not added to any existing topic then
        topics = topics ∪ w
    end if
end for
```

The algorithm, shown in pseudo-code in Algorithm 1, starts constructing topics by removing a word from the vocabulary of interest and creating a new topic containing only that word. For all remaining words in the start vocabulary, we remove one word at a time and calculate the similarity distance between that word and all current topics using Divisi2 concept pair similarity. This function returns a similarity value in the real interval -1 (completely opposite) and 1 (identical). If the value is above a *similarity threshold* (i.e. the concepts are similar enough), the result counts as a positive vote that the word should belong to the topic. If a high enough *voting percentage* from

[1] Using the NLTK stopword list [Bird, 2006]

[2] We performed spellchecking using the *enchant* Python library [Perkins, 2010]

the words that are already part of the topic is positive, the new word is included in the topic. The process repeats itself until there are no more words in the vocabulary of interest. We can control the specificity of each topic by adjusting both the minimum similarity threshold as well as the minimum percentage of votes. For example, by using a high minimum similarity (0.5) but a low voting percentage (50%) loose topics are generated, such as the following:

- ant, owl, bird, ladybug, duck;
- noise, tap;
- color, blue, purple, green, yellow;
- need, say, let, want;
- ladybug, bird, beetle;
- happen, let, pass;
- cricket, bird;
- purple, different, green;
- yummy, hungry;
- firefly, ladybug ;
- push, angry.

Using a lower similarity threshold (0.3) but a higher minimum voting percentage (85%) produces tighter topics, mostly because a newly introduced word has to have some association with most other words present in the topics, such as the following example:

- owl, bird, duck;
- push, tap;
- blue, purple, different, green, yellow;
- ant, firefly, cricket, ladybug, beetle;
- push, say, let, pass;
- need, let, want;
- push, happen, let, pass;
- yummy, hungry.

Setting both thresholds high produces raw topics that are very conservative and contain very few words. We do not include these results for the sake of brevity, but we do not consider such results practical. Similarly, setting both thresholds low produces looser and noisier topics. Since the goal of the raw topic stage is to restrict the search space of the refinement algorithm, and not to provide high quality topics itself, choosing either extreme is detrimental to the final result.

3.4.3 Topic Refinement

Since it is based on similarity measure derived from the graph's SVD, the topic generation method we introduced in the previous section produces imprecise results, in which words may be erroneously associated into the same topic. However, it has the advantage of speed over directly ex-

ploring the highly connected ConceptNet graph, segmenting the initially very large search space into smaller regions. Directly searching for connected components is unfeasible not because identifying connected components in a graph is a hard problem, but because semantic networks are very large and linear time algorithms are not sufficient. In ConceptNet, nodes with a degree of 30 or higher are common. The large size of data makes retrieval time significant as well. The key idea of refining topics is to find connected components of the subgraph represented by the raw topic in ConceptNet. Starting from raw topics makes this problem tractable by reducing the number of concept pairs that need to be tested. In this section we introduce a method of refining those results by directly exploring the graph structure of ConceptNet.

We consider two concepts to belong to the same topic after refinement if there is a path between the two respective connected components of at most the length of the search depth. For example, in Figure 2.1, concepts "robin" and "worms" have no direct edge connecting them, but are both connected to "bird" by "Is A" and "Desires" relations, respectively. A search with the depth of 1 will separate them into different topics, while a search depth of 2 will group them into the same topic. Algorithm 2 shows in pseudocode for refining topics.

This approach eliminates spurious associations introduced in the raw topic formation step by efficiently searching for connected components within groups of similar words, which limits the exploration space. In our evaluation we found that the most practical search depth is 2 since it best matched the words selected by crowdsourced workers (Table 3.2). We show an example of the effect of search depth on refining a small topic in Table 3.2. All types of relations are taken into account for these topic refinement results.

Algorithm 2 Pseudocode for refining topics. t is the topic that is being refined, p is the resulting set of topics after separating the words from t and d is the search depth.

```
p = φ
for w in t do
    candidates ← nearest_neighbors(w, d)
    split ← True
    for q in p do
        if q ∩ candidates = φ then
            q ← q ∪ {w}
            split ← False
        end if
        if split = True then
            p = p ∪ {w}
        end if
    end for
end for
```

3.4.4 Evaluation of Topic Quality

We conducted two surveys to evaluate our topic refinement algorithm. To obtain the topics used in both surveys, we constructed topics from the 45 reading session transcriptions from the training

Raw topic	{**deer wing frog duck**}	{**owl bird duck**}	{**push happen let pass**}
Depth = 1	{deer}{wing} {frog}{duck}	{owl}{bird} {duck}	{push}{pass} {happen}{let}
Depth = 2	{deer}{wing}{frog duck}	{owl}{bird duck}	{push}{pass} {happen}{let}
Depth = 3	{deer}{wing}{frog duck}	{bird duck} {owl}	{push}{pass} {happen let}
Depth = 4	{wing duck}{frog duck} {deer}	{owl bird duck}	{happen let pass} {push}

Table 3.2: Topic refinement results for increasing search depths.

corpus and then sampled uniformly from the resulting set of topics. The topics presented to workers were un-refined topics, and the crowd's choices were compared against the selections made by our topic refinement algorithm. For both evaluations, we crowdsourced workers through the Crowdflower platform. We used the default task distributions options (tasks are also relayed to other platforms such as Amazon Mechanical Turk), but we selected only workers from countries with English as the majority language.

The first survey required participants to read a list of words and select a subset that forms a common topic through check boxes corresponding to each word selection. For example, when presented with the set of words {*brown, old, long, hello, thing, green, yellow, okay, yes, whole, white, red,*} a possible response would be to check {*brown, green, yellow, white, red*}. In total 12 such topics were evaluated through surveys and the results were compared with the topic refinement algorithm output for search depths of 1, 2 and 3. Each topic received 10 evaluations, for a total of 120 responses. To compute the inter-worker agreement that a word from the list was part of the topic, we used the proportion answers that marked that word. The mean inter-worker agreement value for all tasks was 19.25%, with a total of 17 participants in the survey.

To obtain topic selections from the agreement values, we binarized these values per word via clustering (fitting two clusters using the k-means algorithm or a bimodal Gaussian Mixture Model produced identical results), and selected the cluster corresponding to the majority of selections as the final topic. For example, selection answer agreement values for the raw topic *brown, thing, green, orange, white, whole, hello, red* were *18%, 3.6%, 18%, 18%, 18%, 3.6%, 3.6%, 18%*, respectively; for these selection values the cluster assignments were *1, 0, 1, 1, 1, 0, 0, 1*. Cluster 1 corresponds to a higher mean selection, thus the refined topic selected by the crowd is *brown, green, orange, white, red*.

We then computed the agreement between the topic refined by the crowd with the output of our algorithm at different search depths as the proportion between the number of selection matches per word and the size of the unrefined topic. The results of this comparison were not conclusive: the average agreement between our algorithm and the crowd was 58%, 52%, and 56% for respective refinement depths of 1, 2 and 3, with values distributed uniformly in a wide interval, from 25% to 89%. This result, together with the low inter-worker agreement of 19%, showed that our results

Raw Topic	**Outliers - Topic Refinement**	**Outliers - Survey Responses**
Blue, Purple, Different, Green, Yellow	Depth 2, 3: Different	**Different (13/15)** Purple(1/15) $None$(1/15)
Ant, Firefly, Cricket, Ladybug, Beetle	Depth 2: Cricket Depth 3: $None$	Ant(2/15) Firefly(1/15) Cricket(2/15) Beetle(1/15) ***None (9/15)***
Push, Say, Let, Pass	Depth 2, 3: Push, Say, Let, Pass (No common topic found)	Push(1/15) **Say(8/15)** Let(1/15) $None$(2/15)
Need, Want, Let	Depth 2, 3: Let	Need(1/15) Let(7/15) $None$(7/15)
Deer, Wing, Frog, Duck	Depth 2, 3: Deer, Duck	**Wing(12/13)** Duck(1/13)

Table 3.3: Comparison between our approach and survey responses for refining raw topics. For the survey responses, the number of agreeing responses is shown as a fraction of the total responses for that particular question. If there is one, the predominant decision is in bold.

are not orthogonal to the crowd's judgment. However, it is difficult to draw conclusions because of the high degree of variability between individual answers.

As a result, we designed a second survey, which consisted of a set of "odd word out" problems in which respondents were given a short list of words and instructed to select the one that did not fit with the others. We produced these lists by refining raw topics using our method, and then adding a word that was excluded by the refinement process back to the topic. The option "None" was also available in the case the workers considered that all words were similar. In total there were 26 topics. On average, each topic received 12 different evaluations, with 312 judgments in total.

The average inter-worker agreement was 73.5%, calculated per task as the percentage of judgments that the most selected option had, out of the total number of response per task – 12 on average. The survey answers are divided into two groups by agreement, high and low. For the nine topics which contained mostly verbs, the agreement ranged between 30% to 60% with an average of 45.4%. For the rest of 17 topics, mostly formed by nouns, the agreement ranged between 75% and 100%, with a mean of 88.4%. This high agreement group of topics contains words that are interpreted similarly by the reviewers. In contrast, topics with low agreement contain words with multiple meanings, thus subjective to evaluate. Using a search depth of 2, the topic refinement algorithm matched the dominant decision of the survey answers for 47% of the topics – it either eliminated the same word or kept the topic unchanged. For a search depth of 3, the percentage is 29%. Table 3.3 presents a few examples of the survey questions, showing the raw topics, the words selected as outliers by the topic refinement algorithm, and words selected as outliers by the survey participants.

Based on these results, we can conclude that our system best matches human respondents when analyzing topics composed of nouns. An exception to this is the example in the last row of Table 3.3, in which the algorithm was unable to differentiate animals (deer, frog, duck) from a limb (wing). We attribute such errors to currently missing edges in the constantly expanding

commonsense knowledge network.

The most significant disagreement, both between our system and the respondents, and between the respondents themselves, occurs on topics consisting of verbs, such as in lines 3 and 4 of Table 3.3. These collections are more difficult to interpret, and refining the topic would imply adopting a specific angle. For example, on line 3 of the table, the consensus in selecting 'Say' might be that it is the only action that produces speech, but similar classifications can be found to eliminate other words.

This evaluation shows that, for the situations in which human respondents reach consensus on the constituency of a topic, our approach successfully matches that consensus.

3.5 Generating Suggestions

In this section we describe the method we introduce to generate prompts for our end-to-end user study. The goal of these prompts is to enhance and foster the conversation parents have with their children via questions and other suggestions. Therefore, we are not interested whether the child is able to correctly answer the questions and do not model or provide any input method for answers. However, given the relatively low complexity of these questions, we assume the parent is capable of understanding and interpreting the prompts, rephrasing them to fit the conversation with the child instead of directly reading them out loud.

As mentioned before, since our corpus contained sufficient data, totaling 45 reading sessions, we used collaborative filtering to obtain the start vocabulary, selecting all words that occurred in at least two separate reading sessions. Using all discussion transcriptions recorded from participants, we created a vocabulary of 1009 containing all spoken words that occurred in more than one discussion. We then created topic models starting from this vocabulary, producing a single set of topics for the entire narrative. Keeping a unified vocabulary allowed us to have the richest connectivity between words, while at the same time enabling topics to cross between pages. For each topic, we employ a number of methods in order to elicit different types of discussion.

We designed these heuristics such that they would use a broader range of commonsense knowledge that is present in ConceptNet. Although not specifically designed for literacy education, the language and world knowledge present in ConceptNet is arguably relevant and related to expanding on the information in the story. While some of this information may be initially considered too abstract for a child to assimilate directly, it is the parent that is the target of our suggestion system. Thus, a suggestion that may seem initially very abstract (e.g. "Why do balls roll?") can be adapted to a discussion about round objects, even exemplified with other objects such as pens, instead of directing the conversation to a topic on solid mechanics. As described in the following list, method 1 targets general similarity evaluation, methods 2a and 2b focus on evaluating type classifications, and methods 3a, 3b, 3c and 3d target reasoning about various characteristics of an object. We provide examples of suggestions generated using the different strategies:

1. We randomly select two words from the topic and ask the readers to either come up with other related words or identify what the words have in common – "What other things are like *minutes* and *years*?"

2. We generate questions based on hypernym-hyponym relations in two ways:

   (a) Within a topic, we find words that have the same super-class and ask what do the words have in common. For example, for the words "duck" and "swan," by asking "What do ducks and swans have in common?" we would expect and answer similar to "They are both birds."

   (b) If within the topic there is a hypernymy relation between a pair of words, we ask a complementary question:
   "A duck is a bird, what other birds do you know?"

3. We test all possible pairs of words in the topic if they are connected by ConceptNet edges expressing properties or capabilities, and ask a question that tests knowledge about that fact. For example, "Why do birds fly?" for the concepts "bird" and "fly" connected by the edge *Capable Of*. We apply similar patterns for the following edge types:

   (a) *Capable Of* – "Why do *balls roll*?"

   (b) *Made Of* – "Why is a *towel* made of *cotton*?"

   (c) *Part Of* – "Why does a *plant* have a *leaf*?"

   (d) *Has Property* – "Can a *friend* be *important*?"

Note that for the edge-based approaches (methods 2 and 3a,b,c), starting from topics reduces the search for possible pairs from the size of the entire vocabulary used throughout the session (hundreds of words) to a much smaller set (6 words per topic, on average).

In order to transpose the graph representation to a human readable form, we use a number of publicly available language libraries and hand-coded patterns specific to each type of question. These libraries include *pylinkgrammar* (the Python implementation of Link Grammar [Sleator and Temperley, 1995]) for checking grammatical correctness of the final result, and *pyinflect* for converting nouns to singular or plural forms. Patterns include fixed translations of edge types to human-readable forms. Finally, we filtered results using a number of hard-coded lists of words, so that no inappropriate words would be included in the final suggestion list. These included any references to generally offensive words, words about religion, age, and gender.

3.5.1 Suggestion Quality

As with our other methods presented in the previous sections, we conducted a crowdsourced evaluation on the final output of the question generation module. In doing so, our focus was twofold:

for the suggestions to be usable in real-world scenarios, the suggestions need to be both grammatically correct and interesting to the readers. Since the quality of applications available for tablets is generally high, our users would quickly start ignoring our suggestions if they were not engaging. Therefore, we used the results of this survey to further filter out unsuitable suggestions.

We conducted the evaluation on the Crowdflower crowdsourcing market. Each worker was presented with a description of the purpose of the task, in which they were informed about the parent-child reading setting and the ultimately educational goals of the project. We then asked workers to evaluate individual suggestions through a list of nine agree/disagree questions. Each task was generated in the following template: *"The question [SUGGESTION] [SURVEY QUERY]"*, in which *[SUGGESTION]* is replaced by one suggestion generated via our method and *[SURVEY QUERY]* is the list of statements shown on the second column of Table 3.4. For each question, the users were asked binary questions i.e. if they agree and disagree with the statement with respect to the suggestion.

In total 294 suggestions were evaluated via 1470 judgments, where each judgment consisted of answering all ten evaluation questions. The average inter-user agreement per question was high at 70%, measured as the percentage of votes in favor of the majority choice. This shows that the tasks were easy to understand and unambiguous. We now discuss what can be deduced from the proportion of users agreeing or disagreeing with individual statements.

The high "Agree" proportion from statements 4, 5, and 6 shows that understanding and answering the questions would require children to learn something new. The results for statements 7, 8 and 9 show that our question target approximately equal proportions between abstract and concrete concepts. We consider this to be a desirable feature, since it may result in a varied level of difficulty. Obtaining perfect grammatical correctness was not one of our primary goals, yet two thirds of the prompts are considered to be so. We find that the responses for statements 1 and 2 to be more difficult to interpret. We observe that only 0.03 of questions were considered offensive in a general setting and not because they were addressed to a child. The degree in which a question is appropriate or not to ask to a child is a personal choice for every parent. While we did not experience any adverse reactions during the end-to-end user study, future deployments could provide the parent with a full list of suggestions in order to eliminate any personal concern.

The responses for statement 10 would indicate that our suggestions are too difficult to incorporate in a discussion with a child, which contradicts the outcome of other responses, especially statement 5. For example, the suggestion "Why does a bird have wing[s]?" received 100% positive responses from the crowd on questions 5 (i.e. it is worth asking to a child), 80% positive responses on question 6 (i.e. adults would consider it interesting), 80% negative responses for question 2 (i.e. it is appropriate to ask), but at the same time 60% agreement for question 10 (i.e. it is too difficult for a child). One possible interpretation of this result is that the respondents answered the question assuming that a child would try to understand the suggestion on his own, which is not the goal of our work – the reading and learning process is primarily directed by the parent,

Statement Number	Statement Text	"Agree" Answer Proportion	Inter-user agreement
1	is inappropriate/offensive to anyone	0.34	0.67
2	is inappropriate only because of the setting, since it is addressed to a child	0.31	0.69
3	is grammatically correct	0.67	0.7
4	makes sense logically	0.72	0.71
5	is worth asking because it is not something necessarily obvious to a four year old child	0.65	0.71
6	would be considered interesting by most adults	0.38	0.68
7	references only concrete objects	0.51	0.69
8	requires understanding an abstract concept	0.62	0.71
9	uses words in their primary meaning	0.7	0.72
10	references concepts outside of the reach of a small child (e.g. theory of relativity)	0.71	0.73

Table 3.4: Crowdsourced evaluation of the qualitative features of the suggestions generated by our system. For each question, the table shows the proportion of Yes (or Agree) answers and the inter-user agreement.

the primer is there to only provide support to the parent. Therefore, because the suggestions are directed toward the parent, they need not to be within the grasp of a child because the parent can explain the concepts that the child is missing, thus achieving better educational impact. The above example could for instance be rephrased by the parent to explain to the child that wings are used for flying, and both birds and airplanes have wings.

For the final selection of suggestions for the end-to-end user study, we conservatively selected out of the pool of 294 suggestions only those for which the responses had "Agree" answers with high confidence (over .80) for statement 5, resulting in a total of 186 suggestions.

3.6 User Study on Suggestion Efficacy

For the final evaluation of our work, we conducted a user study to assess the impact our suggestions have on the interaction. The main goal of this study was to verify whether suggestions generated automatically are an effective method of eliciting verbal interaction between parents and children during reading sessions. The study involved 4-8 year old children ($n = 88, M = 5.64, \sigma = 1.33$) and their parents, recruited from the greater Boston area. The study took place in controlled lab settings. Figure 3.3 shows the study setting. During the study, the tablet was instrumented to record verbal utterances, which were then annotated professionally in the same fashion as the training corpus.

