The 25th International Joint Conference on Artificial Intelligence

Proceedings of the IJCAI 2016 Workshop - Closing the Cognitive Loop: 3rd Workshop on Knowledge, Data, and Systems for Cognitive Computing

Edited by:

Kartik Talamadupula, Shirin Sohrabi, Murray S. Campbell

New York, USA, July 11, 2016

Organising Committee

Kartik Talamadupula, IBM, USA

Shirin Sohrabi, IBM, USA

Murray S. Campbell, IBM, USA

Program Committee

J. Benton, NASA Ames Research Center

Gordon Briggs, Naval Research Laboratory

Minh Do, NASA Ames Research Center

Scott Friedman, SIFT

Robert Goldman, SIFT

Yuheng Hu, Univ. of Illinois Chicago

Udayan Khurana, IBM Research

Andrey Kolobov, Microsoft Research

Janusz Marecki, Google DeepMind

Martin Michalowski, Adventium Labs

Srinivasan Parthasarathy, IBM Research

Anton V. Riabov, IBM Research

Stephanie Rosenthal, Carnegie Mellon University

Francesca Rossi, Univ. of Padova / IBM Research

Siddharth Srivastava, United Technologies Research Center

Reza Zafarani, Syracuse University

Yu Zhang, Arizona State University

Foreword

As AI techniques are increasingly employed in real world applications and scenarios, their contact with humans is ever-increasing. Traditionally, most AI systems have tended to exclude humans and the problems that accompany interaction with them. This has enabled the development of algorithms and even end-to-end systems that produce "optimal" artifacts that cut humans completely out of the loop, while still operating in a world where the assumption is that humans will be the end-consumers of the artifacts produced by such systems. Cognitive computing is a new paradigm that seeks to replace that diffidence and sometimes even mistrust of humans with a vision of successful cooperation and teaming between humans and AI systems and agents. The key idea is that human-machine teams can often achieve better performance than either alone. To enable this, AI techniques must not only accommodate humans in the decision-making loop, but to go to great lengths to make such participation as natural and simple as possible. Building such cognitive computing systems and agents will thus require contributions from many areas of AI as well as related fields. We call this process the "closing of the cognitive loop", and all contributions to the workshop are evaluated on their ability to demonstrate the successful closing of this loop, or technical extensions to existing work that can close it. The aim of this workshop is to bring together the work of researchers who are interested in advancing the state-of-the-art not merely in their specific sub-field of AI, but are also willing to engage in technically directed discussions on what is missing currently from their work that is needed to turn it into a deployed service that can gainfully interact with humans and the world at large.

The CogComp16 Organizers

Table of Contents

STRIDER: Toward an AI Collaborator for Intelligence Analysis

Scott Friedman, Mark Burstein, David McDonald
SIFT, Minneapolis, MN, USA
{friedman, burstein, dmcdonald}@sift.net

Roger Rosewall
George Washington University
rosewall@gwu.edu

J. Benton
NASA ARC & AAMU-RISE
j.benton@nasa.gov

Abstract

Intelligence analysts gather information from diverse sources and integrate it into a product that adheres to standards of quality and provenance, often under time pressures and information overload. The STRIDER system, which we describe in this paper, enables collaborative exploration, hypothesis formation, and information fusion from open-source text. STRIDER presents relevant information to the human analyst in an intuitive fashion and allows various forms of feedback through a diagrammatic interface to enhance its understanding of shared problem-solving objectives. The human analyst supports STRIDER's collaborative workflow by postulating new entities, events, associations, and hypotheses, to improve STRIDER's information extraction and relevance judgments. STRIDER models the analyst's objectives and focus in order to avoid presenting information to the analyst at the wrong time or in the wrong context, which could be distracting, or worse, misleading. The technology in STRIDER is motivated by known human cognitive biases and limitations, and compensates for these to improve the breadth, efficiency, and objectivity of intelligence analysis. We focus on two pillars of collaborative cognitive computing: (1) interfacing for bi-directional human-computer interaction that encodes the analyst's objectives and presents relevant information, and (2) support for mutual decision-making by the analyst and the system. We present preliminary empirical evidence to demonstrate STRIDER's effectiveness in extracting and identifying relevant information.

Introduction

This paper describes progress toward closing the cognitive loop in an intelligence analysis setting, where analysts face an information overload and require up-to-date information relevant to their intelligence objectives. We describe our progress toward this goal with our integrated system STRIDER (*Semantic Targeting of Relevant Individuals, Dispositions, Events, & Relationships*). STRIDER diagrammatically elicits the intelligence objectives of the analyst, automatically gathers relevant information from multiple sources of unstructured text, encodes necessary metadata, and presents information to analysts when relevant. This will facilitate compliance with quality and provenance policies and make analysts more efficient and effective. We describe STRIDER with respect to two primary pillars of cognitive computing: interfacing and decision support.

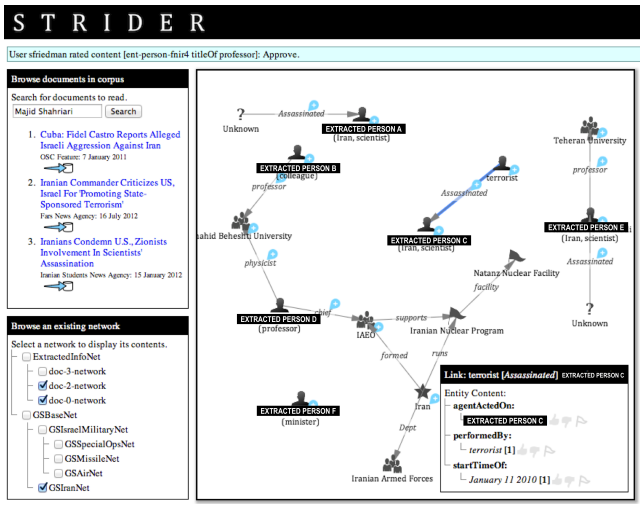


Figure 1: STRIDER's web-based interface.

Interfacing. STRIDER uses *link diagrams* to display individuals, events, organizations, and other entities as nodes in a network, connected by directed semantic links, as shown in the Figure 1 screenshot. Link diagrams are intuitive intelligence analysis interfaces: they do not require technical expertise with ontologies or knowledge representation, and other software systems use these representations for intelligence analysis (e.g., Carley et al., 2012, Stasko, Gorg, and Liu, 2008) and education. STRIDER exploits link diagrams as an interface for two purposes:

- **Soliciting objectives and queries from the analyst.** STRIDER's link diagrams have unambiguous semantics. This allows the analyst to extend and manipulate link diagrams to perform high-level fusion, specify objectives, and issue unambiguous directives to STRIDER.
- **Presenting relevant information to the analyst when appropriate.** While the analyst reads an article, STRIDER presents semantic information from that article— as well as semantically-related information from other sources— in a link diagram.

STRIDER uses the same diagrammatic interface to present information and to elicit objectives and feedback. This provides a shared workspace for the analyst and the machine

to collaborate and share progress toward the explicit— and potentially changing— objectives.

Decision support. Just as human collaborators mutually influence and support each others' decisions, the analyst's actions support STRIDER's decisions, and STRIDER's actions are designed to support the analyst's decisions. The influence is bi-directional:

- **Analyst influences STRIDER's decisions** by extending or annotating link diagrams. For instance, STRIDER labels *information gaps* in the analyst's diagram, which become goals for STRIDER's information extraction. Also, as the link diagram grows or otherwise changes, the set of relevant entities and relationships change, which affects the space of information that STRIDER will decide is relevant enough to present to the analyst.
- **STRIDER influences analyst decisions** by presenting relevant information. STRIDER may thereby influence the analyst's focus, e.g., toward relevant organizations or events. Further, since STRIDER maintains provenance according to IC directives (Office of the Director of National Intelligence, 2007a,b), the analyst may expand relevant entities in the link diagram and peruse other supporting documents from the corpus.

This mutual influence is desirable from a collaborative workflow perspective: it allows the analyst to drive the objectives and utilize their deep intuition and common sense, while exploiting the machine's broad parallel processing and book-keeping. However, if the machine collaborator extracts erroneous information or displays irrelevant information to the analyst, it will distract or mislead the analyst and thereby derail the workflow. We conducted a pilot study, described below, to estimate the precision and completeness of STRIDER's information extraction and relevance judgments, compared to a senior IC analyst, and we present encouraging results. This pilot study precedes more detailed workflow analyses of analysts using STRIDER, which is a central goal of future work, as we describe below.

We continue by outlining the tasks and cognitive biases relevant to intelligence analysis, which motivate STRIDER's complementary cognitive computation. We then describe the STRIDER architecture and the information flows that support the pillars of cognitive computation described above. We present results from a pilot study to demonstrate STRIDER's effectiveness on these tasks, and close with a discussion of relevant and future work.

Strategic Intelligence Analysis

STRIDER's design is guided by the following guidelines of strategic intelligence analysis, based on Intelligence Community Directives (ICDs) (e.g., Office of the Director of National Intelligence, 2007a,b, 2009). We describe each guideline and STRIDER's contribution.

Objectivity. Analysis should be free of emotional content, regard alternative/contrary reports, and acknowledge developments. STRIDER supports this ideal with objective information extraction: deep semantic parsing extracts the semantics reported by the source; its only interpretive bias is the ontology with which it represents information.

Based on all available sources. Analysis should be informed by all relevant information available, and information collectors should address critical gaps. STRIDER explicitly identifies *information gaps* (i.e., missing data about an individual or event) and labels the source coverage (i.e., sources of information that support each datum).

Describe quality & reliability of sources. Open-source references should include metadata such as reference type, author, publication, title/subject, date, and more. STRIDER tracks all of these data, down to the specific paragraphs and character offsets supporting the extracted information.

Distinguish between intelligence & assumptions. Assumptions are hypotheses or foundations on which conclusions are reached, so critical assumptions must be explicitly identified, and so should indicators that may validate or invalidate assumptions. STRIDER helps analysts identify assumptions in their diagrams by automatically identifying unsupported information and information gaps.

Incorporate alternative analyses & hypotheses. Analytic products should identify and qualify alternative hypotheses in light of available information and information gaps. STRIDER's hypothesis-based organization (described below) helps analysts segment and compare competing hypotheses, which helps compensate for known cognitive limitations (Heuer, 1999, Johnston, 2005).

Timeliness. Analytic products must be disseminated to customers with enough time to be actionable. The integrated STRIDER system— from information-gathering to provenance to reporting— aims to improve efficiency.

Approach

Here we outline the general STRIDER approach, starting with STRIDER's interfacing advances, then describing supporting technology, and finally describing STRIDER's decision-making and how it is affected by the analyst's objectives and directives.

Diagrammatic Interfacing

STRIDER uses link diagram interfaces to display information and communicate objectives and queries. Figure 1 illustrates the link diagram display of STRIDER, which supports touch-based, pen-based, or mouse-based HTML5-enabled devices. Figure 2 illustrates how analysts manipulate STRIDER's link diagrams to express their intent.

STRIDER's interface provides an informative, intuitive, and domain-general *shared workspace* for human-machine collaboration, without requiring proficiency with ontologies or knowledge representation. To be sure, link diagrams are not as expressive as natural language, but as shown in Figure 2, annotating a link diagram offers significant flexibility for expressing objectives and queries. These annotations in Figure 2 include the following:

- **Connecting** annotations indicates that STRIDER should find direct or indirect relationships between existing entities or events in the link diagram.