We evaluated one non-promting condition (NONE) and two prompting conditions (CROWD and EXPERT). The prompting conditions add a new element to the user interface in order to deliver the prompts: an animated butterfly character which would appear on-screen on corresponding pages. In order to avoid inconvenience to the users, the application required tapping on the butterfly in order to start speaking the prompts. Participants in the NONE condition included those

who did not receive suggestion prompts, as well as those for which the butterfly character was never selected. Participants in the CROWD condition viewed suggestions generated by the system described in this chapter. And participants in the EXPERT condition were shown samples from a pool of 28 suggestions manually created by a literacy expert, which included:

1. Describe how Baby D (the protagonist) is feeling right now.
2. What color can Baby D be?
3. What are you thinking right now?
4. What will happen next?
5. What can ducks do?

In some cases, users participating in a prompting condition did not tap on the butterfly at all on the course of the reading session and therefore were not exposed to any prompt. Consequently, we included these users in the NONE condition, which is why this condition has more subjects than the prompting conditions. We took this decision because our focus was not to investigate the effectiveness of the prompting method, and subjects that did not tap on the prompting character were effectively unexposed to any suggestion. Choosing an optimal prompting method is outside the scope of this work, but this behavior indicates it may be worthwhile to investigate. Therefore, the final distribution of subjects per condition was the following: NONE – 43, EXPERT – 22, CROWD – 20, totaling 85 subjects.

We tested the following hypotheses:

1. **Vocabulary**: We hypothesize that participants exposed to discussion suggestions will talk more with their children, using a greater number of words and with greater variety during the reading session;

Figure 3.3: Annotated video frame from parent-child interaction during user study.

2. **Dialog**: We hypothesize that participants exposed to discussion suggestions will engage more in conversation, as measured by the number of turn-taking exchanges and number of questions;

3. **Suggestion Automation:** We hypothesize that semantically-generated topic suggestions will result in effects comparable to expert-generated suggestions, as measured by the aforementioned vocabulary and dialog metrics.

We applied the following set of vocabulary measurements on these transcriptions in order to characterize the verbal interaction:

1. **Full Utterances:** the number of all complete utterances, including phrases, sentences or distinct words, normalized across session length. For this metric, complete sentences count as one utterance, the same as single words that were not part of a sentence;

2. **Single Words:** the count of every word spoken during the interaction. In conjunction with the full utterances metric, this metric can indicate the complexity of the sentences that were used – the greater the *single words* metric is compared to the *full utterances*, the longer the spoken sentences were;

3. **Novel Words:** words not directly included in the story or the suggestions, indicating the richness of the vocabulary introduced by parents;

4. **Lexical Diversity:** measured as the ratio between the number of unique words and total number of spoken words.

One of our goals is for the interaction to incorporate more dialog between the parent and child as they are reading the story. We used a second set of metrics to evaluate the level of dialog present in the interactions using two metrics:

1. **Turn-Taking Exchanges:** a measure of the number of conversational exchanges, normalized across session length, between the parent and the child, indicating a two-way conversation rather than just one participant speaking. The number was approximated from the number of transitions in the text transcripts;

2. **Question Utterances:** a count of the number of questions the parent asks the child during the interaction, approximated by the number of question marks present in the text transcript, normalized across session length.

Table 3.5 shows statistical results of tracking these metrics across the three conditions i.e. NONE, EXPERT and CROWD. We can notice that for most metrics, both suggestion methods outperform the baseline. While the EXPERT condition produced more noticeable results on the *single words* and *novel words* metrics, subjects exposed to prompting conditions spoke more words

Metric	NONE	EXPERT	CROWD
Full Utterances	0.245 ± 0.109	0.309 ± 0.077	0.305 ± 0.101
Single Words	1.088 ± 0.463	1.317 ± 0.316	1.312 ± 0.402
Novel Words	0.325 ± 0.154	0.458 ± 0.160	0.431 ± 0.230
Lexical Diversity	0.424 ± 0.117	0.352 ± 0.052	0.400 ± 0.110
Turn-taking exchanges	0.255 ± 0.093	0.357 ± 0.075	0.304 ± 0.075
Question utterances	0.029 ± 0.014	0.034 ± 0.014	0.039 ± 0.019

Table 3.5: Mean and standard deviation for vocabulary and dialog metrics for the three conditions.

Metric	NONE – EXPERT	NONE – CROWD	EXPERT – CROWD
Full Utterances	**0.009**	**0.04**	–
Single Words	**0.024**	*0.058*	–
Novel Words	**0.003**	*0.074*	–
Lexical Diversity	**0.002**	–	–
Turn-taking exchanges	**0**	**0.035**	**0.027**
Question utterances	**0.004**	**0.034**	–

Table 3.6: Results of conducting a one-way Welch Anova test for statistical significance in vocabulary and dialog metrics between conditions. P-values in bold are significant with 95% confidence, the others show trends for a 90% confidence interval.

and also introduced more words in the discussion. The NONE condition exhibits higher lexical diversity values than both prompting conditions. We believe this is the result of the prompting conditions focusing the discussion on the story, which is beneficial since otherwise the readers may become distracted and lose attention to conversation topic. Regarding the metrics measuring dialog, also shown in Table 3.5, we observe that both prompting conditions elicited verbal interaction significantly. By having both a higher number of questions and more dialog exchanges present in the interactions, the literacy benefits of the reading session are increased.

Table 3.6 shows the results of performing a ANOVA test for significance on each metric. The suggestions authored by literacy experts resulted in more significantly different values for all metrics, improving the overall interactivity and quality of the dialog. Compared to this gold standard, our method produced less pronounced results. Nonetheless, both dialog-related metrics indicate significant improvement over the baseline, being comparable to the expert condition. Our method showed significant improvement (95% confidence) in the vocabulary metrics only for the *full utterances* metric, but it is important to note that two other metrics indicate a trend of improvement (90% confidence).

The results of our study reinforce existing results in the literature that the interactivity and richness of joint parent-child reading can be improved through suggestion prompts. Both prompting conditions showed significant improvement over the non-prompting baseline condition. We consider that these results indicate that our method can offer a suitable alternative to expert-authored suggestions, and be particularly useful in domains where the input of a literacy expert is difficult to acquire, such as in applications with dynamic content. This work also provides the future potential of customizing suggestions for individual users and for specific literacy goals.

3.7 Conclusion

The system described and demonstrated in this chapter is designed to stimulate dialog between parents and children by providing suggestions to the parent. It comes as an alternative to guidance offered by child literacy experts, by enabling multimedia applications to support the parent in engaging verbally more with their child. Our contribution supplements a tablet application with the capability to offer intelligent feedback to its users, making the learning experience more effective for pre-literacy children while retaining a game-like approach.

Our approach is independent of the application's content, making it suitable for integration with any other story-driven educational application. The methods we introduce are partly-unsupervised and independent of the content of the discussion. Instead of using word associations derived directly from the observed interactions, we use pre-existing semantic networks to infer topics and generate suggestions. By doing so, we are able to model topics from sparse transcriptions of unstructured noisy dialog.

Using a novel method, we generated topics and suggestions using a corpus of 45 parent-child discussions that were recorded in casual contexts during full traversals of the story in a non-prompting condition. We evaluate our system's output and show that both the topics and the suggestions produced by our system are meaningful for people by conducting crowdsourced surveys. We conducted a user study in controlled lab settings to evaluate the relative efficacy our method has compared to not delivering suggestions at all and compared against suggestions authored by child literacy experts. Our results indicate that the suggestions produced by our system have a significant positive impact on the interactions, overall increasing the level of dialog present in the joint reading sessions. While literacy experts remain preferable if available, our results indicate that automated methods present a viable alternative, particularly in domains where access to expert data is limited or challenging (e.g., dynamically generated content).

CHAPTER 4
ANALOGY SOLVING AND EXPLAINING

4.1 Introduction

Analogy is a powerful cognitive mechanism that enables people to transfer knowledge from one situation or context to another. By identifying similarities between situations, reasoning through analogy facilitates understanding, inference making, learning new abstractions and creating conceptual change [Schiff et al., 2009]. Analogical reasoning is founded on the alignment-based model of similarity – the process of understanding an analogy requires reasoning from the perspective of structural and relational correspondence.

For AI, analogy solving presents an interesting and important problem because it offers the potential for deep problem understanding and automated generalization of learned tasks. In human learning, analogies have long been applied as a measure of verbal intelligence. Of particular interest to this work are verbal analogy questions commonly used on human standardized tests designed to evaluate understanding of relationships between a broad vocabulary of words.

A verbal analogy has the form A:B::C:D, meaning "A is to B, as C is to D". A question is formed by the first pair of words (A:B), followed by a number of possible answer pairs (C:D, E:F, etc.); the task is to select the answer pair for which the relation between the pair of words is the same as for the question pair. For example, given the initial pair *ostrich:bird*, and the options (a) *cat:feline*, (b) *primate:monkey* and (c) *chair:lounge*, the correct answer is (a), because the same *is a* relation connects the first word to the second word on both sides of the analogy.

Prior work in this area of AI includes several techniques for solving analogy questions designed for humans. Automated methods rely on latent analysis [Turney, 2006]. Crowdsourcing answers has been attempted, with no performance gains over statistical methods [Lofi, 2013]. None of these approaches produce interpretable justifications, focusing only on providing correct answer choices.

In this chapter, we argue that complete analogical reasoning requires more than just the ability to select the correct answer choice. Equally importantly, we believe an analogical reasoning system must be able to effectively *model* and *explain* the mutual relationship that connects the pairs of words. Toward this end, we contribute the Semantic Similarity Engine (SSE), a framework that leverages noisy semantic networks to answer and interpret analogy questions.

Analogy questions designed for human test takers cover a broad vocabulary and wide spectrum of relations. Leveraging ConceptNet, which has the necessary coverage despite the inherent noise, we introduce techniques for reducing the concept-relation search space by extracting the graph context, evaluating relational sequence similarity within word pairs, answering questions using similarity ranking across word pairs and generating human-readable explanations for analogies. Figure 4.1 shows an overview of our system. The input and output of our system are closer to human-readable text than to structured representations, and 96% of human evaluators agreed with our analogy explanations (more details in Section 4.5).

Through SSE, we contribute the following:

- An approach that focuses on deriving pairs relational chains form context-graphs for evaluating analogical strength;
- A metric for evaluating how similar an relational path pair is;
- A method of deriving human-readable interpretations from pairs relational chains.

In this chapter we present our approach to modeling analogies in detail. First, we present the Semantic Similarity Engine in Section 4.3. Then we show how these methods can be used for both answering analogy questions and generating human-readable explanations. Finally, we evaluate the performance of our approach for both of these tasks.

4.2 Related Work on Automated Analogical Reasoning

Deriving relational information and using it for inference has long been an area of interest in artificial intelligence. Logic representations can be considered one such example, for example first order logic being used to express relations and conduct inference in applications such as theorem proving [Fitting, 1996]. Logical inference follows a strict, deterministic, chain of productions, while more recent approaches such as Probabilistic Soft Logic modeling non-deterministic systems [Kimmig et al., 2012].

Analogies as used in natural language are different however, since they rely on partial or approximate matches to construct inference and draw parallelism. In an analogical setting, only a subset of the total possible correspondences between the entities that are compared are relevant. Specific to analogies as described in language, several directions have emerged.

One line of work, enabled by the availability of rich and large text corpora, has focused on answering multiple choice analogy questions via unsupervised latent analysis [Turney and Littman, 2005, Turney, 2006]. Similar to LSI [Hofmann, 1999], the authors introduce Latent Relational Analysis (LRA), in which the relation formed by each side of the analogy is modeled as a latent variable. Answers are selected based on likelihood ranking. The peak performance of LRA on the

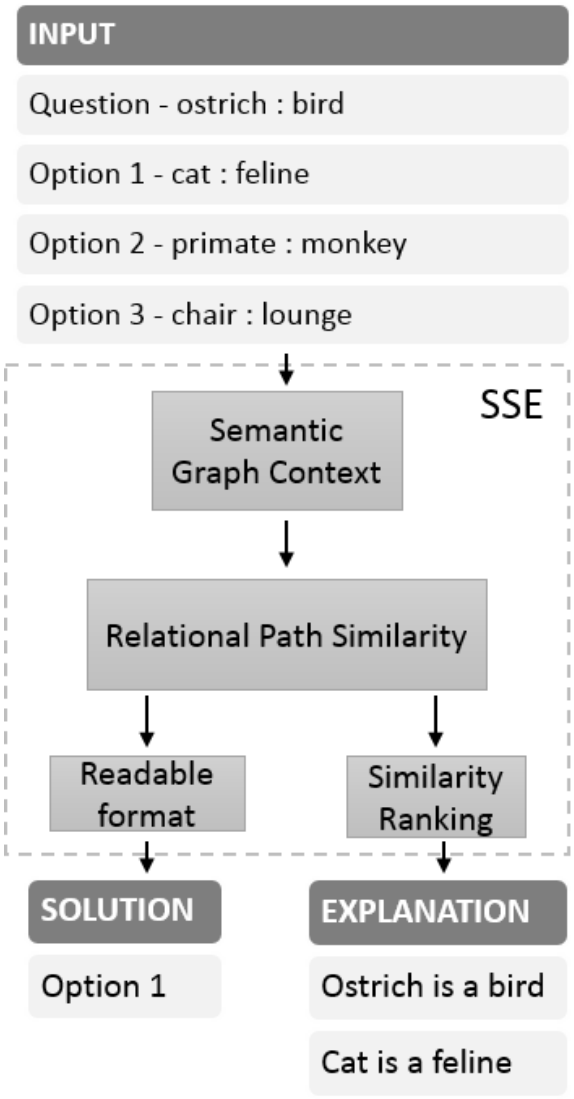


Figure 4.1: Overview of the Semantic Similarity Engine and its application to answering analogy questions and explaining the analogy relationship.

SAT dataset (which we also use in our work for evaluation) is 56%. More recent approaches incorporate supervised learning, building on previous statistical methods [Turney, 2013]. We note that these methods focus only on answering performance. They do not offer explanations for choosing an answer that can be expressed with a known set of relations, which is the main focus of our work. For this reason, we view our work as complementary. If the sole goal is question answering and the performance metric is the number of correct answers, then statistical approaches are likely to have more coverage and to obtain better scores than our method, as these methods do not require detailed relational data linking the concepts.

Analogy Space is another approximate method for estimating relational links between words, enabled by the availability of large semantic networks such as ConceptNet [Speer et al., 2008]. Instead of using raw text like LRA, it uses a singular value decomposition of a large semantic network in order to derive how similar a pair of concepts is.

Structure mapping theory has been developed from psychological observations [Gentner, 1983]. This theory proposes that, when forming analogies, humans focus on matching representations globally instead of locally, and that correspondence relies on both structure and features – equivalent parts of a whole perform equivalent functions. Another concept introduced by structure mapping theory is that in an analogy there is a one-to-one correspondence between parts.

Structure mapping theory has been implemented into the Structure Mapping Engine (SME). SME enables matching of relational characteristics between two semantic frames [Gentner et al., 1997]. SME has been applied in multiple contexts, including sketch classification [Chang and Forbus, 2012] and games [Hinrichs and Forbus, 2007]. Unlike LRA or Analogy Space, SME uses specific relational matches to infer similarity. One limitation of SME is that the relational structures that are evaluated to identify analogies need to be coded manually, which is a non-trivial design.

4.3 Semantic Similarity Engine

Figure 4.1 shows the block diagram for SSE and how it processes analogy questions. The system has two common steps: extracting semantic contexts represented as graphs for each pair of words, and computing sequence similarity. This common core is then used for the tasks of explaining analogies and answering multiple choice questions.

4.3.1 Semantic Context Subgraph Extraction

The first stage of our pipeline is to extract the context defined by a pair of words, which we refer to as the *start words*. The goal of this stage is to model the relationship between the start words by identifying multiple semantic paths between them. We refer to chains of nodes and the relations connecting them as *sequences*, and define the *context* of a pair of start words as a graph containing the start words and sequences of nodes and relations by which they are connected.

It may not be immediately clear why we are seeking to identify multiple paths within the semantic network. In fact, many word pairs in our analogy dataset are directly linked by one (or more) of the 46 relationships within ConceptNet. However, indirect paths through other nodes may provide greater insight into the relationship of the start words themselves. Figure 4.2 presents an example of such a case, visualizing the graph extracted from ConceptNet for the word pair *goose*:*flock*. The correct relation implied by the analogy is *part of*, which is represented in the graph, but only for the superclass of *goose*, i.e. *bird*. The start words *goose* and *flock* are directly

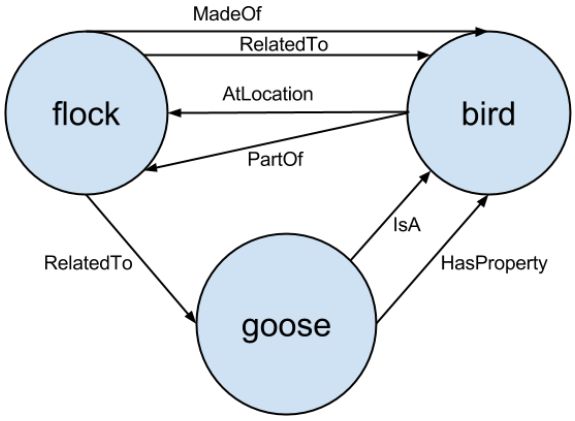
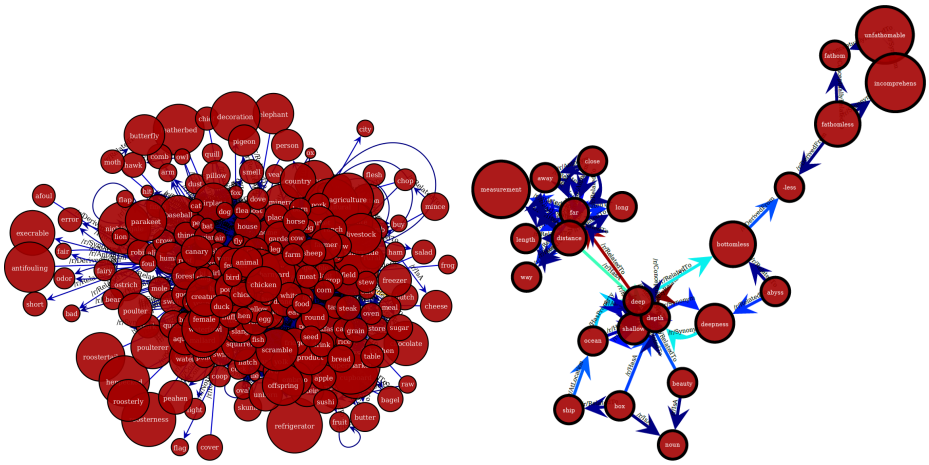


Figure 4.2: Example context surrounding *goose* and *flock*. The most meaningful sequece of relations is through an intermediate node, *bird*, and not the *Related To* edge which is directly connecting the nodes.



(a) (b)

Figure 4.3: Unpruned (a) and pruned (b) context graphs.

connected, but only through a less informative *related to* edge, while both have stronger connections with *bird* through *is a* and *part of* edges, respectively. It is therefore necessary to explore multiple paths of different lengths in order to reason effectively about the relationship between these words and find a good analogy explanation.

We generate the context graph for a given pair of start words in two steps. First, we extract the **unpruned semantic context**. This is performed by recursively expanding concepts in breadth-first order starting from the start words, caching a subgraph from the full semantic graph (i.e. ConceptNet). The entire graph is too large (tens of GB) to be accessed efficiently without caching. The stopping condition is a limit on the number of explored concepts. At each node addition in the

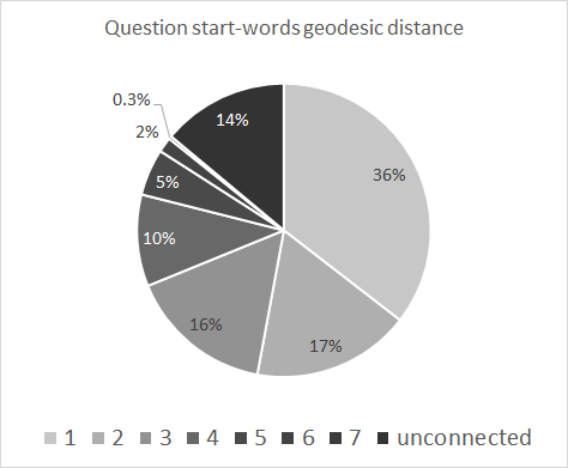


Figure 4.4: Pie chart showing the proportion of start word pairs in the analogy dataset that have a geodesic distance of 1 (directly connected) through 7, or are unconnected.

breadth-first exploration, we test if there are edges to or from the rest of the nodes in the context-graph, and add them if so. This ensures that all existing relations between the context's words are captured. Figure 4.3(a) shows an unpruned graph example.

If the search fails to find a sequence between the start words, then analogical comparisons with another graph are not possible. This occurs for 14% of the word pairs within our dataset when using a 500 word limit for expansion. Using a larger limit did not impact results. Additionally, attempting to identify long sequences connecting the start words does little to aid the analogy solving or explanation process, since long sequences become difficult to interpret.

The context graph contains many leaf nodes that are irrelevant to the analogy. In the second step, we **prune** the graph by removing any nodes that are not part of a sequence between the start nodes. Edge direction and weight are ignored at this step. The result is a much smaller graph, typically consisting of tens of nodes, as illustrated in Figure 4.3(b).

4.3.2 Sequence Similarity

Now that we have a method for extracting the pruned context subgraph for any single pair of start words, we describe how two such contexts can be compared to determine the similarity in the relation between them. Specifically, we present an algorithm for identifying the *highest similarity sequence pair (HSSP)*, the sequence of nodes and edges that has the greatest number of common edges between two contexts.

For a pair of context-graphs, we identify the HSSP by iterating through all possible pairs of sequences of the same length, one from each graph, and selecting the one with the highest similarity score. Algorithm 3 presents the algorithm for calculating the similarity score, which has a value

Algorithm 3 Sequence similarity. s_1 and s_2 have the same length, each connecting a pair of concepts in different contexts.

1: $sim \leftarrow 1.0$
2: **for** $k\ in\ range(1, length(s_1) - 1)$ **do**
3: $\quad s_{1,k} \leftarrow s_1[k : k + 1]$
4: $\quad s_{2,k} \leftarrow s_2[k : k + 1])$
5: $\quad edges_1 \leftarrow context_1.get\_rel\_types(s_{1,k}[0], s_{1,k}[1])$
6: $\quad edges_2 \leftarrow context_2.get\_rel\_types(s_{2,k}[0], s_{2,k}[1])$
7: $\quad sim_k \leftarrow CommonRelProp(edges_1, edges_2)$
8: $\quad sim \leftarrow sim * sim_k$
9: **end for**
10: **return** sim

between 0 and 1. For each sequence pair, s_1 and s_2, the algorithm iterates over the length of the sequence (line 2). For each segment, we compute the size of the set of common relations relative to the total set of relations present on that segment (lines 5-8). For example, comparing segments *A–B* and *C–D*, linked by relations {*p,q,r*} and {*r,p*} respectively, the similarity score becomes $2/3$, because there are two relations in common out of three total relations for this segment. At this stage, we do not yet take into account the weight or direction of edges, as this makes the algorithm more resilient to noisy edge additions or omissions within ConceptNet.

To generalize this algorithm for sequence pairs of arbitrary (but equal) length, we apply this metric to all segments of the sequence and multiply the similarity scores (lines 8). This ensures that if at any point the sequences are entirely dissimilar, the overall similarity is zero.

4.4 Modeling Analogies through SSE

The methods presented in the previous section allow us to find the best common relational link between two different pairs of words by searching a large semantic network. In this section, we describe two applications for understanding the relationship between word pairs: solving analogy questions and explaining analogies.

4.4.1 Answering Analogy Questions

Our approach to solving analogy questions stems directly from the similarity score obtained from the SSE. Our questions take the form presented in Figure 4.1. To select an answer, we first compute the HSSP between the question word pair and each possible answer. Then, we rank all answer options by their respective HSSP score and select the one with the highest score.

Options that have a similarity score of 0, or for which a context-graph connecting the pair of words can not be found, are discarded. We can then use the sequence pair that generated the similarity value (i.e. HSSP) to explain the analogy, as discussed in the following section.

In the results presented in this chapter, we do not attempt to answer the question if there are no answers with a similarity score greater than 0. It is trivial to extend our technique to allow the algorithm to simply guess one of the multiple choice options. We do not utilize random guessing to more accurately reflect the performance of the algorithm, and to facilitate our main goal of studying how the relationship behind the analogy can be explained. An answer obtained through guessing would make our algorithm, just as a human student, unable to explain the similarity between the two word pairs.

4.4.2 Explaining Analogies

Established practices for teaching human students to solve analogies instruct them to do so by forming a full sentence that clearly shows the relationship between the two question words, and then forming a second sentence that shows a similar relationship for their chosen answer word pair. The aim of our work is to generate the sentences that describe these relationships automatically.

If two pairs of words are stated to be an analogy, we can produce an interpretation from the corresponding HSSP. In order to obtain output that is easily readable, we need to reduce the HSSP from having multiple relations per segment to a single chain of relations (Algorithm 4). Therefore, we iterate through each segment along the sequences (line 4), and choose the salient common edge between the two sides of the analogy (lines 6-10), appending it to the explanation pair along with the corresponding concepts (lines 5, 10). Note that edge direction is taken into account in these steps.

We select the common relation with the highest minimum-weight, preventing imbalances in which only one side of the analogy is strongly related while the other is relatively weak. Figure 4.5 shows and example, in which the bolded relations are selected according to Algorithm 4.

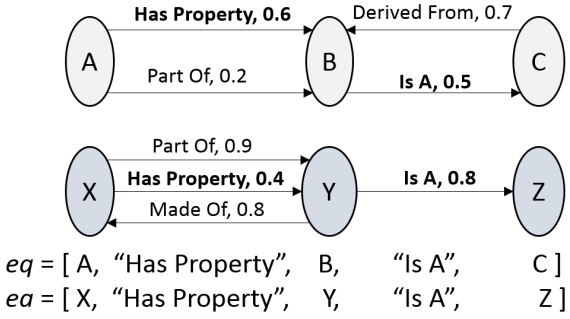


Figure 4.5: Starting from HSSP, we select the common salient edge on each segment to produce human-readable explanations. *eq* and *ea* are then converted to English.

Once a relation is selected for each segment, we convert the resulting list of nodes and relations into English using a dictionary which maps coded relations names to more readable versions (line 13). For example, a *PartOf* edge translates to *"is a part of."* While more sophisticated methods

Algorithm 4 Generating a human-readable explanation from the best similarity sequence pair, which have the same length.

$sq \leftarrow question\_seq$; $sa \leftarrow answer\_seq$
$eq \leftarrow [\,]$; $ea \leftarrow [\,]$
$n \leftarrow length(question_s eq)$
for k *in* $0...(n-1)$ **do**
  $eq.append(sq[k])$; $ea.append(sa[k])$
  for rel *in* $\cap(sq[k:k+1].edges, sa[k:k+1].edges)$ **do**
    $support[rel] = \min(sq[k, k+1].rel.weight, sa[k, k+1].rel.weight))$
  end for
  $rel\_max \leftarrow rel\ for\ which\ \max(support[:])$
  $eq.append(rel\_max)$; $ea.append(rel\_max)$
end for
$eq.append(sq[n])$; $ea.append(sa[n])$
return $convert\_to\_english(eq, ea)$

can be used to generate explanation phrases, grammatical correctness it is not the focus of our work; human evaluators were asked not to assess grammatical correctness.

4.5 Evaluation Setup

We evaluates the SSE's ability to correctly answer and explain analogies using two datasets:

- 373 questions used in SAT US college admittance tests. This dataset was also used in previous work on answering analogies [Turney and Littman, 2005]; Table 4.1 shows question examples;
- 227 questions from a public domain website[1] targeted for grades 1-12. We combine these into four groups: elementary school (grades 1-4), middle school (grades 5-8) and high school (grades 9-12), containing 120, 60, and 47 questions, respectively.

In both datasets, each multiple choice question contains five possible answers. Combined, these questions form a progression of increasingly difficult analogy problems.

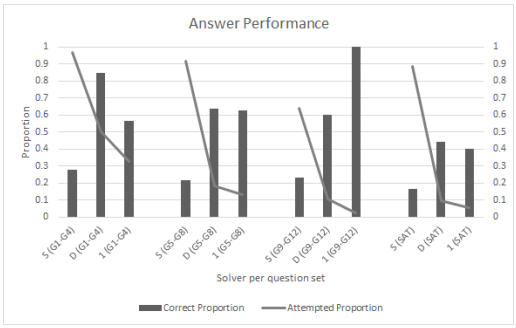
Question Answering Performance

In this section, we evaluate the SSE's performance in answering questions. We track two performance metrics:

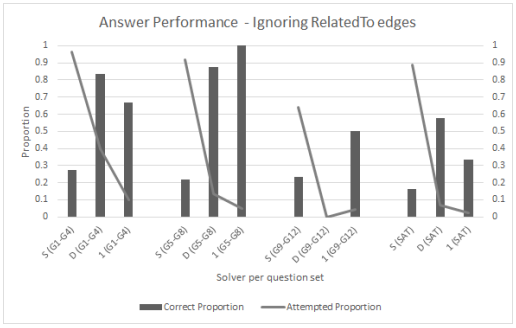
[1]Section "Unit 2: Read Theory Word Pair Analogies" from `http://www.englishforeveryone.org/Topics/Analogies.htm`

Table 4.1: Question examples from the SAT dataset and the answer results of our approach (correct answers shown in bold). ATT stands for Attempted, CORR stands for Correct.

QUESTION	OPTION A	OPTION B	OPTION C	OPTION D	OPTION E	ATT	CORR
custom:society	hypothesis:evidence	testimony:trial	ballot:election	**rule:game**	contest:debate	yes	yes
seed:plant	pouch:kangaroo	root:soil	drop:water	bark:tree	**egg:bird**	yes	yes
lull:trust	balk:fortitude	betray:loyalty	**cajole:compliance**	hinder:destination	soothe:passion	yes	no
virtuoso:music	**bard:poetry**	crescendo:scale	lyricist:melody	portrait:photography	critic:performance	yes	no
querulous:complain	silent:talk	humorous:laugh	dangerous:risk	**deceitful:cheat**	gracious:accept	no	-
audacious:boldness	anonymous:identity	remorseful:misdeed	deleterious:result	impressionable:temptation	**sanctimonious:hypocrisy**	no	-



(a)



(b)

Figure 4.6: Answer performance on four datasets, comparing the direct (D) and one-hop (1H) methods with the Divisi similarity baseline (S) both when using (top) and when ignoring (bottom) statistically derived relations. Questions are grouped by elementary grades (G1-G4), middle school grades (G5-G8), high school (G9-G12) grades and SAT.

1. *answer attempt proportion*, which represents the number of questions for which our algorithm selected an answer. As discussed earlier, our system attempts to answer a question only if there is at least one answer for which the HSSP has a non-zero similarity score, thus preventing random guessing.

2. *answer correctness proportion*, which represents how many of the attempted questions were answered correctly.

In our analysis, we limited the semantic context subgraphs to have a maximum geodesic distance of two. Longer paths, while feasible, proved to result in few answer attempts. In the results, we separately report performance for solutions with a geodesic distance of 1, which we call *direct*, and those with a distance of two, which we call *one-hop*, to demonstrate the frequency of occurrence and reliability of both cases. We show results separately because an answer may be available for each geodesic distance. These answers could be combined via ensemble methods to increase the answer attempt proportion, but that extension is outside the scope of this work.

Additionally, to establish how well the SSE measures similarity, we compare against a baseline approach that relies on ConceptNet's own metric for similarity provided by the Divisi toolkit [Speer et al., 2010]. This metric is a real number between 0 and 1, calculated using the SVD within ConceptNet. In this experimental condition we performed question answering as follows: for each pair of start words, we computed the Divisi similarity value; we selected the answer for which the similarity was numerically closest to the similarity of the question pair, attempting answers only if the question and at least one answer had non-zero similarity.

Figure 4.6(a) presents answering performance for direct, one-hop and baseline methods over the analogy data set. The solid line shows the proportion of attempted questions, while the histogram presents the proportion of correct answers of each method. Across all four question levels, the baseline technique attempts to answer a large percentage of questions, but has low accuracy. Its best performance is on the elementary school data set, where it achieves 28% accuracy. By comparison, both SSE conditions (direct and one-hop), are less likely to attempt an answer, but have a significantly higher accuracy.

In the elementary grade data set, the direct solver achieves an accuracy of 85%. While performance declines as question difficulty increases, both solvers answer correctly 83% of attempts on average in the 1-12 grade dataset. As questions become more difficult, especially for the SAT dataset, knowledge of the words' meanings becomes key. SAT questions often focus on rarely encountered words, so it is unsurprising that the attempt ratio decreases due to lack of connections between the start words. Despite this, the SSE methods achieve answer correctness of 40% on the SAT dataset.

The experimental results presented in Figure 4.6(a) were obtained by utilizing all 46 relations found in ConceptNet. In Figure 4.6(b) we present similar results for a second condition in which we ignore two relation types, *RelatedTo* and *ConceptuallyRelatedTo*, which are unique within ConceptNet because they are derived statistically from document co-occurrence and thus are far more noisy. Moreover, they are not useful for generating explanations. In this condition we note that the attempt proportion is lower, since many edges within the context are ignored. However, the accuracy for attempted question is higher than in the original condition, providing a means of regulating the algorithm's behavior in focusing more or less on accuracy vs. attempts.

In summary, the results demonstrate that the SSE-based methods for analogy solving achieve high accuracy in this difficult domain, but sacrifice coverage by not attempting to answer ques-

tions which would require the algorithm to guess. We made this design choice due to our focus on explaining analogy questions, which requires the ability to accurately model the relationship between word pairs. We discuss our results for generating explanations in the following section.

Explanation Performance

The ultimate goal of our system is to generate full sentence, human readable explanations of analogy question solutions. To evaluate our success, we surveyed people to test whether the explanations generated by our algorithm were meaningful to them. We conducted our survey through the CrowdFlower crowdsourcing market using the 74 explanations (60 direct, 14 one-hop) produced by SSE from the correct answers selected when ignoring *RelatedTo* and *ConceptuallyRelatedTo* edges.

For each explanation, the survey first presented the corresponding analogy question in its full form, including the question statement and all possible answers (as in Figure 4.1). Then we told readers the true answer, followed by the explanation. Participants were asked to choose whether the explanation correctly justified the answer.

To evaluate the quality of our explanations, and to ensure that human readers were paying attention, we ran this study by randomly interleaving examples from three conditions:

1. *SSE-generated explanations* - an explanation generated from a HSSP selected by SSE;
2. *randomized explanations* - we substituted the relations in the SSE-generated set with random selections from the set of 44 relations;
3. *edited explanations* - we substituted the relations in the SSE-generated set with manually edited relations that were contradictory to the true relation.

Table 4.2 presents examples of all three explanation types. We included the randomized and edited conditions in our analysis because many relations within ConceptNet are similar, and thus selecting one at random may result in a relation that was very close, but not identical to, the SSE-selected one. Our goal was to verify that the SSE-selected relations, and the explanations derived from them, were clearly correct and differentiable from both random noise and wrong answers.

In total, the surveys tested 222 explanations (three study conditions applied to 74 questions), and each question received at least 5 judgements (5.79 on average). Figure 4.7 reports the proportion of analogy explanations that participants considered to be valid for each condition, reported separately for direct and one-hop solvers. We observe that human evaluators agreed with 96% of the explanations produced by our method – all but a single one-hop explanation were accepted. Approximately half of the randomly selected explanations were considered valid, and we observed higher disagreement between participants in this dataset (study-wide inter-user agreement was high, 87% on average, but only 78% for randomized condition). When analyzing these instances case by case, we found many of the randomly selected explanations to be reasonable, if not

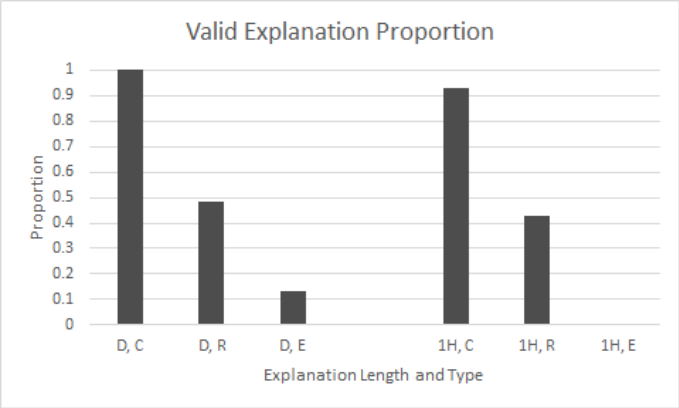


Figure 4.7: Evaluation of explanation quality for direct (D) and one-hop paths (1H), comparing our system's output (C) with randomized (R) and edited (E) conditions.

Table 4.2: Question examples from the SAT dataset and the answer results of our approach (correct answers in bold).

Solver	Dataset	Explanation Pair
Direct	SSE	**fire** has property of **hot** **ice** has property of **cold**
Direct	Randomized	**fire** is a member of **hot** **ice** is a member of **cold**
Direct	Edited	**fire** is not **hot** **ice** is not **cold**
One-hop	SSE	**tub** is used for **bath**, which is at location **bathroom** **stove** is used for **cook**, which is at location **kitchen**
One-hop	Randomized	**tub** is located near **bath**, which is attribute of **bathroom** **stove** is located near **cook**, which is attribute of **kitchen**
One-hop	Edited	**tub** is participle of **bath**, which inherits from **bathroom** **stove** is the participle of **cook**, which inherits from **kitchen**

entirely sound. However, in the edited dataset, which contained intentionally illogical relations, very few were considered valid. This result strongly supports the validity of the SSE-generated similarity relationships and the analogy explanations founded upon them.

Finally, we note that the relatively strong performance of the randomized dataset suggests that humans were less sensitive to the exact wording of the analogy explanation and accepted relatively close relation substitutes, as long as the substituted relation was sufficiently similar to the one intended by the analogy. This result has broader implications, as it suggests that correctly identifying the *ideal* relation represented by the analogy may not be necessary. Computationally, the set of all possible analogy relations is potentially very large. However, if we allow for approximations, multiple analogy relationships can be collapsed, a hypothesis that is supported by these results.