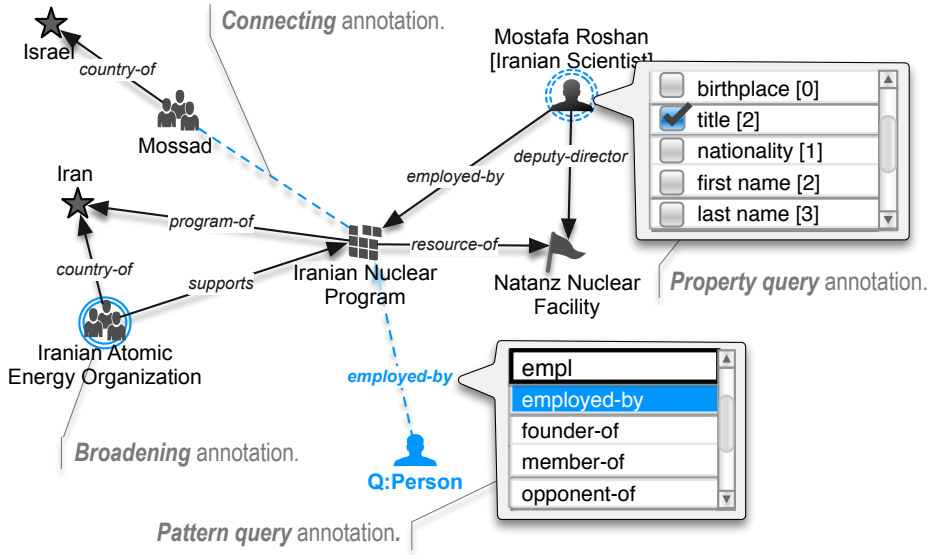


Figure 2: STRIDER's link diagram query interface.

- **Broadening** annotations indicate a broad interest in a specific entity or event in the diagram, and STRIDER should extract additional properties and events related to this.
- **Pattern query** annotations specify a pattern of interest—including a semantic relationship and one or more unknown entities— for STRIDER to match in the corpus.
- **Property query** annotations specify one or more properties of interest of an existing entity or event in the diagram, and STRIDER should extract additional evidence and/or values to fill that property.

STRIDER's use of link diagram manipulation to specify intent is inspired by Visual Query Systems (VQSs) for databases, web services, and other information repositories (e.g., Calvanese et al., 2010, Catarci et al., 1997). VQSs depict the domain of interest and express related requests, and aim to simplify complex query languages such as SQL and SPARQL. *Direct manipulation* (i.e., direct annotating or altering) of VQSs replaces the less-intuitive command language syntax, and benefits the user by reducing barrier of entry. This increases the ease of learning, providing high efficiency with experts, and reducing error rate (Ziegler and Fahnrich, 1988). We believe that STRIDER's direct diagram manipulations for querying and issuing directives, as shown in Figure 2, are novel interactions for specifying intent in a mixed-initiative information-gathering setting.

Deep Natural Language Understanding

Deep parsing allows STRIDER to extract precise semantics and determine entity types from local lexical context. STRIDER uses the SPARSER (McDonald, 1996) rule-based, type-driven semantic parser to read unstructured news articles. SPARSER'S rules succeed only if the types of the constituents to be composed satisfy the type constraints (i.e., value restrictions) specified by the rule. SPARSER compiles a semantic grammar from a semantic model of the information to be analyzed, including a specification of all the ways

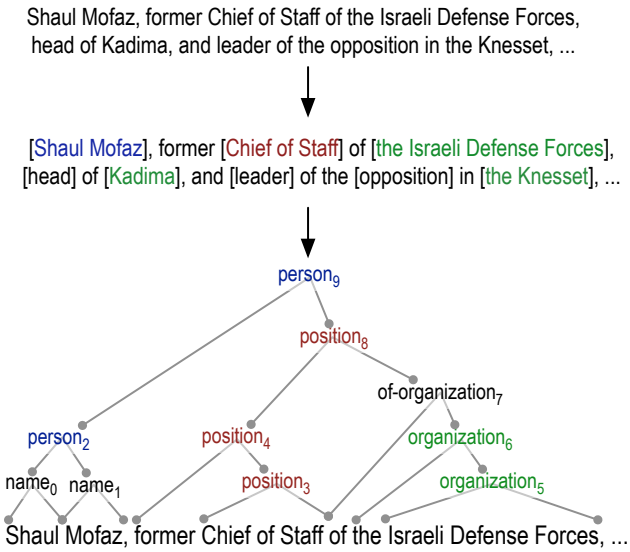


Figure 3: SPARSER efficiently analyzes text over multiple passes. Subscripts in the final semantic structure indicates the order in which SPARSER instantiated the instances.

each of the concepts can be realized in the language of the domain (e.g., open-source news articles). This ensures that everything SPARSER is able to parse it can model, and that every rule in the compiled grammar has an interpretation.

Figure 3 illustrates SPARSER'S scanning algorithm at a high level. In the first step, SPARSER segments or brackets the text into phrases, referred to as *segments*. Some of the segments include both known and unknown words, and some words are not included in any segment. SPARSER then detects instances of people, organizations, titles, times, locations, and more. It links these and other instances to exact locations in the document corpus to preserve the data source,

in compliance with U.S. IC directives (e.g., Office of the Director of National Intelligence, 2007b).

Next, SPARSER uses rules and discourse heuristics to identify relations that connect the phrase segments. This is shown in the final SPARSER scan in Figure 3. Unless an established grammar and conceptualization applies, SPARSER relies on a set of textual relations drawn from its standard syntactic vocabulary. Along with the recognized entities, these segment-spanning relations represent the meaning of the text for STRIDER, and they are used to populate larger content models, as we describe below.

Organizing Competing Hypotheses

Tracking alternative hypotheses— and gathering evidence for each— is a recurring theme in the intelligence analysis literature, and is notably difficult for analysts due to cognitive biases and attentional limitations (Heuer, 1999, Johnston, 2005, Pirolli and Card, 2005).

Traditional link diagrams are *flat*, in that they conflate potentially disjoint or competing hypotheses into the same network structure. Consequently, they may contain entities and links that are unrelated, competing, or disjoint. We believe that competing hypotheses should be displayed and considered separately to preserve relevance and help the analyst weigh alternative outcomes.

We have integrated SPIRE inference and knowledge base (KB) technology— from our ongoing work on parsing biomedical texts (Friedman et al., 2016)— to house the STRIDER knowledge base. This allows us to store STRIDER sub-networks hierarchically in different *logical contexts*. Logical contexts support inheritance, as illustrated in Figure 4: the *Core* context contains the portion of the network shared by each competing scenario; and each competing scenario (e.g., *Covert* and *Missile*) inherits the *Core* network, but none of its sibling networks.

Importantly, the branching of hypotheses can continue beyond the single four-way split shown in Figure 4. For instance, SPIRE could support multiple sub-scenarios that inherit from the *Covert* scenario shown here, where each sub-*Covert* scenario contains mutually exclusive entities and relations. SPIRE also supports multiple inheritance of contexts, so a STRIDER context could inherit from both *SOF* and *Air* hypotheses to describe a joint strike.

Aggregating Semantic Content

STRIDER uses *content models* to organize, inherit, and prioritize knowledge about different types of entities, events, and relationships. Content models relate to the object-level ontology like a database view relates to a database. Each content model is associated with a category in STRIDER's ontology and specifies a partially-ordered list of properties that may be relevant for analysis. For example, the content model *Person CM* in Figure 5, associated with category *Person*, inherits all properties from the content model *Base CM*. STRIDER uses content models to aggregate presentable or queriable information, to support the following capabilities:

1. Detect gaps in information (i.e., unpopulated properties) or evidence (i.e., properties without support from the corpus). This helps the analyst manage uncertainty judgments (Heuer, 1999) and reduces the cognitive cost of monitoring for information gaps (Pirolli and Card, 2005).
2. Determine whether information should be rendered as a node (e.g., like the Person CM) or a link (e.g., like the Assassination CM) in the diagram.
3. Display relevant *drill-down* data for nodes and links.
4. Specify *equivalence classes* over categories, to help STRIDER detect equivalent entities (i.e., references to the same real-world entity or event) and data conflicts (i.e., multiple, inconsistent property values) within and across information sources.

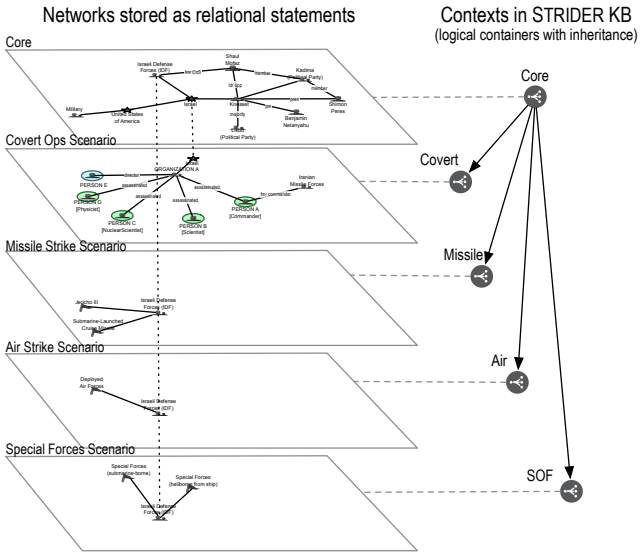


Figure 4: STRIDER records alternative hypotheses in separate hierarchical contexts.

Similarity-Based Reasoning

Given an entity, individual, or event of interest, STRIDER uses similarity-based retrieval to identify semantically similar *analogs* from its knowledge base. These retrieved analogs may help the analyst establish precedence and reason from previous examples to identify possible outcomes. STRIDER's similarity-based retrieval feature is motivated by well-known cognitive biases in memory retrieval and likelihood estimation; for instance, people use the sub-optimal *availability strategy* to estimate the probability of an event based on memory retrieval and imagination: assuming that if an event occurs frequently (and is therefore more probable), we can recall more instances of it (Heuer, 1999). Unfortunately, the ability to recall instances of an event is influenced by recency, context, vividness, and many other factors that are unrelated to the objective probability of an event.

STRIDER uses *structure-mapping*, a constrained graph-matching algorithm (e.g. Falkenhainer, Forbus, and Gentner, 1989, Friedman et al., 2016) to compute isomorphisms between semantic graphs and compute a numerical similarity rating. Given an entity (e.g., an Iranian nuclear scientist) in the network, STRIDER computes a subgraph of the entity

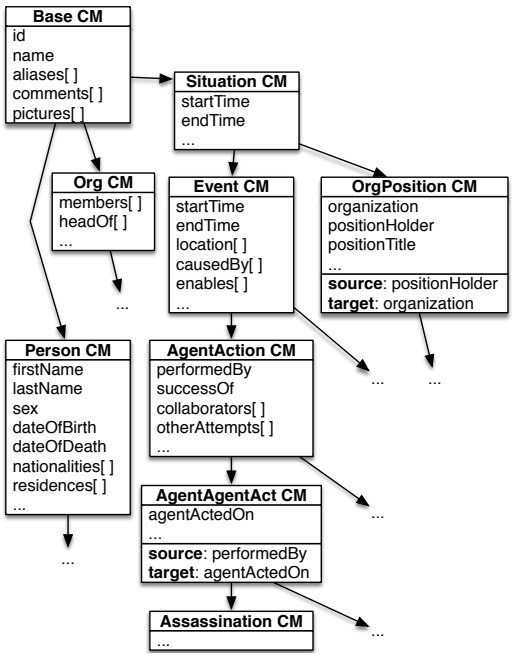


Figure 5: A portion of STRIDER's content model hierarchy.

from the content model and related events (e.g., the event describing the scientist's assassination), and matches it over the rest of the extracted semantic content in the KB to find similar analogues.

In this fashion, STRIDER not only builds a semantic network describing an event or topic of interest; it also relates different event descriptions using structural similarity, finding similar, or related, events and people, which helps to broaden the analysis.

Influences on STRIDER's Decision-Making

STRIDER's decision-making is influenced by the analyst's objectives (i.e., the analyst's annotated link diagram) and the analyst's focus (i.e., the article they are reading).

When STRIDER extracts information from text, it uses its content models to fuse the information into new or existing nodes or edges in its link diagrams. If the analyst has manually created nodes or links for entities and events (e.g., an individual or an organization), and then these are subsequently mentioned in an article, STRIDER will automatically extend the existing nodes and edges with the new information or evidence from the article; otherwise, STRIDER will generate new nodes and edges in the diagram.

To establish link diagram portions that are relevant to an article that the analyst is presently reading, STRIDER (1) parses the article, (2) grounds the entities and events referenced by the article within its link diagram(s), and then (3) displays entities and events within a specified link distance from the mentioned events or entities. STRIDER's distance-based relevance metric is effective for our current means, but we believe other methods, such as token-passing spreading activation, will yield better results as STRIDER accrues dense diagrams, as we describe in future work.