4.6 Conclusion

Analogies are an essential reasoning pattern that enables learning through similarity and skill transfer. We presented a method for evaluating analogical similarity by comparing paths within semantic context graphs derived from large scale noisy semantic networks. We demonstrated the effectiveness of our approach on two applications, solving multiple choice analogy questions and generating human-readable explanations for the analogies. Our results demonstrate that our methods achieve high accuracy in correctly answering attempted questions, and surveyed study participants agreed with the analogy explanations generated by our algorithm in 96% of cases.

We will conclude this chapter with an enumeration of the alternative attempts that we made at solving the problem of explaining wide-vocabulary analogy questions in order to better inform potential future work in this direction. We considered other designs for the context-creation and relational path comparison modules of SSE but these proved to be less successful. Instead of searching for similar path pairs, a different approach would be to seek a single path between the words in either the question or the answer choice, similar to answering queries over semantic networks [Pérez et al., 2006, Unger et al., 2014]. However, one key assumption of query answering is that the requirements are stated in the query, with two main tasks required for answering: (1) mapping the query to the representation used by the semantic store, and (2) selecting the salient answer. This is not the case for analogy answering, which is a fundamentally different task: a search problem instead of an identification problem.

The relational path similarity module follows one of the assumptions of alignment-based theory, in that it uses one-to-one mapping along the path segments. We attempted to forgo this assumption by using probabilistic matching of entire contexts simultaneously. We attempted probabilistic identification both between path pairs, and between more complex contexts, using Probabilistic Soft Logic [Kimmig et al., 2012, Beltagy et al., 2014]. However, this approach proved inadequate for two reasons: (1) analogy questions designed for humans tend to focus on highly specific relations, which results in a highly sparse similarity space; (2) in order to identify equivalence of structures more complex than path pairs, a reliable method for selecting these structures would be needed. Both of these requirements are not possible to obtain *a priori* because the analogy answering problem is a search problem. Instead, we anticipate in future work using probabilistic models to identify relational equivalence after the analogy question has been answered.

Furthermore, we attempted to use edge-specific information to better inform the search process. If inheritance is propagated along *Is-A* relations and the respective nodes are merged, SSE is able to answer a very small number of additional questions correctly. This is because paths of different lengths are collapsed and shortened, effectively normalizing for different resolution representations. However, we did not include this approach in our final results because it is dependent on the semantic network of choice. For future work, it would be possible to construct

relation-specific heuristics that would improve performance at the cost of generality.

CHAPTER 5
OBJECT SUBSTITUTION IN ROBOT TASKS

5.1 Introduction

In most cases, people can easily replace a missing ingredient when packing lunch, or replace a dirty utensil while cooking. Robots, on the other hand, are far less robust in plan execution, with missing objects typically leading to failure. Several research projects have explored how task recovery can be performed by searching the surrounding environment for the missing item using a semantic map [Hermans et al., 2014, Guadarrama et al., 2014]. However, that approach assumes that the object type stays the same. In this chapter, we present an object substitution algorithm to address execution failures due to missing objects. We contribute a context-aware algorithm that leverages task information to propose potential substitutions. Initially, a human user verifies these substitutions; over time, the robot builds a case library of suitable substitutions, which may also be personalized to meet the needs of individual users.

We define *object substitution* as an extension to the broader problem of *plan repair*. Plan repair is an established research area that explores techniques that enable an agent to locally alter an existing plan to overcome changes in the world state [Gerevini and Serina, 2000, Van Der Krogt and De Weerdt, 2005]. Plan repair is preferable over discarding the initial plan and re-planning both computationally, because only few intermediate states need to be recomputed, and from a usability standpoint, since maintaining a similar approach is more predictable[Fox et al., 2006, Koenig and Likhachev, 2002b]. Object substitution is a variant of plan repair that extends the planning domain through a set of valid substitute objects before the plan is repaired in other ways, if needed. For a planning problem $P = (O, I, G)$, where O is a finite set of objects, I the initial state, and G the goal state, we define the *object substitution problem* as identifying O', with $O \subset O'$, such that a plan $P' = (O', I', G')$ is feasible, where I' and G' are similar to the original I and G [Fox et al., 2006].

In this work, we examine the problem of object substitution in the context of high level task planning using Hierarchical Task Networks (HTNs). Previous work has proposed performing substitution based on object affordance properties [Awaad et al., 2013], however such information is often noisy and difficult to obtain. We hypothesize that information about the *task context* can instead be leveraged to determine a successful object substitution with high accuracy. In the fol-

Figure 5.1: Screenshot showing the robot autonomously retrieving a roll of *tape* for the *pack schoolbag* task, after receiving user approval to use it instead of *glue*.

lowing sections we first present our method and then evaluate its performance in nine planning domains. Our results show that using context information leads to better classification performance of valid substitutions than a context-agnostic baseline. Furthermore, our method is resilient to variations in the context while maintaining better performance than the baseline. Additionally, we demonstrate that our method produces actionable substitutions by implementing two of the tasks, *pack schoolbag* and *wipe table*, on a physical robot. Examples of the substitutions our system generates include using chopsticks instead of a fork when setting the table, using a cleaver instead of a knife for chopping vegetables to cook soup, and using tape instead of glue when packing a schoolbag.

5.2 Related Work

Existing work on achieving robust plan execution proposes substituting objects based on affordance similarity [Awaad et al., 2013]. Gathering a comprehensive database of affordances is a difficult knowledge representation problem, which would require considerable effort and careful initial planning. This approach also poses a challenge for applying substitution to previously unseen objects, although its detailed representation has been shown to be useful in monitoring success during task execution [Konecnỳ et al., 2014]. We review methods on robot learning of affordances and affordance representations in Section 5.2.1. Also related to achieving robust robot plans is plan generalization, which has been used in one-shot-learning to remove constraints falsely in-

ferred from a small sample pool so that the learned models have greater coverage [Wilson and Scheutz, 2014]. This method relies on semantic knowledge to propose more general scenarios, which are validated symbolically. Our method may perform similarly by replacing objects with their superclass (e.g. $apple$ with $fruit$). However, our method can also create replacements that are distinct alternatives to the original, and not generalizations. Expanding on these concepts, Section 5.2.2 gives an overview of the fields of symbolic planning and plan repair.

In defining context for the substitution, we leverage abstract concepts encoded in Hierarchical Task Networks (HTNs), which represent plans as trees, with atomic actions on the leaf nodes, and logical groupings on the higher levels [Ehrenmann et al., 2002]. We derive bag-of-words contexts from the HTNs. Initially used in information retrieval [Croft et al., 2010], bag-of-words models have been used in physical-domain applications such as computer vision [Bolovinou et al., 2013], scene recognition [Botterill et al., 2008], or robot navigation [Nicosevici and Garcia, 2012].

In addition to the information that a plan contains, we rely on language knowledge from general purpose semantic networks to evaluate substitutions. To evaluate substitutions, we use ConceptNet Divisi and WordNet path length similarity, which are reviewed along with other similar resources in Chapter 2. Using these semantic networks, we evaluate substitution using a number of measures, including analogical similarity. Analogical similarity represents correspondence at a relational level and is distinct from surface similarity which takes into account directly observable features such as color. Analogical similarity has been modeled using semantic networks. The Structure Mapping Engine (SME) [Gentner, 1983], which is the implementation of alignment-based similarity theory, creates mappings between semantic representations at a global level. Semantic Similarity Engine (SSE) [Boteanu and Chernova, 2015] creates a different type of mapping, that focuses on the parallel relational paths that form an analogy. SSE also relies on some alignment-based concepts such as one-to-one mapping, but it is distinct from SME. In this work, we use the similarity values produced by SSE to measure how similar the target and candidate concepts are to a context word. Semantic networks encode information that is not specific to the task, thus we reuse existing knowledge about concepts. With respect to previous work on reusing knowledge, our approach is more alike improvisation than systematically reusing of known patterns such as case-based learning [Kolodneer, 1991, Veloso and Carbonell, 1993, Craw et al., 2006] or case-based planning [Hammond, 1986, Cox et al., 2005, Borrajo et al., 2015].

Finally, since our approach to object substitution includes using human feedback, we review the field of Learning from Demonstration (LfD) in Section 5.2.3.

5.2.1 Object Affordances

Object substitution in robot tasks implies that affordances applicable to the original object within the task need to be available for the substitute, either the same affordances or equivalent ones. This requirement is particularly apparent for the case of object substitution we address in this the-

sis, in which only the object is replace but not the intermediate-level tasks. One straightforward solution would be to attempt to solve this correspondence problem by maintaining an affordance database. To the best of our knowledge, unlike for most other artificial tasks such as image processing or symbolic planning, there is no readily available general purpose affordance collection; benchmarks are missing as well. One possible reason for this lack is may stem from the lack of a a widely used collection of objects for robotics applications.

Instead of trying to explicitly solve the affordance correspondence problem that object substitution creates, we reply on human feedback to refine substitutions. Part of this feedback, we assume affordances to be evaluated. However, since affordances are difficult to define consistently for a large collection of objects, we argue that human feedback will be initially necessary regardless of whether there is available an affordance database or not because it is improbable that the knowledge base (a semantic network such as the ones we use, or an affordance database) will contain sufficient information to make correct substitutions for a wide variety of tasks and domains. As we detail in Section 5.6, our method may infer connections between substitution quality and relational parallelism between the target and the candidate with respect to the context. Some of these relations (edges in the semantic network) may in turn correspond to broad affordances, such as *used for* or *capable of* in ConceptNet.

With the prospect of inferring such information in the future, we will summarize existing approaches to modeling and learning affordances in the remainder of this section. There is a broad spectrum of concepts covered by the term "affordance" [Raubal and Moratz, 2008]:

- **Physical affordances** enable the interaction and manipulation of an object, for example round objects can roll, solid flat objects are stable resting on a flat surface, objects that a human can sit on have a number of similar characteristics;
- **Perceived affordances** are defined by a context, which usually blends physical and social norms. These affordances are more complex to use since they require interpretation: a knife can be used to cut, but it is only acceptable to cut certain things; the speed limit on a road is imposed by law, it is not derived only from the physical limits of the car.
- **Mental affordances** are perceived with respect to an institution or a service, for example using a bus ticket or making a phone call.

From the above list, the physical and perceived affordances are what our object substitution system targets. By relating the substitution target and candidate with the context, relations corresponding to both can be modeled via the edges in ConceptNet. For example, physical affordances could be expressed explicitly, through *has property* edges, or indirectly, by inheriting properties from another instance within a class, for example *apple* and *peach* can be presumed to have some properties in common since they are both linked to *fruit* through *is a* edges. We note that it is not necessary to explicitly define which affordances are shared, which leads to some robustness

at the expense of specificity. Perceived affordances could be related to ConceptNet edges such as *at location* or *used for*. When applied to concepts related to common human environments, these edges correspond to logical groupings based on use.

We will now examine two methods through which a robot may learn new affordances from the outside world without explicit programming. One line of work in affordance learning is taking inspiration from the natural world and proposes that robots learn about the world and their relation to the world via observation and experimentation. An iterative approach is envisioned, in which existing knowledge is used to learn new skills via layers of joint perception and motor-skill models [Fitzpatrick et al., 2003]. The authors cite neural science results showing that manipulation is an essential part in knowledge acquisition for humans and relate them to the robot affordances described by [Gibson, 1977]. In this view, an affordance is a visual characteristic of an object that can enable actions to be performed on it without a full understanding of what the object is. This work presents a robot that learns simple one-object manipulation tasks through exploration: pushing, pulling and poking. Before the exploration, the robot learns an end effector control policy which takes into account the start and target positions. The policy learning process is structured into two steps: (1) Representation and Learning, in which the robot develops a map between initial hand positions and target positions; (2) Testing the Maps, in which the previously learned policy is tested in order to evaluate the error rate for each position. This approach may require significant effort for each new affordance, which may be impractical for more complex scenarios.

Other work derived from neurological research models the problem of learning object affordances through direct exploration as finding correspondences in Bayesian Network between the perceived object properties, the actions that the robot takes and the observed interaction (motions) [Montesano et al., 2008]. The robot's initial knowledge is limited to vision primitives (segmentation, color and shape classifiers). It then constructs a map between actions, objects and affordances. A shortcoming of this work is that a new classifier must be trained for every new set of objects because the relations are coded rigidly. Other work proposes to use *Statistical Relation Learning* [De Raedt and Kersting, 2008] in order to learn how an action's effects on an object relate to its affordances. SRL blends probabilistic inference, logical programming and machine learning into a programming language that can infer relations from both productions and data [Moldovan et al., 2012]. This work focuses on a tabletop manipulation scenario in which affordances are studied on individual simple objects. The core contribution is the use of a Bayesian Network and a probabilistic language, PPL, to model the scene.

Knowledge Representations for Affordance Learning

Semantic knowledge has been previously used in robot planning [Rogers and Christensen, 2013]. Relations expressing properties of the world can be used to narrow down the target areas in a search problem - knowing where to look is more efficient than an exhaustive search. In this section

we will review methods of representing knowledge, focusing on how these representation impact affordances that can be represented in the system. We can distinguish three stages regarding the way knowledge is acquired and used by a robot system:

1. **Pre-programmed:** Initially, the use of affordance theory in robot applications was narrowly focused on industrial pick-and-place applications. In this context, there is little variability in the environment and in the set of objects that should be manipulated. This allowed classic AI block-world planners to be adapted for grounded robotic work [Kuniyoshi et al., 1994]. Limitations in computing power, along with real time processing restrictions, restricted the object set to non-rotated cuboids. The system could replicate assembly tasks as demonstrated by a human in real time;

2. **Learned:** Most of the works reviewed in this section fall into this category. Either when learned through direct exploration or demonstrated by a human, in these approaches the robot builds a knowledge base by direct sensory input, without the use of knowledge gathered through external means. For example, [Veloso et al., 2005] note that their system knows what a *chair* is in the sense that someone sits on it, but it doesn't know how to identify it in the environment. The system is seeded with the concept of *chair* and it then learns to identify it in the environment;

3. **Assisted by External Knowledge:** In the remainder of this section we will focus on methods that use external knowledge bases such as ConceptNet for affordance learning applications.

Objects can share affordances. Usually this implies that the objects are similar themselves. Defining what similarity means is a challenging problem since similarity is defined relative to comparison criteria. As a result, the same objects can be grouped in various ways. For example, shoes can be grouped based on their size, color, or material if physical attributes are targeted, or they be grouped based on their intended purpose. According to our hypothesis for object substitution, the context and intended action or task determine that if two objects are considered different or not.

There exists theoretical work describing high level principles under which the hierarchy of object classes could be described [Awaad et al., 2013], which aims to define under which similarity conditions objects can be considered equivalent. Unlike our approach, this work only takes into account object similarity from a general standpoint and not their functional relations with the environment. By using only affordances to derive substitutions, it does not account for case-by-case aspects that may prohibit the substitution, such as etiquette or compatibility with other objects. According to this model, a concrete HTN specification for an object can be generalized such that the desired affordance is preserved. For example, a specific instance of a cup can be substituted for another cup, or more generally with any other drinking vessel. The work proposes using an

existing measure of semantic similarity to evaluate which objects, which would use observed parameters such as color and shape. Similar approaches query multiple ontologies in which the result is the intersection of all objects that can be substituted one for another [Varadarajan and Vincze, 2011]; ontologies for language, visual representations, grasp affordances and functional affordances are considered.

5.2.2 Symbolic Plan Repair

We refer to planning in an abstract propositional domain as symbolic planning to differentiate it from, for example, motion planning in robotics. The problem of planning was introduced as a search problem through a domain [Fikes and Nilsson, 1972]. This domain can be represented as a set of rules that allow for transitions between states. The problem is to find a sequence of successively applying the rules such that the state changes from an initial state to a goal state. In its simplest form, the transitions allowed by the rules are deterministic. There is a rich history of solutions that solve this problem. Heuristics [Hoffmann and Nebel, 2001, Bonet and Geffner, 2001, Richter, 2013], abstraction through semantics [Sacerdoti, 1974, Gil et al., 2011], and reusing information from previous plans generated in the same domain are some of the most important approaches to this solution.

Reusing information from known plans has been framed in two ways. Case-based planning (and case-based reasoning) proposes a database of previous plans that may be adapted or even combined in order to generate a new solution [Cox et al., 2005, Hammond, 1986, Cunningham et al., 2003, Kolodneer, 1991, Kolodner, 2014]. We will cite one example of reusing previous plans frequently applied in robotics and virtual games, the D-star algorithm, which extends A-star by maintaining a library of previous paths which were found using heuristic search [Koenig and Likhachev, 2002a].

Conversely, the information that is being reused could come from the original plan for which execution cannot continue. This is the case for dynamic domains, which do not assume deterministic operators. In the real world, partial knowledge of the environment and execution error result in non-deterministic execution [Cimatti et al., 1998]. When a plan fails during execution, there are two possibilities: either abandon the existing plan create a new one, or attempt to repair the existing plan so that execution may resume. In the worst case, repairing plans can lead to an entirely new plan, and may not be more computationally efficient [Nebel and Koehler, 1995]. However, efficiency should not be the only desideratum, in particular if the robot or agent is working in a human environment. In this case, plan stability, which can be measured as the editing distance between the original and the repaired plans, should have priority since it translated to more predictable behavior [Fox et al., 2006]. Solutions for achieving greater stability include performing a local search to replan at the failure point [Hanks and Weld, 1995, Van Der Krogt and De Weerdt, 2005], or backtracking to the root cause of the failure [Bidot et al., 2008].

5.2.3 Learning from Demonstration

In addition to verbal feedback, our approach to substitution could incorporate in future work direct demonstrations from the user. One field of robotics that was founded on the premise of using human input and feedback for robot learning is Learning from Demonstration (LfD), which we will briefly review in this section. There is significant evidence that infants learn from imitating others at least as much as from direct exploration [Lopes et al., 2010]. Whereas direct exploration may lead to increased motor control and low-level capabilities, learning through imitation is more effective in acquiring complex knowledge. In robotics, self-driven exploration of affordances has been limited to simple actions. By observing humans directly as they use objects, robots can learn high-level behaviors as well as more difficult manipulation tasks. Imitation is defined as an action that is copied successfully while understanding the goal of the original action [Call and Carpenter, 2002]. Understanding the goal poses significant challenges since distinguishing between the motor actions themselves, the sub-goals of each motor-action, and the task-level goals is often an ambiguous. In LfD, imitation is further classified based on the perceptual input of the robot [Argall et al., 2009]:

1. **External Observation**, in which the robot usually has visual input of the demonstration and therefore needs to interpret the scene before it can assimilate the demonstration; to aid the visual processing, augmented reality tags [Niekum et al., 2014]. Visual observations of people performing actions can be used to infer affordances [Veloso et al., 2005]. Also focusing on the actions a person executes, instead of the objects that are being used, hand manipulations can be classified based on the motions that the hands themselves perform [Worgotter et al., 2013]. The result is a compact ontology of manipulations that is generally applicable. This can be interpreted as an extension of the basic manipulation abilities self exploring robots usually have (poke, push). Fewer than 30 actions types are identified, all fitting in six categories. Classifying hand actions by category is intended to aid recognition, similar to how [Kjellström et al., 2011] are classifying hand motions directly. Graph-models have been applied to scene understanding for affordance modeling in manipulation [Aksoy et al., 2011]. This approach relies on detecting changes in the scene represented as a *relational scene graph* and segmenting the demonstration based on the changes in the graph's topology. Four relation types are defined to represent the scene's layout: (1) touching; (2) overlapping; (3) non-touching; (4) absent. Other work focuses on inferring high-level relations from similar manipulation demonstrations that use simple objects [Cubek et al., 2015];

2. **Sensors on Teacher and Direct Input**, in which the teacher is wearing or using sensors while demonstrating the task, for example by wearing a motion capture suit [Stanton et al., 2012]. This provides precise measurements of the demonstrated activity and poses fewer correspondence difficulties in mapping the demonstration to the model. Alternatively, the

human demonstrator can directly manipulate the robotic arm in what is called kinesthetic teaching [Akgun et al., 2012].