As mentioned above, if STRIDER incorrectly extracts information and then decides to present it— or if STRIDER decides to present otherwise irrelevant data— then STRIDER could distract or mislead the analyst. We next describe a pilot study to evaluate STRIDER's information extraction and fusion, compared to a senior intelligence analyst.

Information Extraction Pilot Experiment

We conducted a pilot experiment to assess STRIDER's ability to produce report-ready diagrams to aide a human analyst collaborator. In this experiment, a U.S. IC analyst consultant used a third-party diagram tool (i.e., *not* STRIDER) to build a link diagram from news articles, and we compare STRIDER's ability to extract, aggregate, and present the same information in a diagram, completely autonomously.

We tested STRIDER's information extraction on three articles from the Open Source Center (OSC, http://www.opensource.gov) in order to reproduce real link diagrams created by the analyst on the same material. The IC analyst used the third-party tool to build a *gold-standard* link diagram from many articles, citing individual sources. We used the a subset of the gold-standard diagram, including one of STRIDER's three articles, that described four assassinated individuals, an attempted assassination, and more. From the single-article portion of the analyst's gold-standard diagram, we counted 34 *data fragments*, including names, relationships, categories, events, dates, titles, nationalities, affiliations, organizations, and more.

STRIDER extracted all individuals correctly, most organization affiliations, two of four assassination events with dates intact, and more. However, due to gaps in parsing coverage, it missed two assassination events, it missed one unsuccessful assassination attempt, and it did not label a person's nationality:

- 33 total data fragments were extracted (e.g., names, relationships, events, dates, and titles)
- 30 fragments were in the analyst's diagram.
- 4 fragments (three events and one relation) were missing.
- 3 fragments (about an individual) were correct but deemed irrelevant by the analyst.
- 0 fragments were incorrect.

In total, the precision was 1.0, the recall was 0.88, for an F1 score of 0.94. STRIDER analyzed all three articles and recorded the sources of information in under one minute, and the human expert analyzed and recorded this information in one hour. Four elements were not extracted due to complex grammatical constructions that makes intersentence references to "bombings" and "attacks."

We subsequently gave STRIDER the remaining two OSC articles about the same events. From these, STRIDER extracted some consistent and some additional information and used its content models to fuse the new information into the diagram. The additional documents contained two previously-missing assassination events, and three university affiliations (two were diagrammed by the analyst, one is novel). Figure 6 illustrates the diagram STRIDER deemed relevant to the the initial article, with names hidden. It contains relevant information from the two subsequent articles,

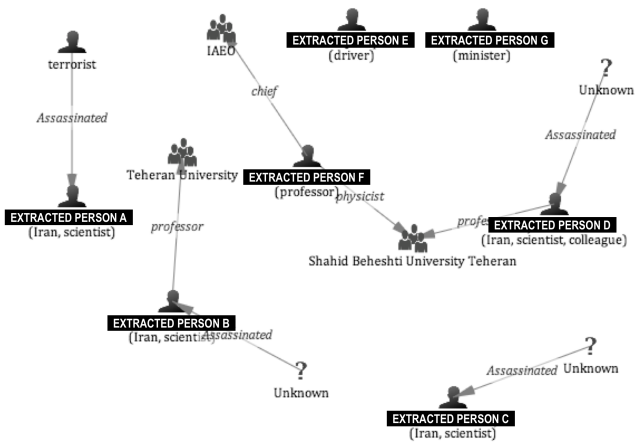


Figure 6: STRIDER screenshot (names hidden) showing information extracted and merged from three OSC articles.

hiding data that was determined irrelevant due to diagram distance from nodes mentioned in the article.

Overall, we demonstrated STRIDER extracting information, identifying gaps, presenting information and incorporating feedback using one consistent interface. These preliminary results suggest that STRIDER can:

- Extract information about individuals (e.g., their names, careers, nationalities, and titles), assassinations, organization affiliations, and role-based slot-fillers (e.g., an article mentions a"terrorist" assassinating somebody instead of a specific organization or individual).
- Extract partial information when complete information is not available.
- Utilize its content model and equivalence classes to merge and organize data across multiple sources.
- Display extracted information for non-technical users.

Information fusion is a time-intensive process of reading documents, extracting events and tying people and entities to related events. We demonstrated how a system like STRIDER can both help with the reading and information extraction process and also support the analyst in interactively tying relevant people and events together from multiple documents, improving the speed the process dramatically.

Related Work

STRIDER's interfacing exploits link diagram interfaces for information presentation and visual queries. Previous work in *network analysis* and *link analysis* utilize link diagrams as information displays. Organizational Risk Analysis (ORA) (Carley et al., 2012), a network analysis tool, automatically analyzes dynamic networks, social networks, geo-spatial data, and workflows. Jigsaw (Stasko, Gorg, and Liu, 2008) represents documents and their entities visually to help analysts examine them more efficiently, with an emphasis on illustrating connections between entities across documents. This reduces the cognitive load of data analysis. CRAFT (Gruen et al., 2008) supports wiki-like analyst collaboration with link diagram and form-based interfaces, to help analysts extend and share hypotheses, inquiries, and ontologies. These tools analyze link diagrams and support collaboration, but to our knowledge, they do not encode and autonomously react to analysts' changing objectives.

Semantic targeting in online advertising incorporates a semantic representation alongside an optional syntactic (i.e., keyword or bag-of-words) representation. Ad and page semantics can be evaluated for proximity (i.e., relevance) by calculating their taxonomic distance, allowing for semantic matches when no exact lexical matches are found. For instance, if a webpage describes a curling event, but no curling ads exist, ads belonging to the semantic class "skiing" (a sibling of class "curling" under the parent "winter sports") could be retrieved and delivered. STRIDER's relevance criteria is inspired, in part, by web-based semantic targeting.

Conclusion & Future Work

We described the cognitive computing technology of the STRIDER system for collaborative intelligence analysis, with a focus on (1) diagrammatic interfaces for eliciting the analyst's objectives and presenting relevant information, and (2) the influence of the analyst's objectives on STRIDER's decision-making. We presented preliminary results of STRIDER's information extraction and fusion, and we compared STRIDER's product to a senior intelligence analyst's diagram, with encouraging results (F1 = 0.94).

Our pilot experiment demonstrates that STRIDER can gather and present information, but it does not evaluate all of STRIDER's features as a collaborative closed-loop system. Evaluating the analyst's workflow with and without STRIDER will help qualify (1) how STRIDER addresses its user's cognitive biases and limitations, (2) the analyst time and effort required for inputting and maintaining information in STRIDER, and (3) STRIDER's overall impact on the breadth, objectivity, and efficiency of intelligence analysis.

In addition to evaluation, we have significant development remaining to realize our vision for STRIDER. For instance, STRIDER computes relevance using a diagram distance metric from the analyst's manually-created or annotated elements of the link diagram. We believe that a semantics-directed spreading activation algorithm will yield more complete and precise results. Also, we plan to have STRIDER learn from analysts' feedback on extracted data: if the analyst resolves a data conflict by choosing data from one source over another, STRIDER can generate or revise a topical model of source credibility.

Finally, STRIDER must support noisy data, e.g., as news reports are revised, as assumptions are violated, and as situations develop. This is crucial for cognitive aides in the intelligence analysis domain, to support human decision-makers in a partially-observable, uncertain, and changing world. We believe that metadata-based approaches for conflict resolution (e.g., Bleiholder and Naumann, 2006), in conjunction with human feedback and collaborative filtering, will help STRIDER semi-automatically prioritize conflicting data, but this is an empirical question and an area of future work.

References

Bleiholder, J., and Naumann, F. 2006. Conflict handling strategies in an integrated information system. In *Proceedings of the IJCAI Workshop on Information on the Web*.

Calvanese, D.; Keet, C. M.; Nutt, W.; Rodriguez-Muro, M.; and Stefanoni, G. 2010. Web-based graphical querying of databases through an ontology: the wonder system. In *SAC '10*.

Carley, K. M.; Pfeffer, J.; Reminga, J.; Storrick, J.; and Columbus, D. 2012. Ora user's guide 2012. Technical Report CMU-ISR-12-105, Carnegie Mellon University, School of Computer Science, Institute for Software Research.

Catarci, T.; Costabile, M. F.; Levialdi, S.; and Batini, C. 1997. Visual query systems for databases: A survey. *Journal of Visual Languages & Computing* 8(2):215–260.

Falkenhainer, B.; Forbus, K. D.; and Gentner, D. 1989. The structure-mapping engine: Algorithm and examples. *Artificial Intelligence* 41(1):1 – 63.

Friedman, S.; Burstein, M.; McDonald, D.; Paullada, A.; Plotnick, A.; Bobrow, R.; Cochran, B.; Pustejovsky, J.; and Anick, P. 2016. Learning by reading: Extending & localizing against a model. In *Proceedings of the Fourth Annual Conference on Advances in Cognitive Systems*.

Gruen, D. M.; Rasmussen, J. C.; Liu, J.; Hupfer, S.; and Ross, S. I. 2008. Collaborative reasoning and collaborative ontology development in craft. In *AAAI Spring Symposium: Symbiotic Relationships between Semantic Web and Knowledge Engineering*, 51–58.

Heuer, R. 1999. *Psychology of Intelligence Analysis*. Center for the Study of Intelligence, Central Intelligence Agency.

Johnston, R. 2005. *Analytic Culture in the United States Intelligence Community: An Ethnographic Study*. Central Intelligence Agency: Washington, DC.

McDonald, D. D. 1996. The interplay of syntactic and semantic node labels in partial parsing. In Bunt, H., and Tomita, M., eds., *Recent Advances in Parsing Technology*. Kluwer Academic Publishers. 295323.

Office of the Director of National Intelligence. 2007a. Intelligence community directive 203: Analytic standards.

Office of the Director of National Intelligence. 2007b. Intelligence community directive 206: Sourcing requirements for disseminated analytic products.

Office of the Director of National Intelligence. 2009. Intelligence community directive 501: Discovery and dissemination or retrieval of information within the intelligence community.

Pirolli, P., and Card, S. 2005. Sensemaking processes of intelligence analysts and possible leverage points as identified through cognitive task analysis. In *Proceedings of the 2005 International Conference on Intelligence Analysis*.

Stasko, J.; Gorg, C.; and Liu, Z. 2008. Jigsaw: supporting investigative analysis through interactive visualization. *Information Visualization* 7:118–132.

Ziegler, J. E., and Fahnrich, K. P. 1988. Direct manipulation. In Helander, M., ed., *Handbook of Human-Computer Interaction*. North-Holland, Amsterdam. 123–133.

Automating the recoding, analysis, and interpretation pipeline using naturalistic visual scenes

April M. Schweinhart, Baxter S. Eaves Jr., and **Patrick Shafto**
Rutgers University - Newark
{april.schweinhart, baxter.eaves, patrick.shafto} @rutgers.edu

Abstract

Machine learning often focuses on how best to infer structure from data. Also important is the ability to convey that structure to human users. We investigate a system for automating quantification, analysis, and presentation of data to human users. We focus on the domain of natural scenes, an area in which human performance has been well explored, and can thus be used to inform choices of computational tools. Informed by perceptual science, we characterize a corpus of images in terms of the statistics of their orientation distributions. In two experiments, we compare mixture and topic models for analysis, and teaching-optimized versus average images for conveying model structure to people. Using a categorization task, in Experiment 1, we find that, when subclusters are overlapping and categorization difficult, examples selected to teach the category structure lead to improved categorization performance relative to examples closest to the mean. Experiment 2 further shows that mixture models outperformed topic models and teaching examples outperformed maximum likelihood. By leveraging cognitively natural machine learning methods to facilitate automatic analysis and summary of naturalistic data, this work has implications for conveying both typicality and variability of experiences in complex data.

1 Introduction

For decades the focus of machine learning has been to take a prepared data set and, given some parameters, infer structure from it. While it is important to find such patterns in data, it is also important to ensure that structure can be easily accessed by humans attempting to make sense of the data. Indeed, problems anywhere in the pipeline—from quantifying experiences, to inferring structure, to interpreting and acting on results—may to lead to incorrect outcomes. Thus, a critical problem for machine learning, data science, and artificial intelligence more generally is how to make choices about each step so that the people at the end of the data analysis pipeline understand the output and make correct decisions.

In this paper, we investigate automating the quantification, analysis, and interpretation pipeline. Solving this problem is a long term goal. Whereas typical machine learning and data science approaches rely on highly educated experts to implement and interpret analyses, we present a specific example of a general approach based on leveraging humans' uniquely powerful ability to learn from small amounts of data generated by teachers. A naive computational learner infers some structure in the data and a computational teacher selects a small subset of the original data that best convey that structure to humans learners. We compare teaching decisions to well-known alternative methods not motivated by human reasoning in the domain. Success on this project would greatly increase the accessibility of data-driven decision making by reducing the need for specific training.

We focus on a domain (natural scene perception) and task (categorization) that have been well studied in the human learning literature. This allows us to select methods of quantifying and analyzing data that are strongly informed by existing science. Specifically, we leverage known human competencies in perception, cognition and social learning. In two experiments, we investigate different machine learning methods—mixture models and topic models—and methods of summarizing their results—selecting examples through computational models of teaching or that capture the mean or maximum likelihood estimate. Teaching has computational support in the literatures on perception, cognition, and social learning, while choosing data close to the mean or that maximize likelihood do not.

The paper unfolds in three sections. First, we discuss foundational work in the areas of natural scene perception, categorization, and computational models of teaching. Second, we describe the pipeline for quantifying images, extracting categories from these data, and selecting images to teach the resulting categories. Third, we describe two experiments that investigate the performance of the approach with untrained learners. Experiment 1 focuses on the last step of the pipeline, the selection of images, via teaching or the maximum likelihood data, to communicate the results to the user. Experiment 2 additionally manipulates model used to infer structure from the data, comparing mixture models with topic models. The results show that when the problem is difficult, computational models of teaching outperform methods based on the mean or maximum likelihood. Results also show that a more cogni-

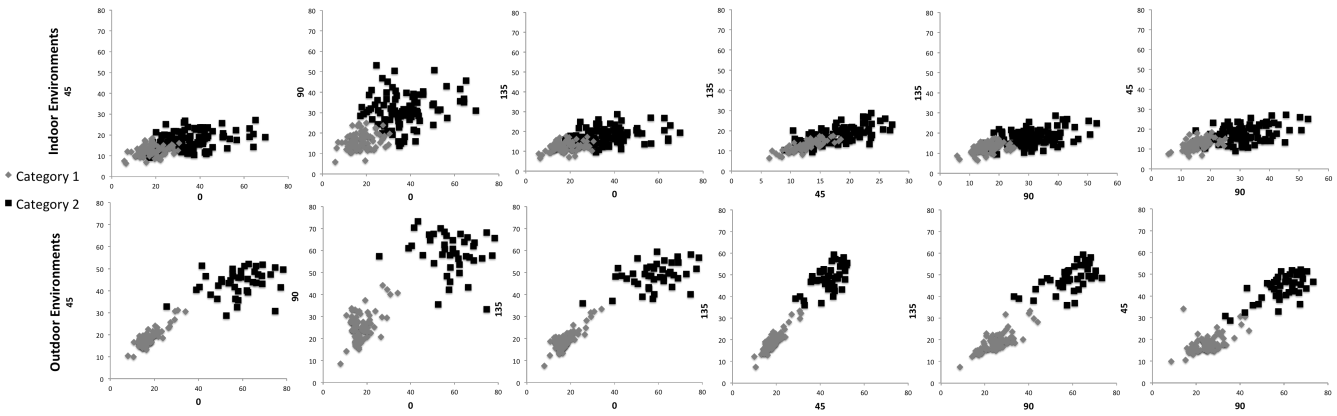


Figure 1: Scene category results. Orientation-orientation scatter plots of random samples from the target model. Different marker colors denote difference inner categories. The top row represents Indoor scenes and the bottom row represents Outdoor scenes. The indoor scene categories have considerably greater overlap than the natural scene categories.

tively natural representation of the domain—modeling scenes as mixture distributions—outperform a less cognitively natural, but otherwise effective, model in this visual domain.

2 Background

The natural-scene-category-teaching pipeline relies on findings from three literatures: natural scene perception, categorization, and computational models of teaching.

2.1 Natural scene perception

Natural scenes are semantically, structurally, and perceptually complex, and this complexity is decomposed by the human visual system, starting with low level features such as orientation. There is a characteristically biased distribution of oriented contours in natural scenes [1–3] and this anisotropy is reflected in the visual cortex at one of the earliest levels of visual processing [e.g. 4, 5]. Perception takes advantage of the regular anisotropy present in natural scenes [6] making it a logical structural property by which to quantify images as several previous image categorization approaches have done [7]. Therefore, it is sensible to quantify the structure of natural scenes in the orientation domain and determine if such structure can be taught to human users.

2.2 Categorization

There is a long history of behavioral research on human categorization (see [8, 9] for reviews). Anderson [10] [also 11] derived a model for learning an unknown number of categories, which was essentially a Dirichlet-process mixture model [12, 13]. Rasmussen [14] later proposed an efficient Gibbs sampling algorithm for this model. The Dirichlet process mixture model framework has since been widely adopted as a model of human category learning and in unsupervised machine learning, and has been used to model scene categories in images [7, 15]. Extending this previous work into communicating categories to human users is a logical next step.

2.3 Computational models of teaching

Computational models of teaching formalize the purposeful selection of examples whose goal is to enable the learner to infer the correct hypothesis [16–18]. Shafto & Goodman [16] introduced a Bayesian model of pedagogical data selection and learning, and used a simple teaching game to demonstrate that human teachers choose data consistently with the model and that human learners make stronger inferences from pedagogically-sampled data than from randomly-sampled data (data generated according to the true distribution; [19, 20]). More recently, Eaves Jr. *et al.* [21] employed advances in Monte Carlo approximation to facilitate tractable Bayesian teaching. Although enjoying considerable evidence in lab-based tasks where the target knowledge is selected by the experimenter, no prior work has investigated the possibility that this approach may be used to facilitate human learning from machine-derived knowledge.

3 Data analysis pipeline

3.1 Quantifying images

The first stage in the pipeline involves quantifying the complex information in an image. To do this, we extract the orientation information using a previously developed image rotation method [see 2]. In this method, each frame is rotated to the orientation of interest and the amplitude of the cardinal orientations (horizontal and vertical) extracted and stored via fast Fourier transform filtering. Repeating this process at different orientations allows each image to be condensed into a series of 4 (Experiment 1) or 36 (Experiment 2) data points representing the amount of oriented structure in the environment at four primary orientations 0, 45, 90, and 135 degrees in global (Experiment 1) or local (Experiment 2) image regions. We processed 200 images for both outdoor and indoor environments. The resulting four-orientation data can be seen in Figure 1.

The second stage involves inferring structure (categories) from the orientation data so that it may be taught. We compare two methods for image categorization: the infinite Gaus-

sian mixture model (IGMM) and latent Dirichlet allocation (LDA).

Infinite Mixtures

In experiment 1, we represent scenes as categories in continuous, multidimensional amplitude space. We model these categories as multidimensional Gaussians with mean μ and covariance matrix Σ. Learners must learn how many categories there are, their means and covariance matrices, and must infer of which category each datum is a member. We capture this with the *infinite Gaussian mixture model* framework [IGMM 14].

Infinite mixtures allow for as few as one or as many as n mixture components (categories). The IGMM infers an assignment, z, of data to categories, which is assumed to follow a Dirichlet process–in this work, the Chinese restaurant process–prior with concentration parameter α, CRP(α) [22]. The likelihood of the data, x, is then

$$\ell(x \mid \theta) = \prod_{i=1}^{n} \mathcal{N}(x_i; \mu_{z_i}, \Sigma_{z_i}). \tag{1}$$

where $\mathcal{N}(x; \mu, \Sigma)$ is the Gaussian (Normal) density of x given mean μ and covariance matrix Σ.

We place a conjugate, Normal inverse-Wishart prior on μ and Σ [23],

$$\Sigma \sim \text{Inverse-Wishart}_{\nu_0}(\Lambda_0^{-1}), \tag{2}$$

$$\mu|\Sigma \sim \mathcal{N}(\mu_0, \Sigma/\kappa_0). \tag{3}$$

Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) is a bag-of-words model for inferring the topics in corpora [24, 25]. Topic models have been adopted for use with images [7, 15], by treating each image as a document composed of visual *words* from a number of visual *topics*, T, which we treat as categories. To generate D documents from a W-word vocabulary under LDA with parameters $\alpha, \beta \in (0, \infty)$,

for all topics, $t \in 1, \ldots, T$ **do**
  $\phi_t \sim \text{Dirichlet}_W(\beta)$
end for
for all documents, $d \in 1, \ldots, D$ **do**
  $\theta_d \sim \text{Dirichlet}_T(\alpha)$
  for $i \in \{1, \ldots, w_d\}$ **do**
    $z \sim \text{Discrete}(\theta_d)$
    $w_{d|i} \sim \text{Discrete}(\phi_z)$
  end for
end for

The likelihood of a corpus, C, given $\Phi = \{\phi_1, \ldots, \phi_T\}$ under LDA is

$$\ell(C|T, \Phi, \alpha, \beta) = \sum_z \left[\left(\prod_{d=1}^{D} \text{DirCat}(z_d|\alpha) \right) \prod_{i=1}^{n} \phi_{w_i}^{(z_i)} \right], \tag{4}$$

where DirCat denotes the Dirichlet-categorical distribution, the sum over z denotes the sum over all possible assignments of the n words in the corpus to topics, z_d indicates the assignment of words in document d, and $\phi_{w_i}^{(z_i)}$ indicates the probability of the i^{th} word under the topic to which it is assigned, z_i.

3.2 Bayesian teaching

Teaching implies choosing data, x, that lead a learner to a specific hypothesis, θ, which we shall refer to as the *target*. In a Bayesian setting, teaching means choosing data in proportion with their induced posterior density:

$$p_T(x \mid \theta) = \frac{p_L(\theta \mid x)}{\int p_L(\theta \mid x) dx} \propto \frac{\ell(x \mid \theta)}{m(x)}, \tag{5}$$

where $\ell(x \mid \theta)$ is the likelihood of x under θ, $m(x) = \int \ell(x \mid \theta)\pi(\theta)d\theta$ is the marginal likelihood of x, and the subscripts T and L denote probabilities from the teacher's and learner's perspective, respectively.

Teaching infinite mixture models

Given data $x = x_1, \ldots, x_n$ we wish to teach the assignment of data to categories, z, and the category means and covariance matrices. The IGMM framework assumes that learner knows only the prior parameters (μ_0, Λ_0, ν_0, and κ_0) and that all other quantities are unknown.

The teacher's target model, θ, consists of K means and covariance matrices, and an n-length assignment of data to categories. To draw data from $p_T(x \mid \theta)$, we employ random-walk Metropolis sampling [26, 27]. An initial set of n data are drawn from the target model, after which new data, x', are proposed by adding Gaussian noise to x. The new data are accepted ($x := x'$) according to the acceptance probability:

$$p(x' \mid x) := \min[A, 1], \quad A = \frac{\ell(x' \mid \theta)m(x)}{\ell(x \mid \theta)m(x')}. \tag{6}$$

To search for $\text{argmax}_x p_T(x \mid \theta)$ one may employ *simulated annealing* [28] by replacing A with $A^{1/T}$, such that T goes to zero with the number of Metropolis steps.

Exploiting conjugacy, we can calculate $m(x)$ exactly for a small number of data by enumerating over the set of possible assignment vectors, $z \in Z$, and for each z calculating the product of the marginal likelihoods of the data in each component given the prior parameters:

$$m(x) = \sum_{z \in Z} \text{CRP}(z; \alpha) \prod_{k=1}^{K_z} f(x_i : z_i = k \mid \Lambda_0, \mu_0, \nu_0, \kappa_0), \tag{7}$$

where K_z is the number of mixture components in z.