As a teacher, the human can not only be an isolated source of training data from the robot, but instead can interactively provide iterative demonstrations that correct the robot's behavior [Chao et al., 2010]. Socially Guided Machine Learning (SG ML) uses the concept of *human in the loop* to design an affordance learning system in which the robot improves under human guidance [Thomaz and Cakmak, 2009]. As the robot tries to replicate a demonstrated affordance, the human can intervene to correct or stop the demonstration. In an interactive teaching scenario, taking into consideration aspects of human-robot interaction becomes important. For example, gaze is an effective way of non-verbal social communication [Yoshikawa et al., 2006]; gaze can also be used in a teaching scenario to signal uncertainty in the robot's state [Thomaz and Cakmak, 2009].

5.3 Object Substitution within the Task Context

In this section we present our object substitution algorithm, which consists of three steps: generating candidates for the target, extracting context from the HTN, and evaluating the fitness of each candidate within the context.

5.3.1 Generating Candidates

Through candidate generation, we limit the number of concepts which are tested for substitution to a tractable set of likely candidates. In an open-world application that uses a full-sized vocabulary, candidate substitution avoids testing an untractable number of possible substitutions.

We call *target* the word that corresponds to the missing object. In our HTN definitions, this word is the input of a primitive task (pre-defined tasks that are on the lowest abstraction level in the HTN e.g. get, $place$). For each target, we generate a set of words, the *substitution candidates*. We generate candidates by selecting the concepts from ConceptNet which share the same parent with the target, for either of the following relations: *has-property*, *capable-of*, *used-for*. For example, two candidates for $apple$ are $cherry$ and $brick$, because both are also connected to red by *has-property* edges. We selected these relations because they represent affordances, following the proposal of [Awaad et al., 2013]. However, as our candidate annotation described later shows, this process does not define sufficient conditions for generating substitutions reliably. Instead, it provides a good starting point for our model over a less informed candidate generation method such as randomly drawing words from a large vocabulary.

We also experimented with generating candidates by using existing taxonomies, specifically by using sibling concepts from WordNet i.e. concepts which share the same hypernym. For example, the hypernym of $apple.n.01$ is $edible\_fruit.n.01$, which contains candidate hyponym

synsets such as $banana.n.01$ and $fig.n.04$. We found this method to produce a smaller set of candidates, which were generally a subset of the ConceptNet candidates.

5.3.2 Deriving Context from the Task

Our hypothesis is that taking into account the broader context of the other objects and actions in the task is important for evaluating whether a substitution candidate is viable or not. We generate the context, represented as a bag-of-words, from the task definition. Figure 5.2 shows a portion of one of the tasks we used. We choose this approach over a structured representation, for example one that borrows from the HTN structure, in order to not confine the similarity model to the task representation model.

For each task node in the HTN, we extract all words present in the task's name, as well as the object labels used for its input names. We convert all words to lemmas using the NLTK [Loper and Bird, 2002]. For example, the node label *GraspSpoon* is converted into $\{$*grasp, spoon*$\}$. We do not model multi-word concepts because looking up such concepts in ConceptNet and WordNet is not sufficiently reliable.

We implemented five strategies for generating context vocabularies from the HTN. These represent different methods of aggregating words present in the nodes of the HTN tree, varying the breath and level of abstraction of the resulting context with respect to the entire task vocabulary:

- *Node*: all words included in the node of the HTN that contains the substitution target;
- *Task Name*: the node name of the root of the HTN;
- *Root Path*: the union of all node contexts, for all nodes from the node corresponding to the target to the root;
- *Leaves*: the union of all contexts for all leaf nodes;
- *Full*: the union of all nodes in the HTN.

We compare the use of these different contexts in the Results section.

5.3.3 Concept Similarity

Having a number of substitution candidates and contexts, we compare each candidate with the target with respect to a given context using a variety of similarity measures. In expressing these measures, we use the following notation: T for a target concept, C for a candidate substitution concept, and V for the vocabulary that forms the context.

We constructed the measures using numeric similarity metrics derived from semantic networks. First, the WordNet path similarity, which, given two concepts, returns a score normalized in the $[0, 1]$ interval based on the minimum distance between those concepts in the hypernym/hyponym taxonomy [Pedersen et al., 2004]. The WordNet path similarity is a taxonomical

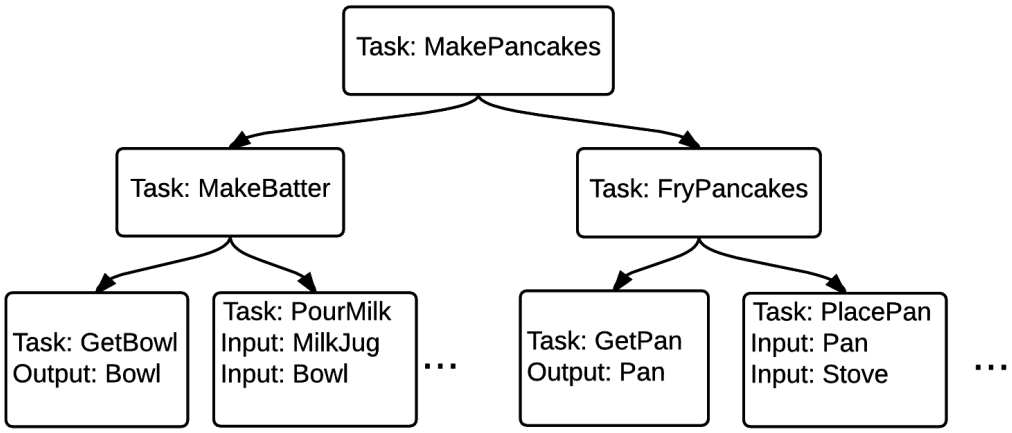


Figure 5.2: Portion of the *set table* task definition, which is one of the HTNs we used in our evaluation.

measure of concept relatedness. Complementary to it, we use ConceptNet Divisi pairwise similarity. It too produces a $[0, 1]$ similarity value, as the result of applying spectral graph methods to estimate how strongly connected the two concepts are in the graph as a whole. The third similarity metric we use is the proportional analogy score obtained using the Semantic Similarity Engine (SSE) [Boteanu and Chernova, 2015], which is also based on ConceptNet. Proportional analogies are commonly phrased as "A is to B as C is to D", or A:B::C:D. We compute the SSE similarity score of the analogy $T : V_i :: C : V_i, V_i \in V$. The SSE score is obtained using the most similar relational path pair linking A to B and C to D, respectively. The relational path pair is identified by exploring and scoring all path pairs of equal length that connect A to B on one side and C to D on the other side. Unlike the other two metrics, the SSE score requires two pairs of concepts instead of just one.

Depending on how we apply these metrics between the target, candidate and the context words, we obtain different global similarity values. We group the similarity metrics used to evaluate candidates by whether they take the context into account or not. In total, there are two baseline attributes and eight context-aware attributes.The baseline we used throughout the experiments only uses attributes that do not take the context into account, and measure only the similarity between the target and candidate:

- WordNet path similarity, or $sim\_wordnet(T, C)$;
- Divisi concept pair similarity, or $sim\_divisi(T, C)$;

We introduce context-aware similarity metrics to complement the baseline measures. We define eight in total:

- The average WordNet similarity between the target and the each of the context words, $sim\_wordnet(T, V_i)$. We compute the equivalent for the candidate as $sim\_wordnet(C, V_i)$. We also account for the difference between these similarity metrics, $delta\_sim\_wordnet(T, C, V_i)$;
- The average Divisi similarity between the target and the context words, $sim\_divisi(T, V_i)$, along with the corresponding metric for the candidate, $sim\_divisi(C, V_i)$. We also account for the difference between these similarity metrics, $delta\_sim\_divisi(T, C, V_i)$;
- The average of non-zero SSE analogy scores for the analogies $sim\_analogy(T : V_i :: C : V_i), V_i \in V$. We ignore all constructions in which there is no relational parallelism between the target and a context word on one side, and the candidate and a context word on the other. We also compute the proportion of context words that do not form analogical relations as the $no\_analogy$ metric.

The following section describes how the above metrics are used by a classifier to predict successful object substitutions.

5.4 Experimental Setup

Our evaluation focuses on two aspects: (1) investigating the benefit of using context information in addition to the baseline similarity metrics, and (2) analyzing the resilience our method has to variations in the context constituency. We defined the evaluation as a series of supervised classification problems, in which we trained supervised models using data derived from nine HTNs which we defined. The datasets we used for training classifiers contain an instance for each (task, target, candidate) tuple, for which we computed the context-average similarity scores according to each experiment. We used only these metrics to predict whether a substitution is valid or not, ignoring nominal attributes such as the task or the target.

We will now describe the tasks from a vocabulary standpoint. Table 5.1 contains word information for each task. The average set overlap between the vocabulary of each task (Jaccard index) is 8.9%. The average overlap of the sets of targets between tasks is 3.2%. Thus, the tasks are mostly orthogonal in the vocabularies they use, with the majority of the overlap being in the names of the primitive actions, all related to manipulation.

In selecting substitution targets, we assume that only objects directly manipulated can be substituted, which in our representation reside on leaf nodes. These contain a primitive action (e.g. get) and an input (e.g. cup); we consider all these inputs substitution targets. Our set of tasks contained 37 substitution targets ($M = 5.9; SD = 1.5$). Starting from these targets, we generated a total of 1832 substitution candidates ($M = 203.5; SD = 111.8$), which greatly exceeds the number of targets, but of these only 6.16% (113) were annotated as valid ($M = 14.7; SD = 10.4$), according to the method presented in the next section.

Table 5.1: Vocabulary sizes for each task, the valid candidates counts are accoding to the expert annotation.

Task	Total words	Targets	Candidates	Valid cand.
Eat Soup	8	4	156	6
Make Pancakes	15	7	94	8
Set the Table	13	5	302	10
Cook Soup	17	6	307	30
Make Hot Drink	26	9	173	8
Rotate Car Tires	15	6	44	0
Make Fruit Basket	8	5	365	31
Wipe Table	7	4	83	17
Pack School Bag	10	7	308	23

5.4.1 Labeling Valid Substitutions

To establish the ground truth for classification, we had all candidates labeled as suitable or not by two experts familiar with the task definition and the scope of this work. For 31 candidates out of 1832 in total, the experts disagreed, in which case the final label was decided after discussion.

To verify the expert labels, we selected 228 of the candidates to be annotated on the Crowdflower[1] platform. The survey included a brief background description, the name of the task, and the target and candidate objects. We noted the final annotation result as the majority vote from at least 5 participants (16.1 on average). We selected workers only from English-speaking countries (111 workers in total).

The expert and crowd annotations match for 80.7% of the substitution candidates. The experts rejected 35 candidates that the crowd accepted: e.g. replacing a *knife* with a *spoon* in the *cook soup* task, or *glass* with *chalk* for the *set the table* task. On the other hand, there were 9 instances in which the crowd accepted a substitution that the experts rejected, for example the crowd did not consider that a *knife* can be replaced with a *cleaver* for the same task. We attribute these distinctions to personal preference, and also to crowdsourcing noise [Hsueh et al., 2009]. Overall, following this survey we concluded that the expert annotation is representative of what a common person might accept in terms of substitution.

5.4.2 Performance Metrics and Classification Approach

We tracked two metrics for evaluating the classification accuracy. First, the global accuracy of the classifier, i.e. on both valid and invalid substitutions. Second, we report the rate of success on valid substitutions, since this value most accurately represents the user-observable performance of our method:

$$\frac{valid\ substitutions}{valid\ substitutions + false\ positives + false\ negatives}$$

Having established our metrics, the first step was to choose a classification model. We compared the performance of our method for *full* context metrics with the baseline over six classifi-

[1] http://www.crowdflower.com

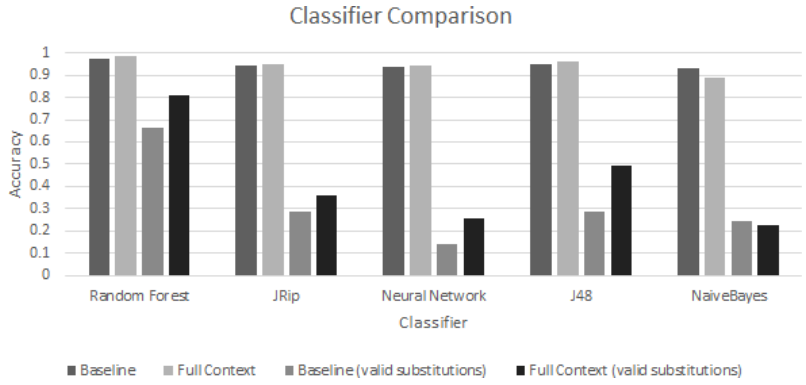


Figure 5.3: Relative classification performance of different classification methods. Following these results, we chose the Random Forest classifiers for our experiments, as it outperformed the others.

cation methods[2]. With 95% confidence according to a paired t-test, the *Random Forest Classifier* outperforms the other methods for both the baseline and for our method (the second best is *J4.8 Decision Tree*). Figure 5.3 shows the relative performance of all classifiers we tested. In addition, the context-aware method outperforms the baseline with 95% confidence for all models except *Naive Bayes*. Therefore we chose the *Random Forest* classifier for the rest of our experiments.

5.5 Substitution Prediction Results

In this section we evaluate the performance and scalability of our approach using the setup presented above. We present results on different strategies for constructing context from the task vocabulary, showing that including all words from the task outperforms the baseline and other approaches. We then investigate the performance of our method relative to each semantic resource and similarity metric, and to omissions in the task vocabulary. Such omissions may arise from tasks that are not fully annotated with human-readable labels, or if some unusual words are not present in the semantic network. Next, we present two experiments on the transfer-learning potential and generality of the models produced by our method. Finally, we give details on the setup of the tasks we implemented on the physical robot.

5.5.1 Evaluation of Context-creation Strategies

We compared the methods for generating context vocabularies from the HTN as separate approaches of extracting bag-of-words contexts from the task definition. Figure 5.4 shows the classification 10-fold cross validation performance over all tasks together. This result justifies our

[2]For all classification experiments presented here, we used Weka 3.7.12 [Hall et al., 2009]

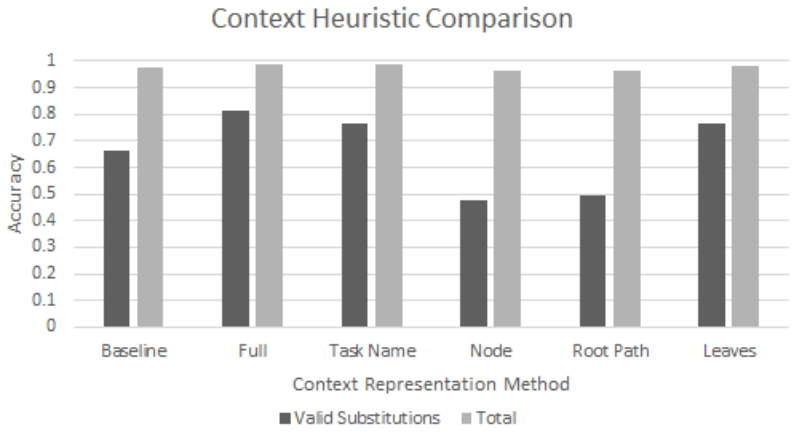


Figure 5.4: Relative classification performance for different methods of generating context vocabularies from the HTN and the task-agnostic baseline.

focus on valid substitutions, since all methods are indistinguishable in overall performance, but there are significant differences when taking into account only valid substitutions.

The full context methods outperforms all other methods (81% valid-only accuracy, compared to 66% for the baseline). According to our hypothesis, context should be beneficial, but the results indicate that without higher-level concepts and a larger number of words in the context, this is not the case. The $node$ and $root - path$ contexts do not perform as well, possibly because of the similarity noise introduced by having a small number of words in the context, out of which most are generic manipulation actions. However, even in this case we can observe that adding higher-level concepts benefits the classification accuracy, since $root - path$ outperforms $node$.

The *rotate tires* task has no valid substitution candidates. For this task, the full-context classifier outputs no false positives and clearly outperforms the baseline, which attempts to replace the nuts fix on wheels with various types of fruit – without context information, the word senses would need to be manually annotated in the task, which would be a formidable challenge to do consistently and reliably.

5.5.2 Resource and Attribute Importance

We investigated the relative importance of WordNet and ConceptNet, along with their corresponding similarity metrics, in characterizing substitution quality. Table 5.2 shows the classification performance when removing some attributes from all instances in the full context dataset. We divide these results by resource type and report the 10-fold classification accuracy. The ConceptNet set of attributes contains all Divisi and analogy scores.

We also ranked individual features by their information gain[3]. Table 5.3 shows the resulting

[3]Using the *InfoGainAttributeEval* class from Weka

Table 5.2: Classification performance for using similarity metrics from only WordNet, only ConceptNet, or both, for the full context classifier. Values reported for a random forest classifier over 10-fold cross-validation.

WordNet	ConceptNet	Global Accuracy	Valid Substitution Accuracy
•		95%	72%
	•	92%	59%
•	•	95.7%	76%

Table 5.3: Rank scores for the similarity metrics.

Similarity Measure	Ranker Score
$sim\_wordnet(T, C)$	0.1633
$sim\_divisi(T, C)$	0.1591
$sim\_divisi(T, V)$	0.1095
$sim\_divisi(C, V)$	0.0907
$sim\_analogy(T : V :: C : V)$	0.0803
$no\_analogy$	0.0795
$sim\_wordnet(T, V)$	0.0781
$sim\_wordnet(C, V)$	0.0605
$delta\_sim\_wordnet(T, C, V)$	0
$delta\_sim\_divisi(T, C, V)$	0

weights, higher values mean more importance. We observe that the baseline features are dominant, which indicate that the direct similarity between the target and candidate are a good starting point; it is reasonable to assume that similar objects will perform similarly. However, all other types of attributes contribute as well. In particular, we notice that the Divisi similarity with respect to the context, for both the target and the substitute, is more informative than the WordNet distance, which shows that a richer semantic net is preferable for relating the context. As expected, the delta features do not add any information from an entropy standpoint. However, we observed that some of the classification methods perform better with these attributes.

5.5.3 Context Vocabulary Sensitivity

We investigated how resilient our method is to task definitions that are not well populated by words. Such tasks may arise, for example, from automatically constructing task trees from human demonstration, in which case the root task and leaf nodes would be well populated, but the automatically generated abstractions of the intermediate layers may not contain actual words at all [Mohseni-Kabir et al., 2015].

First, we trained a classifier on the entire dataset. Then we generated a test dataset by sampling sets of words from the full context of each task, computed the context scores per candidate with respect to the sample, and used these values to classify. We evaluated sampling different fixed percentages from the task vocabulary, excluding the target. For each sample size, we use at most 30 samples per task.

We compare this performance to the average classification accuracy of the baseline using 10-

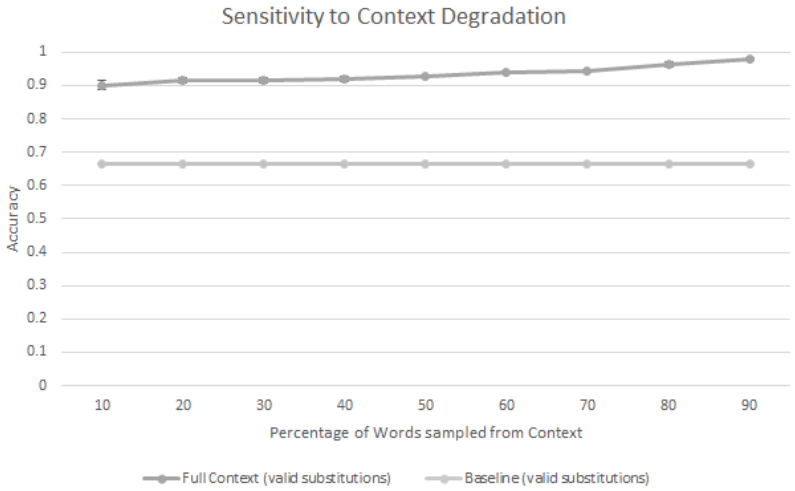


Figure 5.5: Mean and std. deviation of valid-only accuracy accuracy of full-context classifier for a varying percentage of words sampled from the context, compared to the 10-fold cross validation accuracy of the baseline.

fold cross validation. We do so because, if we were to apply the same approach for the baseline arguments, the classifier would have been trained and tested on the same dataset since the baseline attribute values do not change depending on context. By using the 10-fold cross validation performance of the baseline we get a more realistic estimate of its performance. Figures 5.5 shows this performance for a given percentage of the total context size of a task. These results show that prediction models trained using our method are resilient to changes in the task vocabulary, maintaining good performance on tasks defined within the same domain.

5.5.4 Generality of Substitution Models

In this section we explore the potential for generalization our method has. First, we show that, when training models separately for each task, the resulting average performance is lower than that of the model trained over all tasks together. This may indicate cross-domain transfer, which we explore further by predicting substitution quality in novel tasks through leave-one-task-out training.

Furthermore, we observed that in some cases, the substitutions generated by our method are instantiations or generalizations of the task's definition. For the sake of brevity, we will mention a few examples: (1) in the *cook soup* task, the object *vegetable* was replaced specific vegetables (*pepper, peas*), among other soup ingredients (e.g. *chicken*); (2) in the *fruit basket* task, *apple* was replaced with *fruit*, which is an abstraction; (3) in the *wipe table* task, the *table* was replaced with *board, plate*, and *floor*, which are other surfaces that could be wiped.

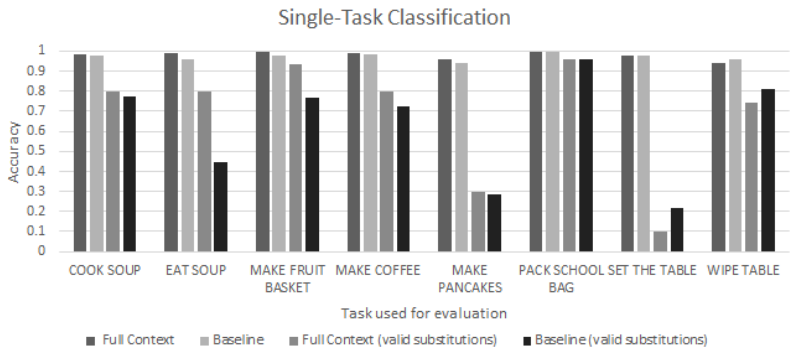


Figure 5.6: Classification performance for single task learning, using 10-fold cross validation.

Single-Task Learning

We evaluated the full-context method for single-task learning by training and testing separate classifiers for each task. We did not include the *rotate tires* task in this experiment since none of its candidates were annotated as valid substitutions. Figure 5.6 shows the 10-fold cross validation performance for using data per individual task for the full-context method and the baseline.

The mean valid-only accuracy, weighted by the number of candidates per task, is 77.4% for the full-context method and for 69.8% the baseline. These values show a drop in overall performance, which may be the result of less data per task. This decrease in performance can also indicate that there is some potential for cross-task learning.

Substitution in Novel Tasks

Finally, to estimate the performance of our approach on novel tasks, we trained a model on all but one of the tasks and then tested it on the remaining task. As shown in the vocabulary section, tasks have mostly orthogonal vocabularies. Thus, the task we hold out mostly contains novel concepts for which similarity has not been measured directly during training. We found that for this experiment only, the *Random Forest* classifier is outperformed by a small margin by *J4.8 Decision Tree*. Figure 5.7 shows performance of the full-context and the baseline methods for each task that was used for testing.

We notice that the context method is tied with the baseline on most tasks. This is because context is not directly translatable between very different domains, which is to be expected. However, the *eat soup* and *cook soup* tasks exhibit better performance, showing that transfer is occurring. The *tire rotation* task has very low accuracy for both methods. This is because, lacking word sense information, both classifiers accept the substitutions that were learned before, despite words having a different sense in the novel task.

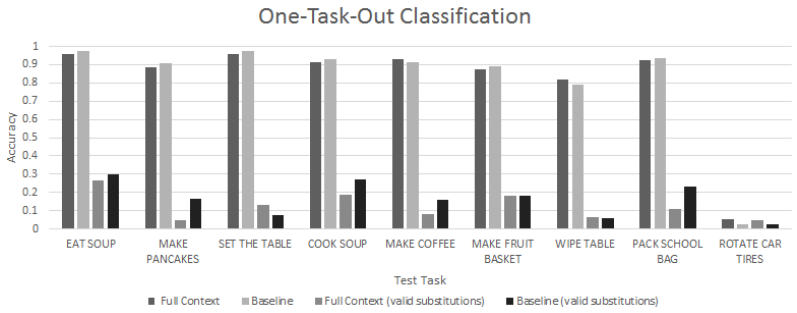


Figure 5.7: Classification performance when trained on all tasks but one and tested on the one left out.

5.6 Inferring Substitution Characteristics

The object substitution problem consists of identifying replacements for objects that are not found during task execution, such that the plan can be repaired and execution can resume. In the previous sections of this chapter we presented a method for constructing classification models that will produce valid solutions for the object substitution problem. We framed our approach as a binary classification problem, in which the model discriminates between valid and invalid substitution candidates using a number of numeric similarity metrics that compare a candidate with the target in a given context. We use supervised learning to construct this model. For this process to be feasible, we also described a candidate annotation method. We will now present an approach for abstracting these substitutions that leverages the analogy relational paths SSE generates as part of computing one of the similarity metrics we use for classification.

As presented in Chapter 4, SSE searches for the highest similarity path pair (HSSP) between two pairs of words, $A : B$ and $C : D$, given as input. When evaluating object substitution, we reformulate the proportional analogy problem, $A : B :: C : D$, to evaluate the relational similarity that the target and candidate share with each context word, by forming the analogy $T : V_i :: C : V_i$. To create the dataset that we will use to infer general substitution characteristics, we will form a set of relations that are in common between the two sides of the HSSP for each respective segment, and the unify these relations into a single set. For example, in Figure 4.5, the final resulting set is $\{$*Has-Property, Is-A* $\}$.

Using these edges derived from the HSSP, we generated a dataset with an instance per (task, target, candidate, context word) tuple. The instances include all similarity metrics shown in Section 5.3.3 and 48 binary attributes, each corresponding to a ConceptNet edge. The binary attributes are set if the corresponding edge is present in the HSSP according to the method from the previous paragraph. We also included the substitution validity annotation. Thus, there are three key differences between this dataset and the one we used for predicting substitutions: (1) each instance

contains similarity metrics corresponding to a single context word, instead of averages per context; (2) in addition to the metrics, each instance contains edge information from the HSSP; (3) we only included the (task, target, candidate, context word) tuples for which there existed a non-zero HSSP.

Using these attributes, we define a new classification problem: predict if a substitution is valid starting from only a single context word, using both edge and similarity information. The total dataset contained 657 positive and 6514 negative examples. We investigated two scenarios: within-task learning and task-independent learning by including or excluding the task attribute from the model. For both experiments, we ignored the target, candidate and context word attributes.

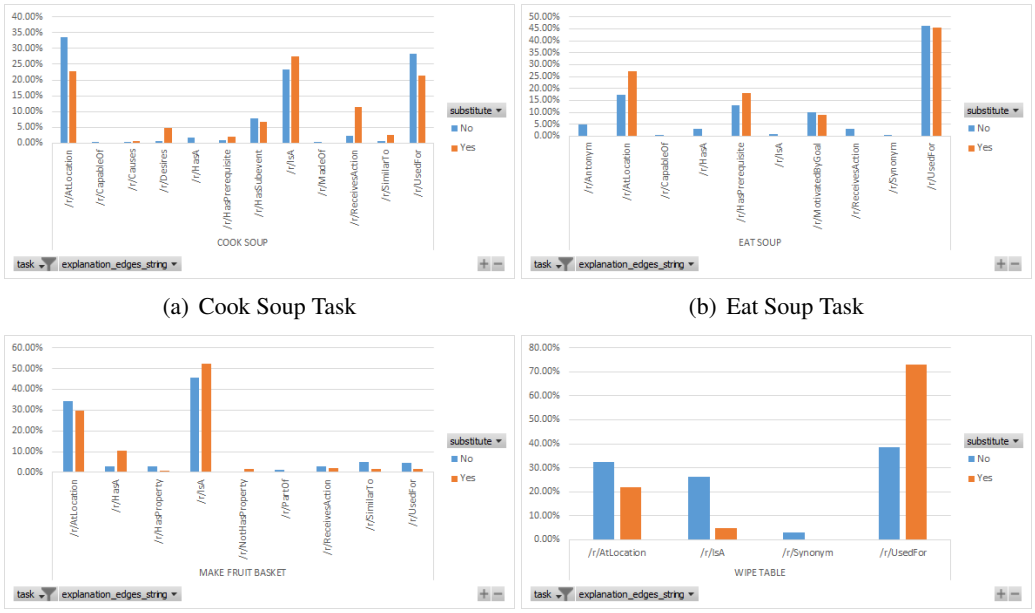
To compare the discriminating power the edge information has relative to the similarity metrics, we first present the results of the information gain ranker in Table 5.4. We observe that some edges receive a non-zero weight, however all similarity metrics receive a higher weight than any of the edges. This is to be expected, since non-zero edge information is sparse, whereas the similarity metrics have real values with no missing values. We also constructed a random forest classifier all combinations of the following binary conditions: using/ignoring task information; using/ignoring edge information. These results are presented in Table 5.5. We notice that including task information (i.e. the task attribute) improves the classification performance, particularly when the similarity attributes are removed. This shows that (1) the similarity metrics are dominant and essential for performance, in concordance with the information-ranker output; (2) that using edge information for inferring suitable candidates is task dependent.

We bring further evidence that edge information can be used for prediction in a task-dependent manner by comparing the percentage of instances that contain a given edge type between valid and invalid substitutions. Figure 5.8 shows edge histogram comparisons between valid and invalid substitutions for a number of tasks, plotted using the same dataset. We notice that certain edges are over-represented in the valid substitution set, showing that these edges correlated with valid substitutions. We will enumerate some examples:

- Wipe Table Task: the *Used-For* edge is present in 72% of the valid substitution candidates, indicating that the purpose of an object as a tool is a defining criteria in substitutions for this task;

- Eat Soup Task: the *At-Location* is mode common for valid substitutions, showing that co-located objects are likely substitutes for the objects used in this task. This is a reasonable conclusion since pots, dishes and silverware usually are found in the same area. We note that the task does not specifically encode the word "kitchen," or words related to it.

- Fruit Basket Task: the *Is-A* edge is slightly more common for valid substitutes, showing that objects of the same type (i.e. fruit) are interchangeable for this task.



(a) Cook Soup Task

(b) Eat Soup Task

(c) Fruit Basket Task

(d) Wipe Table Task

Figure 5.8: Edge histograms for individual tasks, showing the differences in edge distribution between valid and invalid substitutions.

5.7 Physical Robot Implementation

We implemented two of the tasks on a physical robot: *pack schoolbag* and *wipe table*. The supplementary video shows a robot carrying out the tasks in a lab setup that resembles a household environment. The robot is using substitutions generated and learned using the *full-context* classifier. In addition to this, the video shows how user feedback can be included in the system. When performing a substitution for the first time on a task, the robot asks the user for confirmation, to which the user can respond by either accepting or rejecting the substitution. For rejection, the user has two options which distinguish between rejections based on practical versus preferential reasons. For future work, we plan to use this feedback to refine the substitution model.

When proposing substitutions, the robot first generated and validated them at a word level, after which it verified if these were available in the environment. The surfaces on which objects were found were known in advance, but our method is compatible with probabilistic retrieval solutions such as semantic mapping. The navigation map was learned in advance in order to aid localization. The robot autonomously navigated, recognized objects and grasped them using pre-trained models. The user interaction is presented for exemplification. While we did not use voice recognition to interpret the user's answers, distinguishing between three possible answers is easily achievable using off-the-shelf software. Figure 5.1 shows a still from the video.

Table 5.4: Information ranker scores for attributes, including similarity metrics, task label and edge annotation.

Attribute	Score
$sim\_divisi(T, C)$	0.197452
$sim\_wordnet(T, C)$	0.076416
$sim\_divisi(C, V)$	0.033216
task	0.027482
$sim\_wordnet(C, V)$	0.020657
$sim\_divisi(T, V)$	0.017102
$sim\_analogy(T : V :: C : V)$	0.008095
$sim\_wordnet(T, V)$	0.006446
edge-ReceivesAction	0.004719
edge-Desires	0.002689
edge-IsA	0.002356
edge-HasContext	0.001926
edge-AtLocation	0.001634
edge-HasA	0.001586
edge-NotHasProperty	0.000887
edge-MadeOf	0.0006

Table 5.5: Classification performance showing the changes in performance as attributes related to task and edge information are used or removed.

Task	Similarity Metrics	Edge Information	Global Accuracy	Positive Only Accuracy
•			90.83%	0.0%
	•		97.47%	76.02%
		•	90.85%	2.67%
•		•	90.96%	5.81%
	•	•	97.18%	73.49%
•	•	•	97.28%	73.96%

5.8 Conclusion

We presented a novel method for generating valid substitutions for open-world robot tasks. Our approach uses contexts derived from the task description to compare the original object with the substitution candidates. To evaluate similarity, we use two semantic networks and a similarity metrics, including analogical similarity. In our evaluation we showed that our method outperforms a baseline that is context-agnostic. Our method is robust to words missing from the context vocabulary. We plan to develop this work into a system that can execute the substitutions on a physical robotic manipulator by initially learning how to manipulate substitutes from human demonstration.

Object substitution could also be complemented by methods focusing on physical affordancess suchas AfNet [Varadarajan and Vincze, 2012, Varadarajan and Vincze, 2013]. These databases contain information such as shape and material properties. In conjunction with object substitution, evaluating substitution candidates from a physical standpoint should further refine the system's output. Informally stated, object substitution produces judgments whether a substitution *could* be allowed, while reasoning over physical affordances would decide if the substitution *can* be

performed.

The object substitution framework incorporates SSE as one of its similarity metrics. In future work, SSE will also provide explanations for the system's behavior, helping users understand the system's state and outputs. We hypothesize that natural language explanations will be preferable over directly inspecting the system's state because through natural language salient information can be conveyed to both expert and naive users. The main challenge in producing such explanations is the user interface, which would have to succinctly present justifications derived from large contexts. In addition, producing explanations will allow users to customize the criteria that the robot would use when evaluating substitutions, as an extension of the trends we underlined in Figure 5.8.

We stated object substitution as the induction step for more complex plan repair methods, such as identifying alternate recipes. In order to enable higher-level replacements within a task, object substitution could play a key role in validating the objects that such recipes would use. However, such an extension would require two additions: (1) a subtask library, sufficiently annotated to allow for exact prerequisite-effect evaluations, and (2) a succinct evaluation method that would be used to benchmark such a system. Currently we are not aware of significant task databases that are designed for robotic applications in human environments. However, we consider our current approach of presenting evaluators with a brief narrative of the task scenario and the substitution proposal sufficiently descriptive to allow subtask-level substitutions to be evaluated.

CONCLUSION

Computational models of similarity enable a broad range of artificial intelligence applications. In this work we contributed two similarity models which build on the idea of context-graphs. Defining and using context within an application is one of the key challenges in artificial intelligence. Through *context-graphs*, we model context in a flexible and problem-independent manner, from which we construct two complementary similarity models: (1) topic modeling as a form of general similarity; (2) proportional analogical similarity, a form of relational similarity.

We use these models in three applications: to generate discussion suggestions for a pre-literacy primer, which helps parents better conversations with their children, and to explain broad-vocabulary analogy questions to humans; to generate context-aware object substitutions which enable more robust robot plan execution. In addition to context-graphs, this work has the overarching theme of providing interpretable feedback and output in the form of natural language. Each application produces human-understandable outputs: discussion suggestion prompts, analogy explanations, and, as shown in the attached video, object substitution validation prompts.

We validated the contributions of this thesis through a variety of methods: performance metrics such as classification accuracy, overlap with human responses, direct human evaluation of the system's output through surveys, and a user study for the end-to-end effectiveness of the application. Through all these evaluations, we showed that our contributions achieve good high-confidence performance without sacrificing the interpretability of the output.

Future Work

We will mention a few directions in which the contributions of this work could be developed in the near future. First, our topic modeling method allows for words to be part of multiple topics, consistent with existing literature, which opens the question of how similar two topics are. We briefly explored this question as part of a hybrid recommendation engine, which used both previous ratings and inter-user topic overlap to predict user ratings [Boteanu and Chernova, 2013b]. In this application, we compared extent to which topics overlap through a metric that took into account both the proportion of words two topics have in common, as well as the similarity between their difference. In the future, such topic overlap metrics could enable better topic tracking over long discussions by revealing how topics succeed in a discussion or narrative. This direction would be

particularly interesting to analyze from a structural standpoint by using explanations derived from the edges of the semantic network.

Second, with respect to generating explanations, SSE provides multiple interpretations of an analogy, depending on the length of each explanation path. While we showed that alternate explanations can offer different perspectives on an analogy, it would be of interest in future work to explore relational equivalence in semantic networks. This direction would be particularly motivated by Figure 4.7, which shows that a significant proportion of the explanations in which edges were randomized were accepted by respondents. This shows that some relations can be considered equivalent in the context of a specific problem. The next step would be to show relational equivalence between paths of different lengths, which would relax one of the main assumptions that SSE shares with alignment-based theory, that one-to-one correspondences are to be identified.