Teaching topic models

The number of topics is known under LDA, so we need only teach the learner $\Phi = \{\phi_1, \ldots, \phi_T\}$. We do not teach the assignment of visual words to visual topics, z, and thus marginalize over all possible z. The marginal likelihood for LDA is

$$\sum_z \left(\prod_{d=1}^{D} \text{DirCat}(z_d \mid \alpha) \right) \left(\prod_{t=1}^{T} \text{DirCat}(\{w_i : z_i = t\} \mid \beta) \right). \tag{8}$$

There are T^n terms in the sum over assignments of words to topics, thus neither Equation 4 nor Equation 8 can be computed exactly for most real-word problems. We estimate these quantities using sequential importance sampling [e.g., 21, 29].

4 Experiments

Different types of visual experience were collected by wearing a head mounted camera (NET CMOS iCube USB 3.0; 54.9o X 37.0o FOV) which sent an outgoing video feed to a laptop. Videos were recorded as observers walked around different types of environments for variable amounts of time (a nature preserve, inside a house, down-town in a city, around a University, etc). Subsequently, every 500th frame of the videos was taken as a representative sample of a given video and sample images were sorted into purely natural, outdoor scenes (no man-made structure) or scenes from indoor experience.

To derive a target distribution (means and covariance matrices of subcategories), we applied expectation maximization [EM; 30] to the orientation data from each setting (see Figure 1). EM found two categories for both indoor and outdoor images. Although each image comprises information about the amplitude of structure at specific orientations, there were qualitative visual implications of the choice of images used for teaching (see Figure 2).

The target visual topic model for LDA taken from the LDA sampler state, Φ, at the 1000^{th} iteration of Gibbs sampling. The number of topics was set to 2 to match the number of categories in the IGMM target model. The parameters, α and β, were set to maximize the probability of the images under two topics.

4.1 General Methods

To determine if our teaching model better conveyed the environmental data to humans we ran a series of psychophysical categorization tasks. If the teaching model captures cognitively natural aspects of the selection of evidence for learning, then we would expect this group to perform better than those provided examples that capture the center (mean) of the category distribution. Rather than have subjects categorize all possible images from the distribution, we focused on images that should be difficult to categorize – ambiguous images that lie somewhere between the two categories. We compared categorization of ambiguous images based on either one of the three best teaching pairs or one of the three image pairs that captured the central tendency of each inner category (the mean for Experiment 1; or most likely under the model in Experiment 2). By using multiple pairs of images for comparison, we sought to eliminate any effects of idiosyncratic semantic content (i.e. filing cabinets) in individual images. Participants were recruited through Amazon Mechanical Turk and paid for completing the task. Using a completely on-line categorization task allowed us to test the optimality of teaching categories to untrained observers. Participants were presented with a machine-selected exemplar pair (see examples in Figure 2) and 24 sequentially presented ambiguous images which they were asked to categorize as either category one (left) or category 2 (right). At least one additional image was presented as an *attention check*; one of the exemplar images was presented as a image to be categorized to eliminate subjects who were not paying attention to the task. The data from any subject (n = 43 total) who failed to correctly categorize the attention checks was not used in further analyses.

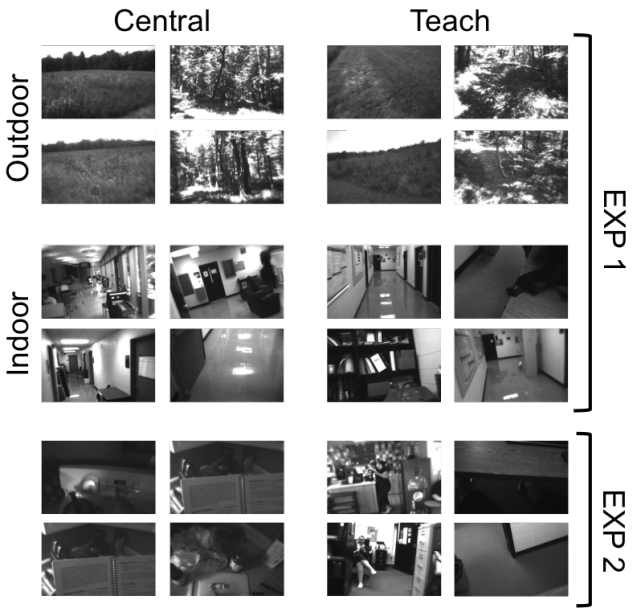


Figure 2: Examples of different exemplar pairs used in the categorization experiment for subject reference. The top row shows images used for outdoor scenes and the bottom row shows images used for indoor scenes. The left column shows the images that best capture the mean of the inner category distributions while the right column shows the example pairs picked by the model to teach the category.

4.2 Experiment 1

Experiment 1 focused on distinguishing indoor and outdoor scene types and determining if the teaching model provided better examples than images closest to the mean for each category. The ambiguous images in this experiment were chosen by calculating the Euclidean distance in orientation space each image lay from each inner category mean. The summed difference from each mean was then compared to the distance between the category means and the middle third of images closest to this value were labeled 'ambiguous'. A total of approximately 60 subjects were run in each of the 12 possible conditions (357 total). In order to minimize learning effects, the first four trials for trials were considered training and all results are based on performance on the last 18 images.

4.3 Results

In order to assess the results of the teaching model, we collapsed across the three exemplar pairs by first determining that there were no differences between them. Separate one-way ANOVAs were run and, while there were no differences for the indoor images, for outdoor images one pair in the teaching condition showed significantly lower performance than the other two runs $F(2, 124) = 54.26,\ p < .001$ (see Figure 3). Moreover, the standard deviation in this condition (.19) was more than twice any of the other 5 conditions (average = .093). Thirty percent of participants in this condition performed above chance level, correctly categorizing eleven images that were incorrectly categorized by most of

the participants who performed below chance. Interestingly, these eleven images were of open fields, which only the high-performers were grouping with the images selected for teaching that contained bodies of water. Our method for quantifying images based only on orientation content does not distinguish between fields and bodies of water (In Experiment 2 we explore a method that may map more closely to human perception by quantifying orientation in regions).

Subsequent analyses focus on the remaining two pairs of images for the teaching condition. To compensate for eliminating an entire pair of images from the outdoor condition, a random sample of equal size was extracted from the pooled total subjects who completed the task with indoor images. Further statistical analyses confirmed no differences across exemplar pairs between different runs within image type (Outdoor vs Indoor) and exemplar condition (Teach vs Mean) and thus the data from different exemplar pairs was pooled.

We tested categorization performance for the remaining teaching and mean pairs for outdoor and indoor images separately. Results plotted in Figure 3 show that participants were able to more easily categorize outdoor images than indoor images, $F(3, 319) = 20.12,\ p < .001$, which is consistent with the increased cluster separation of the outdoor categories (see Figure 1). Consequently, there were no differences in categorization performance based on the teaching exemplars as compared to the mean exemplars for outdoor images, $p = .90$; all participants performed well presumably because even the most ambiguous images were not especially difficult to categorize.

For indoor images, participants' categorization performance was significantly better for the teaching images relatively to the mean images, $p < .001$. These categories were less well separated in orientation space (see Figure 1). Consequently, the representative images selected for each category had a greater potential influence. Indeed, the teaching images, which are selected by the model to highlight the structure of the category *and* to contrast with the alternative category lead to better performance. Overall, the results of Experiment 1 indicate that for images whose categories are difficult to distinguish, the teaching model provides better exemplars for human category learners.

4.4 Experiment 2

Given the results of Experiment 1, Experiment 2 focused only on indoor images and used a higher-dimensional image quantification method. Inspired by the *spatial envelope* quantification of [7], we ran our global orientation analysis on nine sub regions of each image from Experiment 1. Each image is quantified into 36 orientation dimensions, corresponding to the 4 primary orientations (0, 45, 90, 135) in each of nine square image regions. We also sought to investigate whether the IGMM classification method outperformed Latent Dirichlet Allocation (LDA), which is less cognitively natural for categorization but is widely used for image analysis [7, 15]. The 36 dimensional data was fed directly into the IGMM, but was further quantized for the LDA model. Each image region was treated as a *word* and the *vocabulary* for indoor images was determined by K-means clustering. Each image

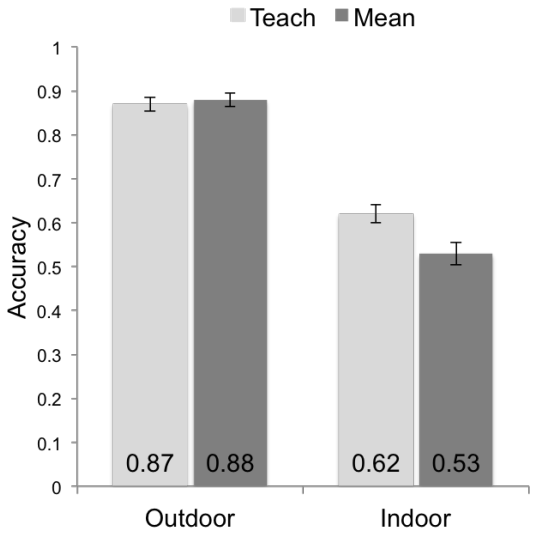


Figure 3: Results of psychophysical categorization experiments. Error bars represent two standard error of the mean.

contained nine regions summarized by four-dimensional continuous orientation data. The data from the nine regions of all of the two hundred images was assigned a cluster (word) by K-means (200 images times 9 data per image resulted in 1800 data to K-means). Elbow plots revealed that K=20 was the optimal vocabulary size. Each image was then a visual document composed of nine visual words from a 20-word vocabulary. To compare most directly with the IGMM we generated a target model with two topics by running LDA on the visual corpus. The images were then classified into one of two topics based on the higher percentage of words belonging to a given topic. Both the IGMM and LDA classification results were then fed into the teaching model to determine the best three image pairs for teaching each classification model. The highest likelihood pairs from each model were used for comparison. Ambiguous images were selected by finding their log-likelihood value under each category/topic (depending on the model), subtracting the two values, and finding the center third of images whose log likelihood difference score was closest to zero. This process led to 55 images under each model, 35 of which were identical across models. Approximately 23 subjects ran each of the 12 conditions for a total of 285 participants. Eleven were removed for incorrectly categorizing the attention checks. Preliminary analyses showed no learning affects and thus all 24 trials were included in the results.

4.5 Results

Again, we collapsed across the three exemplar pairs by first determining that there were no differences between those used within conditions (Teach vs. Likelihood). Separate one-way ANOVAs determined that the only significant difference between exemplar pairs was between pair 1 and pair 2 in the GMM likelihood condition ($F(2,68) = 3.59$, $p = 0.03$). However, neither pair 1 nor pair 2 was significantly different from pair 3 and thus all three exemplar pairs were collapsed into

overall teaching and likelihood conditions for each model. The overall 2 (model) by 2 (condition) way between subjects ANOVA showed significant main effects of teaching and model, but no interaction: $F(1,270) = 8.93$, $p = .003$, $F(1,270) = 24.87$, $p < .001$, and $F(1,274) = 0.93$, $p = 0.34$ respectively. As can be seen in Figure 4, the main effect of teaching is driven predominately by the higher accuracy scores in the GMM condition; the LDA condition shows no difference between teaching and likelihood exemplars. Accuracy scores were also significantly higher under the GMM model than the LDA model, in general. This suggests an additive improvement in performance for analysis with the mixture model and example selection with the teaching algorithm for these images on this task. Notably, performance in the GMM-teach condition is similar to that in Experiment 1, suggesting that characterizing images at the four orientations (cardinals and obliques) was sufficient to capture information about the orientation distribution in the images.

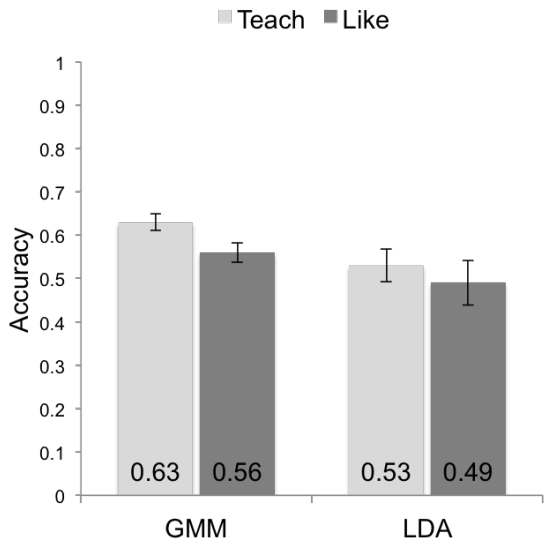


Figure 4: Results of Experiment 2. Error bars represent two standard error of the mean.

5 Conclusion

We presented an approach to optimizing the data analysis pipeline to minimize required expertise/training in data analytics in order to make informed decisions and increase accuracy. We leveraged known human competencies in perception, cognition and social learning as well as information classification methods from machine learning. We illustrated the approach in a domain and on a task where human competencies are well-investigated—scene perception and categorization, respectively—which allowed us to select established solutions to the problems of quantifying and analyzing the data. We presented an experimental investigation into the use of social learning methods, specifically a computational formalization of teaching, to provide a generic method of translating analytic results into human-understandable format. Because these experiments relied on the entire data analysis pipeline, our experiments necessarily tested both the efficacy of the computational model of teaching and the methods of quantifying and analyzing data.

Our results showed that human performance was significantly greater than chance for the two problems tested and that performance was related to the difficulty of the categorization. Results also showed that the computational teaching method performed well, exhibiting specific gains when the data analysis problem was hard. In this particularly difficult condition, the mean images failed to communicate the necessary distinction between categories. This failure demonstrates how a loss of information anywhere in the data analysis pipeline can indeed lead to incorrect outcomes. This work is a step forward in solving the problem of how to make choices along the pipeline such that users at the end are able to make informative decisions about the data without extensive training. It should be noted, however, that these results are specific to this domain of visual teaching with a relatively small sample size. The primary conclusion of this work is that complete automation and optimization of the data analysis pipeline is possible as long as one chooses a psychologically appropriate data model.

The results also highlighted known limitations in our data analysis pipeline. Specifically, we quantified images exclusively in terms of orientation content—one of the earliest steps of visual processing. This underestimates people's categorization abilities and our results revealed that while our approach performed better in general, there were specific cases where this quantification hampered decision making. Two notable instances are in the overall accuracy—we used only the most difficult images—and the case of semantic differences in outdoor images, which represent information that was not available to the models. Given the known limitations of this approach, we take this to be a promising negative result and an area for future work. Other areas for future work include generalization to domains that are less perceptually natural (e.g. radiography), to a broader array of representative decision making tasks, and exploring cases where analysis uncertainty is passed through to the decision maker. Regardless, our results indicate that creating data analysis pipelines based on human perceptual, cognitive, and social learning capacities is possible and a potentially fruitful direction for future research.

6 Acknowledgements

This work was support in part by NSF awards DRL-1149116 and CHS-1524888 to P.S. Undergraduate RA, Yunier Consuegra, programmed the Mechanical Turk experiments.

References

1. Hansen, B. C. & Essock, E. a. A horizontal bias in human visual processing of orientation and its correspondence to the structural components of natural scenes. *Journal of vision* **4,** 1044–1060 (2004).

2. Schweinhart, A. M. & Essock, E. a. Structural content in paintings: Artists overregularize oriented content of paintings relative to the typical natural scene bias. *Perception* **42,** 1311–1332 (2013).
3. Wainwright, M. J. Visual adaptation as optimal information transmission. *Vision Research* **39,** 3960–3974 (1999).
4. Furmanski, C. S. & Engel, S. A. An oblique effect in human primary visual cortex. *Nature neuroscience* **3,** 535–536 (2000).
5. Sun, P. *et al.* Demonstration of tuning to stimulus orientation in the human visual cortex: a high-resolution fMRI study with a novel continuous and periodic stimulation paradigm. *Cerebral Cortex,* bhs149 (2012).
6. Hansen, B. C. & Essock, E. A. A horizontal bias in human visual processing of orientation and its correspondence to the structural components of natural scenes. *Journal of Vision* **4,** 5–5 (2004).
7. Oliva, A. & Torralba, A. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision* **42,** 145–175 (2001).
8. Richler, J. J. & Palmeri, T. J. Visual category learning. *Wiley Interdisciplinary Reviews: Cognitive Science* **5,** 75–94 (2014).
9. Ashby, F. G. & Maddox, W. T. Human category learning. *Annu. Rev. Psychol.* **56,** 149–178 (2005).
10. Anderson, J. The adaptive nature of human categorization. *Psychological Review* **98,** 409 (1991).
11. Anderson, J. R. & Matessa, M. Explorations of an incremental, Bayesian algorithm for categorization. *Machine Learning* **9,** 275–308 (1992).
12. Antoniak, C. E. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The annals of statistics,* 1152–1174 (1974).
13. Ferguson, T. S. A Bayesian analysis of some nonparametric problems. *The annals of statistics,* 209–230 (1973).
14. Rasmussen, C. The infinite Gaussian mixture model. *Advances in neural information processing,* 554–560 (2000).
15. Sivic, J., Russell, B. C., Efros, A. A., Zisserman, A. & Freeman, W. T. *Discovering objects and their location in images* in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on* **1** (2005), 370–377.
16. Shafto, P. & Goodman, N. D. *Teaching games: Statistical sampling assumptions for learning in pedagogical situations* in *Proceedings of the Thirtieth Annual Conference of the Cognitive Science Society* (2008).
17. Shafto, P., Goodman, N. D. & Frank, M. C. Learning From Others: The Consequences of Psychological Reasoning for Human Learning. *Perspectives on Psychological Science* **7,** 341–351 (June 2012).
18. Shafto, P., Goodman, N. D. & Griffiths, T. L. A rational account of pedagogical reasoning: Teaching by, and learning from, examples. *Cognitive psychology* **71C,** 55–89 (Mar. 2014).
19. Bonawitz, E. *et al.* The double-edged sword of pedagogy: Instruction limits spontaneous exploration and discovery. *Cognition* **120,** 322–30 (Sept. 2011).
20. Zhu, X. Machine Teaching for Bayesian Learners in the Exponential Family. *Advances in Neural Information Processing Systems,* 1–9 (2013).
21. Eaves Jr., B. S., Schweinhart, A. & Shafto, P. in *Big Data in Cognitive Science* (ed Jones, M.) (Psychology Press, New York, NY, in press).
22. Aldous, D. Exchangeability and related topics. *Ecole dÉté de Probabilités de SaintFlour XIII* **1117,** 1–198 (1985).
23. Murphy, K. P. *Conjugate Bayesian analysis of the Gaussian distribution* tech. rep. (University of British Columbia, 2007).
24. Blei, D. M., Ng, A. Y. & Jordan, M. I. Latent Dirichlet Allocation. *Journal of Machine Learning Research* **3,** 993–1022 (2003).
25. Griffiths, T. L. & Steyvers, M. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America* **101 Suppl,** 5228–35 (2004).
26. Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. & Teller, E. Equation of state calculations by fast computing machines. *The journal of chemical physics* **21,** 1087–1092 (1953).
27. Hastings, W. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* **57,** 97–109 (1970).
28. Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P., *et al.* Optimization by simulated annealing. *science* **220,** 671–680 (1983).
29. Maceachern, S. N., Clyde, M. & Liu, J. S. Sequential importance sampling for nonparametric Bayes models: The next generation. *Canadian Journal of Statistics* **27,** 251–267 (1999).
30. Dempster, A. P., Laird, N. M. & Rubin, D. B. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the royal statistical society. Series B (methodological)* **39,** 1–38 (1977).

Sensory Cue Integration with High-Order Deep Neural Networks

Kyoung-Woon On, Eun-Sol Kim and Byoung-Tak Zhang
Department of Computer Science and Engineering
Cognitive Robotics Artificial Intelligence Center (CRAIC)
Seoul National University
Seoul 151-744, Korea
{kwon, eskim, btzhang}@bi.snu.ac.kr

Abstract

Humans can easily capture real world concepts from multi-modal signals by constructing joint representations of these signals. The joint representations may contain abstract information of multiple modalities and relationships across the modalities. Contrary to humans, it is not easy to obtain joint representations reflecting the structure of multi-modal data with machine learning algorithms, especially with conventional neural networks. This is because these models only have additive interaction across modalities. To deal with this issue, we propose a novel machine learning algorithm which captures multiplicative interaction between multiple modalities by using high-order edges. With these edges, the proposed method is able to learn the highly non-linear correlation among modalities. In the experimental results, we demonstrate the effect of this high-order interaction by showing improved results compared to other conventional models with a benchmark dataset, MNIST.

1 Introduction

The brain can easily perceive information of the environment from signals of multiple sensory modalities, including vision, audition and touch. This perception is facilitated by integrating of multiple sources of information efficiently, both within the senses and between them [Burr *et al.*, 2011]. Although there have been many research to imitate the multi-modal information processing mechanism of the brain, it still remains a difficult problem. This is because the information which consists of multiple input modalities has highly distinct statistical properties and each modality has a different kind of representation and correlational structure [Srivastava and Salakhutdinov, 2012]. Also, noise exists at every level of information processing, which makes the information unreliable and inaccurate [Ernst and Di Luca, 2011]. From this point of view, it is desirable to construct a multi-modal learning model which can discover the highly non-linear relationship across the modalities to replicate human-level perception.

A good multi-modal learning model should consist of competent joint representations that satisfy certain properties. It must be such that similarity in the representation space implies similarity of the corresponding 'concepts' [Srivastava and Salakhutdinov, 2012]. Furthermore, if there is noise in either modality which is uncorrelated with the other modality, then the joint representations are required to exclude this noise [Andrew *et al.*, 2013]. In addition, the joint representations ought to be useful for discriminative tasks.

In this paper, we develop a novel multi-modal learning model, High-order deep neural networks (HODNN). the HODNN combines the multi-modal data with high-order interaction to capture highly non-linear correlations among them.

In the following sections, we first explain the details of the HODNN. We then demonstrate the effect of high-order interaction in preliminary experiments. Finally we discuss our results and future work.

2 High-Order Deep Neural Networks

The HODNN connects abstract information of multiple modalities with high-order edges, which lead to a multiplicative interaction rather than just additive interaction used in conventional deep neural networks. This high-order interaction not only captures highly non-linear relationship across the modalities but also suppresses the uncorrelated noise efficiently. In addition, general deep structure is applied to each modality so as to obtain the balanced abstract information from each of it [Srivastava and Salakhutdinov, 2012].

Thus, the HODNN consists of two parts: modal-specific learning layers and a joint representations learning layers. The modal-specific learning layers have connections only within each modality, so the highest hidden layers of each modality represent abstract information of that modality (Figure 1a). The joint representations learning layers is composed of higher-order interactions between the joint hidden units and multiple groups of modal-specific hidden units. The joint hidden units can learn non-linear correlations among the modal-specific hidden units (Figure 1b).