While SSE is designed to answer analogy questions reliably and to provide justifications for the answer choice, it currently lacks the coverage of state-of-the-art multiple-choice analogy answering methods such as Latent Relational Analysis (LRA) [Turney and Littman, 2005]. As we discuss in Chapter 4, SSE is primarily limited by the lack of information on unusual or generally difficult to comprehend words. In addition to directly expanding the semantic network to include more information, which is a non-trivial task, a corpus could be leveraged to obtain additional information. We believe that SSE and LRA could be integrated in the future in a complementary manner, such that whenever SSE is unable to answer a question due to missing information, LRA is used to select an answer option. Furthermore, the answers selected by LRA could be used to disambiguate answer choices that SSE scores similarly.

Our object substitution model operates at a language level. Its output needs to be grounded in the physical environment, which we do not address in this work. We can view the problem of associating a concept in the task with some desired robot behavior as a number of connected problems: the word must match an object that the robot can recognize, and the robot must know the correct behavior needed to use the object for the task. For the first problem, assuming that the recognition system produces natural language labels, the problem could be solved using synonym lists (to allow for different names of the same object), and with possible input from work on entity resolution [Singla and Domingos, 2006]. Learning manipulation primitives is a broad problem, ranging from grasping for pick-and-place tasks [Kent et al., 2014] to assembly tasks [Niekum et al., 2013] and beyond (for example, playing table tennis can be considered to include the subtask of manipulating the paddle [Mulling et al., 2013]). Either through demonstration or through reinforcement learning, the robot will likely need to make adjustments between different objects even if it has a model for manipulation. However, currently robots do not have the necessary feedback and perception to be able to accurately determine if a certain motion execution is going as planned, if it needs to retry, or if there is little chance of success and help from a human is needed.

More broadly, we anticipate that closing the feedback loop using human and machine interpretable feedback will enable artificial intelligence systems to iterate and improve using human

validation. In Chapter 4 we show that semantic information can be translated directly to natural language, allowing for naive users to understand the reason behind a certain analogy answer without knowing the internal representation. In Chapter 5 we explored the relational differences between valid and invalid substitution candidates, and how these respectively related to the task context. In a closed-loop scenario, these observations will be used to allow a user to better understand the system's state, and also to make better predictions based on the simple ternary feedback we demonstrate in this work. For example, the user could directly correct specific assertions from the semantic network which are the cause of erroneous results. In the long term, the robot could learn to distinguish more informative relational structures within the context. We hypothesize that a model of similarity between relations, as mentioned above, would be particularly necessary to allow selecting only relevant edges from the context.

Finally, in defining our ultimate goal of a closed feedback loop between a human and an agent, we should mention existing efforts in lifelong robot learning [Bou Ammar et al., 2015, Ruvolo and Eaton, 2013]. In this thesis we showed empirically that SSE produces reliable answers and justifications for analogies, and that it does not produce an output if data is insufficient. We believe similar features will be key in lifelong learning applications, and also that the confidence of each answers should be quantified in the future. To some extent, our object substitution framework achieves this goal by taking into account the proportion of context words that did not form an analogy with the target and candidate. However, we believe that putting a confidence bound on qualitative results such that it prevents propagating errors in the long term is an open problem.

BIBLIOGRAPHY

[Abney, 1992] Abney, S. P. (1992). *Parsing by chunks*. Springer.

[Aihe and Gonzalez, 2015] Aihe, D. O. and Gonzalez, A. J. (2015). Correcting flawed expert knowledge through reinforcement learning. *Expert Systems with Applications*, 42(17):6457–6471.

[Akgun et al., 2012] Akgun, B., Cakmak, M., Yoo, J. W., and Thomaz, A. L. (2012). Trajectories and keyframes for kinesthetic teaching: A human-robot interaction perspective. In *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, pages 391–398. ACM.

[Aksoy et al., 2011] Aksoy, E. E., Abramov, A., Dörr, J., Ning, K., Dellen, B., and Wörgötter, F. (2011). Learning the semantics of object–action relations by observation. *The International Journal of Robotics Research*, 30(10):1229–1249.

[Argall et al., 2009] Argall, B. D., Chernova, S., Veloso, M., and Browning, B. (2009). A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483.

[Arnold and Whitehurst, 1994] Arnold, D. S. and Whitehurst, G. J. (1994). Accelerating language development through picture book reading: A summary of dialogic reading and its effect.

[Arvola et al., 2011] Arvola, P., Kekäläinen, J., and Junkkari, M. (2011). Contextualization models for xml retrieval. *Information Processing & Management*, 47(5):762–776.

[Awaad et al., 2013] Awaad, I., Kraetzschmar, G. K., and Hertzberg, J. (2013). Affordance-based reasoning in robot task planning. In *Planning and Robotics (PlanRob) Workshop ICAPS-2013*.

[Baillargeon, 1991] Baillargeon, R. (1991). Reasoning about the height and location of a hidden object in 4.5-and 6.5-month-old infants. *Cognition*, 38(1):13–42.

[Baillargeon et al., 1985] Baillargeon, R., Spelke, E. S., and Wasserman, S. (1985). Object permanence in five-month-old infants. *Cognition*, 20(3):191–208.

[Baker et al., 1998] Baker, C. F., Fillmore, C. J., and Lowe, J. B. (1998). The berkeley framenet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational*

Linguistics and 17th International Conference on Computational Linguistics-Volume 1, pages 86–90. Association for Computational Linguistics.

[Bazire and Brézillon, 2005] Bazire, M. and Brézillon, P. (2005). Understanding context before using it. In *Modeling and using context*, pages 29–40. Springer.

[Beltagy et al., 2014] Beltagy, I., Erk, K., and Mooney, R. (2014). Probabilistic soft logic for semantic textual similarity. *Proceedings of Association for Computational Linguistics (ACL-14)*.

[Bennett and Lanning, 2007] Bennett, J. and Lanning, S. (2007). The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, page 35.

[Bidot et al., 2008] Bidot, J., Schattenberg, B., and Biundo, S. (2008). Plan repair in hybrid planning. In *KI 2008: Advances in Artificial Intelligence*, pages 169–176. Springer.

[Bird, 2006] Bird, S. (2006). Nltk: the natural language toolkit. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, pages 69–72. Association for Computational Linguistics.

[Blei et al., 2003] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.

[Bolovinou et al., 2013] Bolovinou, A., Pratikakis, I., and Perantonis, S. (2013). Bag of spatio-visual words for context inference in scene classification. *Pattern Recognition*, 46(3):1039–1053.

[Bonet and Geffner, 2001] Bonet, B. and Geffner, H. (2001). Planning as heuristic search. *Artificial Intelligence*, 129(1):5–33.

[Borrajo et al., 2015] Borrajo, D., Roubíčková, A., and Serina, I. (2015). Progress in case-based planning. *ACM Computing Surveys (CSUR)*, 47(2):35.

[Boteanu and Chernova, 2013a] Boteanu, A. and Chernova, S. (2013a). Modeling discussion topics in interactions with a tablet reading primer. In *Proceedings of the 2013 international conference on Intelligent user interfaces*, pages 75–84. ACM.

[Boteanu and Chernova, 2013b] Boteanu, A. and Chernova, S. (2013b). Unsupervised rating prediction based on local and global semantic models. In *2013 AAAI Fall Symposium Series*.

[Boteanu and Chernova, 2015] Boteanu, A. and Chernova, S. (2015). Solving and explaining analogy questions using semantic networks. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

[Botterill et al., 2008] Botterill, T., Mills, S., and Green, R. (2008). Speeded-up bag-of-words algorithm for robot localisation through scene recognition. In *Image and Vision Computing New Zealand, 2008. IVCNZ 2008. 23rd International Conference*, pages 1–6. IEEE.

[Bou Ammar et al., 2015] Bou Ammar, H., Eaton, E., Luna, J. M., and Ruvolo, P. (2015). Autonomous cross-domain knowledge transfer in lifelong policy gradient reinforcement learning. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-15)*.

[Bouchard et al., 2006] Bouchard, B., Bouzouane, A., and Giroux, S. (2006). A smart home agent for plan recognition. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, AAMAS '06, pages 320–322, New York, NY, USA. ACM.

[Brézillon, 1999] Brézillon, P. (1999). Context in artificial intelligence: I. a survey of the literature. *Computers and artificial intelligence*, 18:321–340.

[Brézillon, 2014] Brézillon, P. (2014). A context-centered architecture for intelligent assistant systems. In *Innovations in Intelligent Machines-4*, pages 103–127. Springer.

[Burns et al., 1999] Burns, M., Griffin, P., and Snows, C. (1999). Starting out right. a guide promoting children's reading success. wdc.

[Bus et al., 1995] Bus, A. G., Van Ijzendoorn, M. H., and Pellegrini, A. D. (1995). Joint book reading makes for success in learning to read: A meta-analysis on intergenerational transmission of literacy. *Review of educational research*, 65(1):1–21.

[Call and Carpenter, 2002] Call, J. and Carpenter, M. (2002). Three sources of information in social learning. *Imitation in animals and artifacts*, pages 211–228.

[Cambria et al., 2010] Cambria, E., Speer, R., Havasi, C., and Hussain, A. (2010). Senticnet: A publicly available semantic resource for opinion mining. In *AAAI Fall Symposium: Commonsense Knowledge*, volume 10, page 02.

[Chang and Breazeal, 2011] Chang, A. and Breazeal, C. (2011). Tinkrbook: Shared reading interfaces for storytelling. In *Proceedings of the 10th International Conference on Interaction Design and Children*.

[Chang and Forbus, 2012] Chang, M. D. and Forbus, K. D. (2012). Using quantitative information to improve analogical matching between sketches. *Innovative Applications of Artificial Intelligence (IAAI). Toronto, Canada*.

[Chang et al., 2007] Chang, S.-F., Ellis, D., Jiang, W., Lee, K., Yanagawa, A., Loui, A. C., and Luo, J. (2007). Large-scale multimodal semantic concept detection for consumer video. In *Proceedings of the international workshop on Workshop on multimedia information retrieval*, pages 255–264. ACM.

[Chao et al., 2010] Chao, C., Cakmak, M., and Thomaz, A. L. (2010). Transparent active learning for robots. In *Human-Robot Interaction (HRI), 2010 5th ACM/IEEE International Conference on*, pages 317–324. IEEE.

[Chen and Palmer, 2009] Chen, J. and Palmer, M. S. (2009). Improving english verb sense disambiguation performance with linguistically motivated features and clear sense distinction boundaries. *Language Resources and Evaluation*, 43(2):181–208.

[Cho et al., 2007] Cho, Y.-R., Hwang, W., Ramanathan, M., and Zhang, A. (2007). Semantic integration to identify overlapping functional modules in protein interaction networks. *BMC bioinformatics*, 8(1):265.

[Choi et al., 2012] Choi, M. J., Torralba, A., and Willsky, A. S. (2012). A tree-based context model for object recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(2):240–252.

[Cimatti et al., 1998] Cimatti, A., Roveri, M., and Traverso, P. (1998). Strong planning in non-deterministic domains via model checking. In *AIPS*, volume 98, pages 36–43.

[Cox et al., 2005] Cox, M. T., Munoz-Avila, H., and Bergmann, R. (2005). Case-based planning. *The Knowledge Engineering Review*, 20(03):283–287.

[Craw et al., 2006] Craw, S., Wiratunga, N., and Rowe, R. C. (2006). Learning adaptation knowledge to improve case-based reasoning. *Artificial Intelligence*, 170(16):1175–1192.

[Croft et al., 2010] Croft, W. B., Metzler, D., and Strohman, T. (2010). *Search engines: Information retrieval in practice*. Addison-Wesley Reading.

[Cubek et al., 2015] Cubek, R., Ertel, W., and Palm, G. (2015). High-level learning from demonstration with conceptual spaces and subspace clustering. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 2592–2597. IEEE.

[Cunningham et al., 2003] Cunningham, P., Nowlan, N., Delany, S. J., and Haahr, M. (2003). A case-based approach to spam filtering that can track concept drift. In *The ICCBR*, volume 3, pages 03–2003.

[De Raedt and Kersting, 2008] De Raedt, L. and Kersting, K. (2008). *Probabilistic inductive logic programming*. Springer.

[Deerwester et al., 1990] Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., and Harshman, R. A. (1990). Indexing by latent semantic analysis. *JASIS*, 41(6):391–407.

[Doan et al., 2004] Doan, A., Madhavan, J., Domingos, P., and Halevy, A. (2004). Ontology matching: A machine learning approach. In *Handbook on ontologies*, pages 385–403. Springer.

[Dumais, 2004] Dumais, S. T. (2004). Latent semantic analysis. *Annual review of information science and technology*, 38(1):188–230.

[Duursma et al., 2008] Duursma, E. v., Augustyn, M., and Zuckerman, B. (2008). Reading aloud to children: the evidence. *Archives of disease in childhood*, 93(7):554–557.

[Ehrenmann et al., 2002] Ehrenmann, M., Zollner, R., Rogalla, O., and Dillmann, R. (2002). Programming service tasks in household environments by human demonstration. In *Robot and Human Interactive Communication, 2002. Proceedings. 11th IEEE International Workshop on*, pages 460–467. IEEE.

[Eisenstein and Barzilay, 2008] Eisenstein, J. and Barzilay, R. (2008). Bayesian unsupervised topic segmentation.

[Euzenat et al., 2007] Euzenat, J., Shvaiko, P., et al. (2007). *Ontology matching*, volume 333. Springer.

[Feng et al., 2006] Feng, M., Heffernan, N., and Koedinger, K. (2006). Addressing the testing challenge with a web-based e-assessment system that tutors as it assesses. In *WWW '06 Proceedings of the 15th international conference on World Wide Web*, pages 307–316.

[Fikes and Nilsson, 1972] Fikes, R. E. and Nilsson, N. J. (1972). Strips: A new approach to the application of theorem proving to problem solving. *Artificial intelligence*, 2(3):189–208.

[Fitting, 1996] Fitting, M. (1996). *First-order logic and automated theorem proving*. Springer Science & Business Media.

[Fitzpatrick et al., 2003] Fitzpatrick, P., Metta, G., Natale, L., Rao, S., and Sandini, G. (2003). Learning about objects through action-initial steps towards artificial cognition. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 3, pages 3140–3145. IEEE.

[Fox et al., 2006] Fox, M., Gerevini, A., Long, D., and Serina, I. (2006). Plan stability: Replanning versus plan repair. In *ICAPS*, volume 6, pages 212–221.

[Fundel et al., 2007] Fundel, K., Küffner, R., and Zimmer, R. (2007). Relex—relation extraction using dependency parse trees. *Bioinformatics*, 23(3):365–371.

[Gentner, 1983] Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy*. *Cognitive science*, 7(2):155–170.

[Gentner et al., 1997] Gentner, D., Brem, S., Ferguson, R. W., Markman, A. B., Levidow, B. B., Wolff, P., and Forbus, K. D. (1997). Analogical reasoning and conceptual change: A case study of johannes kepler. *The journal of the learning sciences*, 6(1):3–40.

[Gerevini and Serina, 2000] Gerevini, A. and Serina, I. (2000). Fast plan adaptation through planning graphs: Local and systematic search techniques. In *AIPS*, pages 112–121.

[Gibson, 1977] Gibson, J. (1977). The concept of affordances. *Perceiving, acting, and knowing*, pages 67–82.

[Gil et al., 2011] Gil, Y., Gonzalez-Calero, P. A., Kim, J., Moody, J., and Ratnakar, V. (2011). A semantic framework for automatic generation of computational workflows using distributed data and component catalogues. *Journal of Experimental & Theoretical Artificial Intelligence*, 23(4):389–467.

[Gildea and Palmer, 2002] Gildea, D. and Palmer, M. (2002). The necessity of parsing for predicate argument recognition. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 239–246. Association for Computational Linguistics.

[Gloor et al., 2009] Gloor, P., Krauss, J., Nann, S., Fischbach, K., Schoder, D., et al. (2009). Web science 2.0: Identifying trends through semantic social network analysis. In *Computational Science and Engineering, 2009. CSE'09. International Conference on*, volume 4, pages 215–222. IEEE.

[Goldstone and Son, 2005] Goldstone, R. L. and Son, J. Y. (2005). *Similarity.* Cambridge University Press.

[Guadarrama et al., 2014] Guadarrama, S., Rodner, E., Saenko, K., Zhang, N., Farrell, R., Donahue, J., and Darrell, T. (2014). Open-vocabulary object retrieval. RSS.

[Hall et al., 2009] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18.

[Hammond, 1986] Hammond, K. J. (1986). Chef: A model of case-based planning. In *AAAI*, pages 267–271.

[Hanks and Weld, 1995] Hanks, S. and Weld, D. S. (1995). A domain-independent algorithm for plan adaptation. *Journal of Artificial Intelligence Research*, pages 319–360.

[Hausendorf and Quasthoff, 1992] Hausendorf, H. and Quasthoff, U. (1992). Patterns of adult-child interaction as a machanism of discourse acquisition. *Journal of Pragmatics*, 17:241–259.

[Havasi et al., 2007] Havasi, C., Speer, R., and Alonso, J. (2007). Conceptnet 3: a flexible, multilingual semantic network for common sense knowledge. In *Recent Advances in Natural Language Processing*, pages 27–29.

[Havasi et al., 2009] Havasi, C., Speer, R., and Alonso, J. (2009). *ConceptNet: A lexical resource for common sense knowledge*, volume 5. John Benjamins Publishing Company.

[Hermans et al., 2014] Hermans, A., Floros, G., and Leibe, B. (2014). Dense 3d semantic mapping of indoor scenes from rgb-d images. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 2631–2638. IEEE.

[Hilbert and Eis, 2014] Hilbert, D. D. and Eis, S. D. (2014). Early intervention for emergent literacy development in a collaborative community pre-kindergarten. *Early Childhood Education Journal*, 42(2):105–113.

[Hinrichs and Forbus, 2007] Hinrichs, T. R. and Forbus, K. D. (2007). Analogical learning in a turn-based strategy game. In *IJCAI*, pages 853–858.

[Hoffmann and Nebel, 2001] Hoffmann, J. and Nebel, B. (2001). The ff planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research*, pages 253–302.

[Hofmann, 1999] Hofmann, T. (1999). Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM.

[Hotho et al., 2006a] Hotho, A., Jäschke, R., Schmitz, C., and Stumme, G. (2006a). *Information retrieval in folksonomies: Search and ranking*. Springer.

[Hotho et al., 2006b] Hotho, A., Jäschke, R., Schmitz, C., and Stumme, G. (2006b). *Trend detection in folksonomies*. Springer.

[Hovy et al., 2006] Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., and Weischedel, R. (2006). Ontonotes: the 90% solution. In *Proceedings of the human language technology conference of the NAACL, Companion Volume: Short Papers*, pages 57–60. Association for Computational Linguistics.

[Hsueh et al., 2009] Hsueh, P.-Y., Melville, P., and Sindhwani, V. (2009). Data quality from crowdsourcing: a study of annotation selection criteria. In *Proceedings of the NAACL HLT 2009 workshop on active learning for natural language processing*, pages 27–35. Association for Computational Linguistics.