In detail, the modal-specific learning layers follow a general neural networks framework. The hidden representations h_j^1 of the specific modality(Figure 1a) are obtained by Equation 1.

$$h_j^1 = \sigma(\sum_i W_{ij}^{v^1} v_i^1 + \textit{bias}) \tag{1}$$

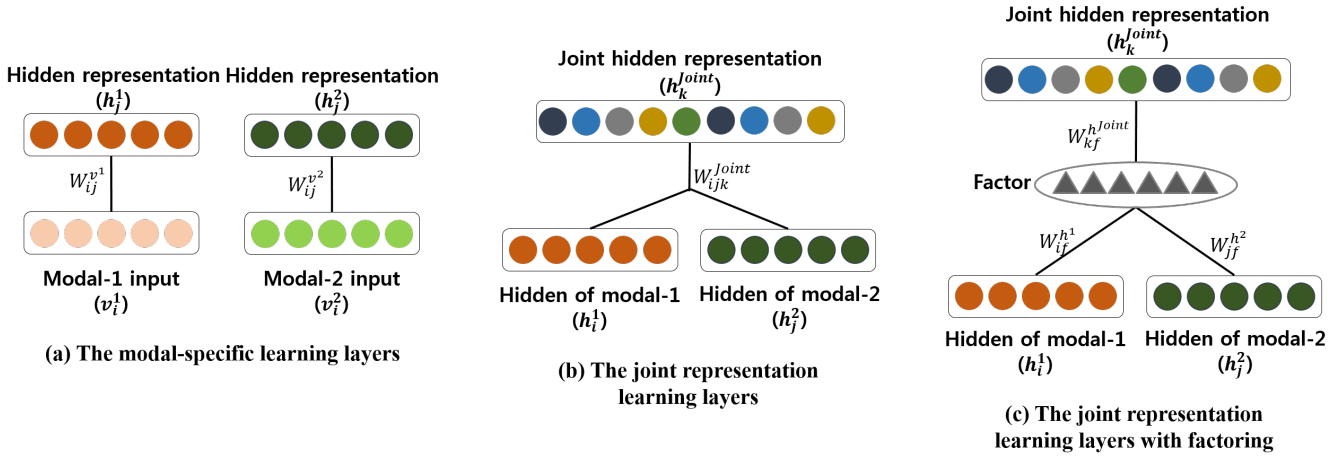


Figure 1: Building blocks of High-order deep neural networks. (a): Modal-specific learning layers, composed of RBM seperately. (b): Joint representations learning layers which capture correlations across the modalities. (c) Factorized version of joint the representations learning layers. The High-order deep neural networks (in Figure 2) stacks (a) and (c) to compose a deep architecture.

where $\sigma()$ is an activation function, which is a sigmoid function in our work.

In the joint representations learning layers, the weight is an $(n+1)$-way interaction tensor which connects n-groups of modal-specific hidden units and a group of joint hidden units. For simplicity, we show the case of a 2-modalities environment as an example, but it can be easily expanded across n-modalities case. The joint hidden representations h_j^{Joint} (Figure 1b) is obtained from Equation 2.

$$h_k^{Joint} = \sigma(\sum_{i,j} W_{ijk}^{Joint} h_i^1 h_j^2 + bias) \quad (2)$$

Motivated by [Memisevic and Hinton, 2010], the multi-way

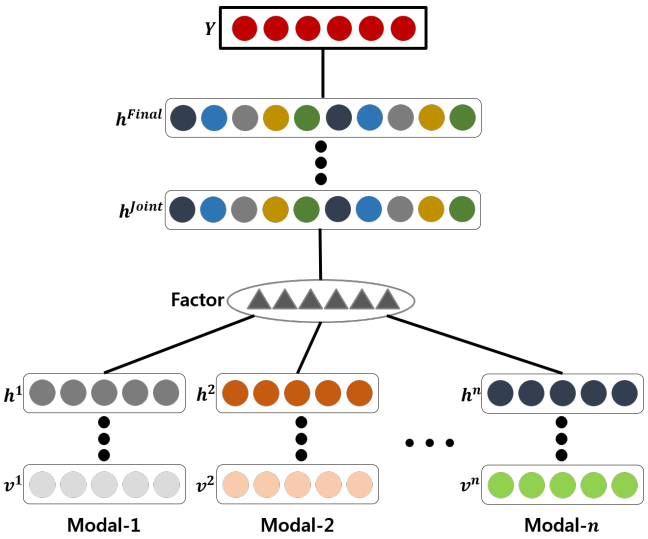


Figure 2: Architecture of High-order deep neural networks

factoring method is employed to maintain efficient model complexity without heavily loosing the capacity of the model (Figure 1c). As a consequence, we can obtain the joint hidden representations h_k^{joint} as follows:

$$h_k^{Joint} = \sigma\left(\sum_f W_{kf}^{h^{Joint}} (\sum_i W_{if}^{h^1} h_i^1)(\sum_j W_{jf}^{h^2} h_j^2) + bias\right) \quad (3)$$

The overall HODNN architecture is illustrated in Figure 2. In the next sections, we explain how the model can be learned by a general deep learning framework.

3 Learning

As for most of the deep neural networks, HODNN adopts a strategy of fine-tuning with backpropagation after pre-training each module. Since the training method of the modal-specific learning layers can easily be derived in the same way, in this section, we only focus on how the joint representations learning layers can be trained.

3.1 Learning: Pre-training

With the presented architecture and activation rules of the joint hidden units of the HODDN, a high-order Boltzmann machine can be used as a building block for the joint representations learning layers [Sejnowski, 1986]. For clarity, we use the notation in Figure 1 and refer to h^1 and h^2 as visible units and h^{Joint} as hidden units in order to see joint representations learning layers as an isolated high-order Boltzmann machine. The high-order Boltzmann machine connects more than two layers using weight tensor W^{Joint}. Then the joint probability distribution of $P(h^1, h^2, h^{Joint})$ is defined as follow:

$$P(h^1, h^2, h^{Joint}) = \frac{1}{Z} exp(-E\left(h^1, h^2, h^{Joint}\right)),$$
$$Z = \sum_{h^1, h^2, h^{Joint}} exp\left(-E(h^1, h^2, h^{Joint})\right) \quad (4)$$

In Equation 4, the energy function $E(h^1, h^2, h^{Joint})$ is given by:

$$-E(h^1, h^2, h^{Joint}) = \sum_{ijk} W_{ijk}^{Joint} h_i^1 h_j^2 h_k^{Joint} + \sum_i b_i^{h^1} h_i^1 + \sum_j b_j^{h^2} h_j^2 + \sum_k b_k^{h^{Joint}} h_k^{Joint} \quad (5)$$

where the b indicates the bias of each layer.

To train the model, the negative log-likelihood need to be minimized by using gradient descent. The derivative of the negative log-likelihood with regard to parameter θ is given by:

$$-\frac{\partial ln L(\theta|h^1, h^2)}{\partial \theta} = \sum_{h^{Joint}} P(h^{Joint}|h^1, h^2) \frac{\partial}{\partial \theta} E(h^1, h^2, h^{Joint}) - \sum_{h^1, h^2, h^{Joint}} P(h^1, h^2, h^{Joint}) \frac{\partial}{\partial \theta} E(h^1, h^2, h^{Joint}) \quad (6)$$

Like a standard RBM, the first term (visible expectation) is easily computed but the second term (model expectation) is intractable. The contrastive divergence (CD) is an useful method to approximate the second term [Hinton, 2002]. However, the conditional dependency of the visible units h_i^1 and h_j^2 given hidden units h^{Joint} precludes to utilize CD as it is. Therefore, a 3-way sampling method is applied by using Equation in 7, 8 and 9 [Reed *et al.*, 2014; Susskind *et al.*, 2011].

$$P(h_k^{Joint} = 1|h^1, h^2) = \sigma(\sum_{i,j} W_{ijk}^{Joint} h_i^1 h_j^2 + b_k^{h^{Joint}}) \quad (7)$$

$$P(h_i^1 = 1|h^{Joint}, h^2) = \sigma(\sum_{j,k} W_{ijk}^{Joint} h_k^{Joint} h_j^2 + b_i^{h^1}) \quad (8)$$

$$P(h_j^2 = 1|h^{Joint}, h^1) = \sigma(\sum_{i,k} W_{ijk}^{Joint} h_k^{Joint} h_i^1 + b_j^{h^2}) \quad (9)$$

As noted earlier, the factoring method is employed to prevent the number of parameter W_{ijk}^{Joint} from growing exponentially. With this factoring method, the 3-dimensional weight tensor W_{ijk}^{Joint} is replaced by the multiplication of three 2-dimensional weight matrices $W_{if}^{h^1}$, $W_{jf}^{h^2}$ and $W_{kf}^{h^{Joint}}$:

$$W_{ijk}^{Joint} = \sum_f W_{if}^{h^1} W_{jf}^{h^2} W_{kf}^{h^{Joint}} \quad (10)$$

There are several research which take a similar approach to ours in terms of the models [Reed *et al.*, 2014; Susskind *et al.*, 2011; Nguyen *et al.*, 2015], but these studies do not cover the multi-modal environment and the deep neural networks architecture.

3.2 Learning: Fine-tuning

The backpropagation algorithm is well known to fine-tune the deep neural network. In this section, we demonstrate that the suggested model can be trained using backpropagation by providing the following derivative equations $\frac{\partial h_k^{Joint}}{\partial h_i^1}$, $\frac{\partial h_k^{Joint}}{\partial W_{kf}^{h^{Joint}}}$ and $\frac{\partial h_k^{Joint}}{\partial W_{if}^{h^1}}$:

$$\frac{\partial h_k^{Joint}}{\partial h_i^1} = h_k^{Joint}(1 - h_k^{Joint}) \times \left(\sum_f W_{kf}^{h^{joint}} W_{if}^{h^1} (\sum_j W_{jf}^{h^2} h_j^2) + b_i^{h^1} \right) \quad (11)$$

$$\frac{\partial h_k^{Joint}}{\partial W_{kf}^{h^{Joint}}} = h_k^{Joint}(1 - h_k^{Joint}) \times (\sum_i W_{if}^{h^1} h_i^1)(\sum_j W_{jf}^{h^2} h_j^2) \quad (12)$$

$$\frac{\partial h_k^{Joint}}{\partial W_{if}^{h^1}} = h_k^{Joint}(1 - h_k^{Joint}) \times (\sum_f W_{kf}^{h^{joint}} h_i^1 (\sum_j W_{jf}^{h^2} h_j^2)) \quad (13)$$

Using Equation 11,12 and 13, we can easily obtain the derivative equation in terms of all weight parameters for backpropagation.

4 Preliminary experiment: Effect of high-order interaction

As preliminary experimental results, we show three results with MNIST dataset. To focus on the effect of high-order interaction, only the joint representations learning layers of HODNN, a factored high-order Boltzmann machine, is used (Figure 3). Also, the MNIST dataset is utilized which consists of hand written digit images and the corresponding labels. Each image contains 28×28 gray scale pixel values and each label is presented with 10-dimensional one-hot encoding vector. While the label vectors are usually used as targets in discriminative tasks, in our experiments, the image and the label vectors are used as two different modalities, the former is for visual information and the latter is for textual information which indicate the concept 'number'. Thus, as the two input vectors have highly non-linear relationships, we apply our model to learn the useful representations by capturing such relationships. To demonstrate this effect, we conducted three comparative experiments:

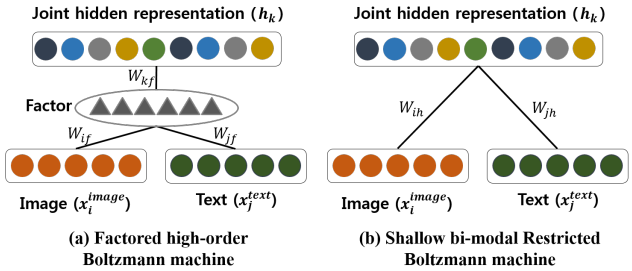


Figure 3: Comparative models for demonstrating the effect of high-order interaction

Figure 4: Weight vector pairs of randomly selected 25 factors out of 100. The upper rows are 28×28 dimensions image filters W_{if}^{image} and the lower rows are 10 dimensions text filters W_{jf}^{text}. The brighter color of filter indicates a higher value of weight.

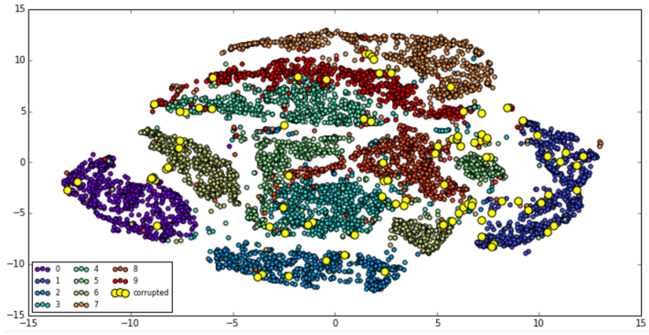
- Show the learning weight that represents correlational structure.
- Explore the learned joint hidden representations space.
- Prove robustness to noise.