[Jung and Euzenat, 2007] Jung, J. J. and Euzenat, J. (2007). Towards semantic social networks. In *The semantic web: research and applications*, pages 267–280. Springer.

[Kent et al., 2014] Kent, D., Behrooz, M., and Chernova, S. (2014). Crowdsourcing the construction of a 3d object recognition database for robotic grasping. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 4526–4531. IEEE.

[Kimmig et al., 2012] Kimmig, A., Bach, S., Broecheler, M., Huang, B., and Getoor, L. (2012). A short introduction to probabilistic soft logic. In *Proceedings of the NIPS Workshop on Probabilistic Programming: Foundations and Applications*, pages 1–4.

[Kingsbury and Palmer, 2003] Kingsbury, P. and Palmer, M. (2003). Propbank: the next level of treebank. In *Proceedings of Treebanks and lexical Theories*, volume 3. Citeseer.

[Kipper et al., 2008] Kipper, K., Korhonen, A., Ryant, N., and Palmer, M. (2008). A large-scale classification of english verbs. *Language Resources and Evaluation*, 42(1):21–40.

[Kjellström et al., 2011] Kjellström, H., Romero, J., and Kragić, D. (2011). Visual object-action recognition: Inferring object affordances from human demonstration. *Computer Vision and Image Understanding*, 115(1):81–90.

[Koenig and Likhachev, 2002a] Koenig, S. and Likhachev, M. (2002a). D* lite. In *Eighteenth national conference on Artificial intelligence*, pages 476–483. American Association for Artificial Intelligence.

[Koenig and Likhachev, 2002b] Koenig, S. and Likhachev, M. (2002b). Improved fast replanning for robot navigation in unknown terrain. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 1, pages 968–975. IEEE.

[Kolodneer, 1991] Kolodneer, J. L. (1991). Improving human decision making through case-based decision aiding. *AI magazine*, 12(2):52.

[Kolodner, 2014] Kolodner, J. (2014). *Case-based reasoning*. Morgan Kaufmann.

[Konecnỳ et al., 2014] Konecnỳ, Š., Stock, S., Pecora, F., and Saffiotti, A. (2014). Planning domain+ execution semantics: A way towards robust execution? In *Qualitative Representations for Robots: Papers from the AAAI Spring Symposium*.

[Korat and Shamir, 2008] Korat, O. and Shamir, A. (2008). The educational electronic book as a tool for supporting children?s emergent literacy in low versus middle ses groups. *Computers & Education*, 50(1):110–124.

[Krause and Guestrin, 2006] Krause, A. and Guestrin, C. (2006). Data association for topic intensity tracking. Technical report, In International Conference on Machine Learning (ICML.

[Kuipers and Byun, 1991] Kuipers, B. and Byun, Y.-T. (1991). A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Robotics and autonomous systems*, 8(1):47–63.

[Kuniyoshi et al., 1994] Kuniyoshi, Y., Inaba, M., and Inoue, H. (1994). Learning by watching: Extracting reusable task knowledge from visual observation of human performance. *Robotics and Automation, IEEE Transactions on*, 10(6):799–822.

[Landauer et al., 1998] Landauer, T. K., Foltz, P. W., and Laham, D. (1998). An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284.

[Lee and Grauman, 2012] Lee, Y. J. and Grauman, K. (2012). Object-graphs for context-aware visual category discovery. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(2):346–358.

[Lenat, 1995] Lenat, D. B. (1995). Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38.

[Lin et al., 2014] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014*, pages 740–755. Springer.

[Linell, 1998] Linell, P. (1998). *Approaching Dialogue*, volume 1. John Benjamins Publishing Company, Philadelphia, PA.

[Lofi, 2013] Lofi, C. (2013). Just ask a human?–controlling quality in relational similarity and analogy processing using the crowd. In *BTW Workshops*, pages 197–210.

[Loper and Bird, 2002] Loper, E. and Bird, S. (2002). Nltk : The natural language toolkit. *Processing*, 1(July):1–4.

[Lopes et al., 2010] Lopes, M., Melo, F., Montesano, L., and Santos-Victor, J. (2010). Abstraction levels for robotic imitation: Overview and computational approaches. In *From Motor Learning to Interaction Learning in Robots*, pages 313–355. Springer.

[Malik et al., 2007] Malik, Z., Rezgui, A., and Sinha, A. K. (2007). Ontologic integration of geoscience data on the semantic web. In *Proceedings of the Geoinformatics Conference, San Diego, CA*.

[Matuszek et al., 2006] Matuszek, C., Cabral, J., Witbrock, M. J., and DeOliveira, J. (2006). An introduction to the syntax and content of cyc. In *AAAI Spring Symposium: Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering*, pages 44–49. Citeseer.

[McFee et al., 2012] McFee, B., Bertin-Mahieux, T., Ellis, D. P., and Lanckriet, G. R. (2012). The million song dataset challenge. In *Proceedings of the 21st international conference companion on World Wide Web*, pages 909–916. ACM.

[Miller, 1995] Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

[Miller et al., 1990] Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. J. (1990). Introduction to wordnet: An on-line lexical database*. *International journal of lexicography*, 3(4):235–244.

[Modi et al., 2005] Modi, J., Veloso, M., Smith, F. S., and Oh, J. (2005). Cmradar: A personal assistant agent for calendar management. In *Lecture Notes in Computer Science*, volume 3508, page 393.

[Mohseni-Kabir et al., 2015] Mohseni-Kabir, A., Rich, C., Chernova, S., Sidner, C. L., and Miller, D. (2015). Interactive hierarchical task learning from a single demonstration. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*, pages 205–212. ACM.

[Moldovan et al., 2012] Moldovan, B., Moreno, P., van Otterlo, M., Santos-Victor, J., and De Raedt, L. (2012). Learning relational affordance models for robots in multi-object manipulation tasks. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 4373–4378. IEEE.

[Montesano et al., 2008] Montesano, L., Lopes, M., Bernardino, A., and Santos-Victor, J. (2008). Learning object affordances: From sensory–motor coordination to imitation. *Robotics, IEEE Transactions on*, 24(1):15–26.

[Mulling et al., 2013] Mulling, K., Kober, J., Kroemer, O., and Peters, J. (2013). Learning to select and generalize striking movements in robot table tennis. *The International Journal of Robotics Research*, 32(3):263–279.

[Navigli, 2009] Navigli, R. (2009). Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)*, 41(2):10.

[Nebel and Koehler, 1995] Nebel, B. and Koehler, J. (1995). Plan reuse versus plan generation: A theoretical and empirical analysis. *Artificial Intelligence*, 76(1):427–454.

[Nicosevici and Garcia, 2012] Nicosevici, T. and Garcia, R. (2012). Automatic visual bag-of-words for online robot navigation and mapping. *Robotics, IEEE Transactions on*, 28(4):886–898.

[Niekum et al., 2013] Niekum, S., Chitta, S., Marthi, B., Osentoski, S., and Barto, A. G. (2013). Incremental semantically grounded learning from demonstration. *Robotics: Science and Systems 2013*.

[Niekum et al., 2014] Niekum, S., Osentoski, S., Atkeson, C. G., and Barto, A. G. (2014). Learning articulation changepoint models from demonstration. In *RSS Workshop on Learning Plans with Context from Human Signals*.

[Otero-Cerdeira et al., 2015] Otero-Cerdeira, L., Rodríguez-Martínez, F. J., and Gómez-Rodríguez, A. (2015). Ontology matching: A literature review. *Expert Systems with Applications*, 42(2):949–971.

[Parekh et al., 2004] Parekh, V., Gwo, J.-P., and Finin, T. W. (2004). Ontology based semantic metadata for geoscience data. In *IKE*, pages 485–490.

[Parish-Morris et al., 2013] Parish-Morris, J., Mahajan, N., Hirsh-Pasek, K., Golinkoff, R. M., and Collins, M. F. (2013). Once upon a time: parent–child dialogue and storybook reading in the electronic era. *Mind, Brain, and Education*, 7(3):200–211.

[Pedersen et al., 2004] Pedersen, T., Patwardhan, S., and Michelizzi, J. (2004). Wordnet:: Similarity: measuring the relatedness of concepts. In *Demonstration Papers at HLT-NAACL 2004*, pages 38–41. Association for Computational Linguistics.

[Pérez et al., 2006] Pérez, J., Arenas, M., and Gutierrez, C. (2006). Semantics and complexity of sparql. In *International semantic web conference*, volume 4273, pages 30–43. Springer.

[Perkins, 2010] Perkins, J. (2010). *Python text processing with NLTK 2.0 cookbook*. Packt Publishing Ltd.

[Pesquita et al., 2009] Pesquita, C., Faria, D., Falcao, A. O., Lord, P., and Couto, F. M. (2009). Semantic similarity in biomedical ontologies. *PLoS Comput Biol*, 5(7):e1000443.

[Pillinger and Wood, 2014] Pillinger, C. and Wood, C. (2014). Pilot study evaluating the impact of dialogic reading and shared reading at transition to primary school: early literacy skills and parental attitudes. *Literacy*.

[Podgorny and Garner, 1979] Podgorny, P. and Garner, W. (1979). Reaction time as a measure of inter-and intraobject visual similarity: Letters of the alphabet. *Perception & Psychophysics*, 26(1):37–52.

[Porteous et al., 2008] Porteous, I., Newman, D., Ihler, A., Asuncion, A., Smyth, P., and Welling, M. (2008). Fast collapsed gibbs sampling for latent dirichlet allocation. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 569–577. ACM.

[Pronobis and Jensfelt, 2012] Pronobis, A. and Jensfelt, P. (2012). Large-scale semantic mapping and reasoning with heterogeneous modalities. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 3515–3522. IEEE.

[Raubal and Moratz, 2008] Raubal, M. and Moratz, R. (2008). A functional model for affordance-based agents. In *Towards Affordance-Based Robot Control*, pages 91–105. Springer.

[Reed et al., 2002] Reed, S. L., Lenat, D. B., et al. (2002). Mapping ontologies into cyc. In *AAAI 2002 Conference Workshop on Ontologies For The Semantic Web*, pages 1–6.

[Ricci et al., 2011] Ricci, F., Rokach, L., and Shapira, B. (2011). *Introduction to recommender systems handbook*. Springer.

[Rich and Sidner, 1998a] Rich, C. and Sidner, C. L. (1998a). Collagen: A collaboration manager for software interface agents. *User Modeling and User-Adapted Interaction*, 8(3-4):315–350.

[Rich and Sidner, 1998b] Rich, C. and Sidner, C. L. (1998b). Collagen: A collaboration manager for software interface agents. *User Modeling and User-Adapted Interaction*, 8:315–350. 10.1023/A:1008204020038.

[Richter, 2013] Richter, S. (2013). Landmark-based heuristics and search control for automated planning. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 3126–3130. AAAI Press.

[Rogers and Christensen, 2013] Rogers, J. G. and Christensen, H. I. (2013). Robot planning with a semantic map. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 2239–2244. IEEE.

[Rusu et al., 2007] Rusu, D., Dali, L., Fortuna, B., Grobelnik, M., and Mladenic, D. (2007). Triplet extraction from sentences. In *Proceedings of the 10th International Multiconference" Information Society-IS*, pages 8–12.

[Ruvolo and Eaton, 2013] Ruvolo, P. and Eaton, E. (2013). Active task selection for lifelong machine learning. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence (AAAI-13)*.

[Sacerdoti, 1974] Sacerdoti, E. D. (1974). Planning in a hierarchy of abstraction spaces. *Artificial intelligence*, 5(2):115–135.

[Schafer et al., 1999] Schafer, J. B., Konstan, J., and Riedl, J. (1999). Recommender systems in e-commerce. In *Proceedings of the 1st ACM conference on Electronic commerce*, pages 158–166. ACM.

[Schiff et al., 2009] Schiff, R., Bauminger, N., and Toledo, I. (2009). Analogical problem solving in children with verbal and nonverbal learning disabilities. *Journal of Learning Disabilities*, 42(1):3–13.

[Schuler, 2005] Schuler, K. K. (2005). Verbnet: A broad-coverage, comprehensive verb lexicon.

[Segal-Drori et al., 2010] Segal-Drori, O., Korat, O., Shamir, A., and Klein, P. (2010). Reading electronic and printed books with and without adult instruction: effects on emergent reading. *Reading and Writing*, 23(8):913–930.

[Shepard, 1962] Shepard, R. N. (1962). The analysis of proximities: Multidimensional scaling with an unknown distance function. i. *Psychometrika*, 27(2):125–140.

[Shi and Mihalcea, 2005] Shi, L. and Mihalcea, R. (2005). Putting pieces together: Combining framenet, verbnet and wordnet for robust semantic parsing. In *Computational linguistics and intelligent text processing*, pages 100–111. Springer.

[Simpson, 2008] Simpson, E. (2008). Clustering tags in enterprise and web folksonomies. In *ICWSM*.

[Sinclair and Cardew-Hall, 2008] Sinclair, J. and Cardew-Hall, M. (2008). The folksonomy tag cloud: when is it useful? *Journal of Information Science*, 34(1):15–29.

[Singla and Domingos, 2006] Singla, P. and Domingos, P. (2006). Entity resolution with markov logic. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pages 572–582. IEEE.

[Sleator and Temperley, 1995] Sleator, D. D. and Temperley, D. (1995). Parsing english with a link grammar. *arXiv preprint cmp-lg/9508004*.

[Speer et al., 2010] Speer, R., Arnold, K., and Havasi, C. (2010). Divisi: Learning from semantic networks and sparse svd. In *Proc. 9th Python in Science Conf.(SCIPY 2010)*.

[Speer and Havasi, 2013] Speer, R. and Havasi, C. (2013). Conceptnet 5: A large semantic network for relational knowledge. In *The People?s Web Meets NLP*, pages 161–176. Springer.

[Speer et al., 2008] Speer, R., Havasi, C., and Lieberman, H. (2008). Analogyspace: Reducing the dimensionality of common sense knowledge. In *AAAI*, volume 8, pages 548–553.

[Stanton et al., 2012] Stanton, C., Bogdanovych, A., and Ratanasena, E. (2012). Teleoperation of a humanoid robot using full-body motion capture, example movements, and machine learning. In *Proceedings of Australasian Conference on Robotics and Automation*, pages 3–5. Victoria University of Wellington New Zealand.

[Steinbach et al., 2000] Steinbach, M., Karypis, G., and Kumar, V. (2000). A comparison of document clustering techniques. In *In KDD Workshop on Text Mining*.

[Thomaz and Cakmak, 2009] Thomaz, A. L. and Cakmak, M. (2009). Learning about objects with human teachers. In *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*, pages 15–22. ACM.

[Turney, 2006] Turney, P. D. (2006). Similarity of semantic relations. *Computational Linguistics*, 32(3):379–416.

[Turney, 2013] Turney, P. D. (2013). Distributional semantics beyond words: Supervised learning of analogy and paraphrase. *arXiv preprint arXiv:1310.5042*.

[Turney and Littman, 2005] Turney, P. D. and Littman, M. L. (2005). Corpus-based learning of analogies and semantic relations. *Machine Learning*, 60(1-3):251–278.

[Tversky, 1977] Tversky, A. (1977). Features of similarity. *Psychological Review*, 84:327–352.

[Unger et al., 2014] Unger, C., Freitas, A., and Cimiano, P. (2014). An introduction to question answering over linked data. In *Reasoning Web. Reasoning on the Web in the Big Data Era*, pages 100–140. Springer.

[Van Der Krogt and De Weerdt, 2005] Van Der Krogt, R. and De Weerdt, M. (2005). Plan repair as an extension of planning. In *ICAPS*, volume 5, pages 161–170.

[Varadarajan and Vincze, 2011] Varadarajan, K. M. and Vincze, M. (2011). Knowledge representation and inference for grasp affordances. In *Computer Vision Systems*, pages 173–182. Springer.

[Varadarajan and Vincze, 2012] Varadarajan, K. M. and Vincze, M. (2012). Afrob: The affordance network ontology for robots. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 1343–1350. IEEE.

[Varadarajan and Vincze, 2013] Varadarajan, K. M. and Vincze, M. (2013). Afnet: the affordance network. In *Computer Vision–ACCV 2012*, pages 512–523. Springer.

[Veloso et al., 2005] Veloso, M., Von Hundelshausen, F., and Rybski, P. E. (2005). Learning visual object definitions by observing human activities. In *Humanoid Robots, 2005 5th IEEE-RAS International Conference on*, pages 148–153. IEEE.

[Veloso and Carbonell, 1993] Veloso, M. M. and Carbonell, J. G. (1993). Derivational analogy in prodigy: Automating case acquisition, storage, and utilization. In *Case-Based Learning*, pages 55–84. Springer.

[Wallach, 2006] Wallach, H. M. (2006). Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd international conference on Machine learning*, pages 977–984. ACM.

[Wang and Blei, 2011] Wang, C. and Blei, D. M. (2011). Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 448–456. ACM.

[Weld et al., 2012] Weld, D., Adar, E., Chilton, L., Hoffmann, R., Horvitz, E., Koch, M., Landay, J., Lin, C., and Mausam, M. (2012). Personalized online education - a crowdsourcing challenge. In *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*.

[Wells, 1985] Wells, G. (1985). *Language development in the pre-school years*, volume 2. CUP Archive.

[Wenger, 2014] Wenger, E. (2014). *Artificial intelligence and tutoring systems: computational and cognitive approaches to the communication of knowledge*. Morgan Kaufmann.

[Whitehurst and Lonigan, 1998] Whitehurst, G. J. and Lonigan, C. J. (1998). Child development and emergent literacy. *Child development*, 69(3):848–872.

[Wiemer-Hastings et al., 2004] Wiemer-Hastings, P., Wiemer-Hastings, K., and Graesser, A. (2004). Latent semantic analysis. In *Proceedings of the 16th international joint conference on Artificial intelligence*, pages 1–14. Citeseer.

[Wilson and Scheutz, 2014] Wilson, J. R. and Scheutz, M. (2014). Analogical generalization of activities from single demonstration. In *Advances in Artificial Intelligence–IBERAMIA 2014*, pages 494–505. Springer.

[Worgotter et al., 2013] Worgotter, F., Aksoy, E. E., Kruger, N., Piater, J., Ude, A., and Tamosiunaite, M. (2013). A simple ontology of manipulation actions based on hand-object relations.

[Xu et al., 2008] Xu, S., Bao, S., Fei, B., Su, Z., and Yu, Y. (2008). Exploring folksonomy for personalized search. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 155–162. ACM.

[Yoshikawa et al., 2006] Yoshikawa, Y., Shinozawa, K., Ishiguro, H., Hagita, N., and Miyamoto, T. (2006). Responsive robot gaze to interaction partner. In *Robotics: Science and systems*.

[Zhao et al., 2011] Zhao, W. X., Jiang, J., Weng, J., He, J., Lim, E.-P., Yan, H., and Li, X. (2011). Comparing twitter and traditional media using topic models. In *Advances in Information Retrieval*, pages 338–349. Springer.