To show the competence of our model, a shallow bi-modal RBM (Figure 3b) is used as a comparative model. The shallow bi-modal RBM also functions as a module combining different modalities in conventional multi-modal deep networks [Srivastava and Salakhutdinov, 2012; Ngiam *et al.*, 2011]. For impartiality, same architecture and hyper-parameters (200 hidden units, and 0.001 learning rate) are used for both module. Also, we used 100 factors for factored high-order Boltzmann machine. it is worth noticing that our model has lower complexity than the comparative model. (the number of weights of shallow bi-modal RBM: $200 \times (784 + 10)$, the number of weights of factored high-order Boltzmann machine: $200 \times 100 + 784 \times 100 + 10 \times 100$)

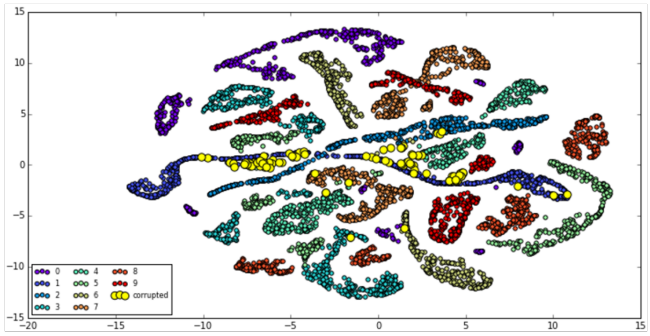
4.1 Learning weight that represent correlational structure

Factorization leads to a matching of weight (filter) responses (W_{if}, W_{jf} in Figure 3a). The observed weight pairs from the trained model can be used to determine how well the model learns the heterogeneous multi-modal inputs. To look closely, we visualized only 25 randomly selected weight vector pairs of factors out of 100 factors (Figure 4). In Figure 4, each pair of weight vectors consists of two sub images. Upper large images represents image filters which size is 28×28 and lower small images represents 10-dimensional text filters. If an image filter mainly passes visual information of a certain number, the corresponding text filter also tends to pass the textual information of that number. In addition, there are some filter pairs that behave in an opposite way. However, it is obvious that both of the tendencies support the fact that the model can effectively capture the relationship between the two heterogeneous inputs.

Figure 5: Some examples of 100 corrupted images. The corrupted images are generated by mixing of randomly selected image of number 1 and other numbers



(a) 2D t_SNE visualization of learned representations of the Shallow bi-modal RBM with corrupted image



(b) 2D t_SNE visualization of learned representations of the Factored high-order Boltzmann machine with corrupted image

Figure 6: 2D t_SNE visualization of learned representation of both models using corrupted image information and clear text information. (a) is for shallow bi-modal RBM and (b) is for factored high-order Boltzmann machine.

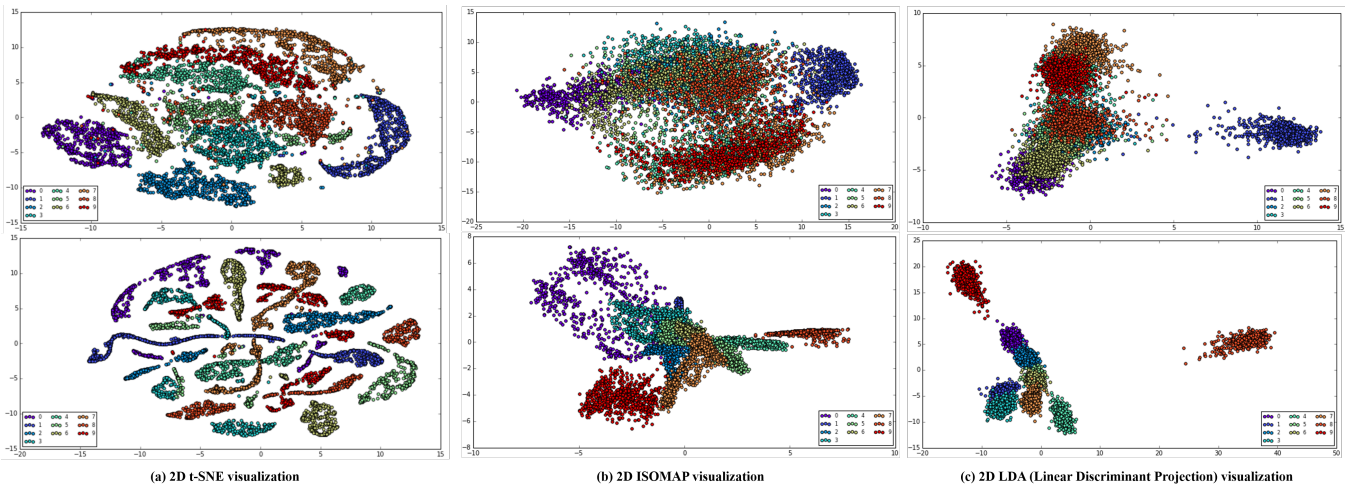


Figure 7: Comparison of learned representations between shallow bi-modal RBM and factored high-order Boltzmann machine. The upper row shows the shallow bi-modal RBM and lower row shows the factored high-order Boltzmann machine. (a): 2D t-SNE visualization, (b): 2D ISOMAP visualization and (c): 2D LDP(Linear Discriminent Projection) visualization.

4.2 Learned joint hidden representations space

In order to see how the joint hidden units represent the abstract information across the modalities, 2D embedding algorithm is applied to hidden units of both factored high-order Boltzmann machine and shallow bi-modal RBM .To ensure the generality, various techniques are used, which are t-SNE, ISOMAP and Linear discriminant projection. Figure 7 shows the embedding results of both models. All three results of factored high-order Boltzmann machine are more visually discriminative than those of the shallow bi-modal RBM. Also, it is interesting to notice that the t-SNE embedding result of the joint hidden representations with factored high-order Boltzmann machine looks as if the representations of each number form their own sub-manifold structure. These interpretations imply that the representations power of factored high-order Boltzmann machine is greater than the that of shallow bi-modal RBM.

4.3 Robustness to noise

To demonstrate how the high-order interaction efficiently cancels out the noise of either modality which is uncorrelated with other modality, it is appropriate to compare the joint hidden representations of both models when the noisy input is fed in. For this experiment, we firstly generated a corrupted dataset which consists of the 100 image inputs of number 1 corrupted by other numbers and corresponding clear text inputs of number 1 (Figure 5). The joint hidden representations of the corrupted dataset are shown with normal dataset by using 2D t-SNE visualization 6. The big yellow circle is for the corrupted dataset and others are for the normal dataset. Figure 6a shows the case using shallow bi-modal RBM and Figure 6b shows the case using factored high-order Boltzmann machine. As expected, in the factored high-order Boltzmann machine, the representations of the corrupted dataset are located near the representations of number 1. However, in the shallow bi-modal RBM, the representations of corrupted dataset lie scattered across the numbers. It reveals the property of high-order interaction that remove the uncorrelated noise information of either modality based on the other modality.

5 Conclusion

In this paper, we suggest the High-order deep neural networks for learning representations of multi-modal data which has highly non-linear relationships. We performed three preliminary experiments to demonstrate the effect of high-order interaction. Based on experimental results, we conclude that the high-order interaction can capture not only intra-relationships of multiple modalities but also inter-relationships.

In future works, we aim to apply our model to the study of event cognition which have emerged over the last years as a vibrant topic of scientific study. It is an important research topic because much of our behavior is guided by our understanding of events which is what happens to us, what we do, what we anticipate and what we remember in our daily life [Radvansky and Zacks, 2014]. For better understanding of the study, we employ multiple wearable sensors to record the daily-life of a person. This is because the collected data from the viewpoint of the first-person plays a significant role in learning of human behavior [Zhang, 2013; Kim *et al.*, 2016; Lee *et al.*, 2016]. Through this study, we hope to suggest an event cognition model which can perceive real-time events in real-life by using multiple wearable sensors.

References

[Andrew *et al.*, 2013] Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. Deep canonical correlation analysis. In *Proceedings of the 30th International Conference on Machine Learning*, pages 1247–1255, 2013.

[Burr *et al.*, 2011] David Burr, Paola Binda, and Monica Gori. Multisensory integration and calibration in adults

and in children. *Sensory Cue Integration. Oxford University Press, Oxford*, pages 173–194, 2011.

[Ernst and Di Luca, 2011] Marc O Ernst and Massimiliano Di Luca. Multisensory perception: from integration to remapping. *Sensory cue integration (Trommershauser J, Körding KP, Landy MS, eds). Oxford University Press, Oxford*, pages 224–250, 2011.

[Hinton, 2002] Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

[Kim *et al.*, 2016] E.-S. Kim, K.-W. On, and B.-T. Zhang. Deepschema: Automatic schema acquisition from wearable sensor data in restaurant situations. In *Twenty-Fifth International Joint Conference on Artificial Intelligence*, 2016.

[Lee *et al.*, 2016] S.-W. Lee, C.-Y. LEE, D.-H. Kwak, J. Kim, and B.-T. Zhang. Dual-memory deep learning architectures for lifelong learning of everyday human behaviors. In *Twenty-Fifth International Joint Conference on Artificial Intelligence*, 2016.

[Memisevic and Hinton, 2010] Roland Memisevic and Geoffrey E Hinton. Learning to represent spatial transformations with factored higher-order boltzmann machines. *Neural Computation*, 22(6):1473–1492, 2010.

[Ngiam *et al.*, 2011] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. Multimodal deep learning. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 689–696, 2011.

[Nguyen *et al.*, 2015] Tu Dinh Nguyen, Truyen Tran, Dinh Q Phung, and Svetha Venkatesh. Tensor-variate restricted boltzmann machines. In *AAAI*, pages 2887–2893, 2015.

[Radvansky and Zacks, 2014] Gabriel A Radvansky and Jeffrey M Zacks. *Event cognition*. Oxford University Press, 2014.

[Reed *et al.*, 2014] Scott Reed, Kihyuk Sohn, Yuting Zhang, and Honglak Lee. Learning to disentangle factors of variation with manifold interaction. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1431–1439, 2014.

[Sejnowski, 1986] Terrence J Sejnowski. Higher-order boltzmann machines. In *AIP Conference Proceedings*, volume 151, pages 398–403, 1986.

[Srivastava and Salakhutdinov, 2012] Nitish Srivastava and Ruslan R Salakhutdinov. Multimodal learning with deep boltzmann machines. In *Advances in neural information processing systems*, pages 2222–2230, 2012.

[Susskind *et al.*, 2011] Joshua Susskind, Roland Memisevic, Geoffrey Hinton, and Marc Pollefeys. Modeling the joint density of two images under a variety of transformations. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2793–2800. IEEE, 2011.

[Zhang, 2013] B.-T. Zhang. Information-theoretic objective functions for lifelong learning. In *AAAI Spring Symposium: Lifelong Machine Learning*, pages 62–69. Citeseer, 2013.

Using an AI Agent and Coordinated Expert Sourcing to Construct Content for a Dialog System

Matthew Davis, Marco Crasso, Yasaman Khazaeni, Praveen Chandar,
Dmitri Levitin, Werner Geyer
IBM Research, Cambridge US
{davismat, yasaman.khazaeni, pcravich, dleviti, werner.geyer} @us.ibm.com, crasso@ar.ibm.com

Abstract

Despite the utility of AI agent systems for question and answer in domains such as customer service, the cost of human effort to establish and maintain the knowledge underlying these systems remains high. In this work, we introduce a process to radically reduce investment of human time required to generate and maintain the data necessary for implementing such a system. We demonstrate the utilization of different type of data sources in building a knowledge base for the AI agent using machine learning techniques. In cases where human expert verification is required for confidence in the system, the AI agent collaborates with the human expert to verify reliable knowledge in the AI agent's own dialog system.

1 Background

1.1 Conversational QA Systems

Artificial Intelligence agents are increasingly efficacious for question-and-answer in domains such as customer service, see Fader, Zettlemoyer, and Etzioni (2014); Tan et al. (2000). Agents must understand user questions, retrieve useful information, and relay answers to users. A few underlying data are required to implement such agents. Natural language understanding can be approached with techniques as simple as string matching or as complex as deep learning classification, but in any case requires many variations of the user question. Dialog systems can be deterministic or probabilistic, but in either case require a mapping between the discerned user question and the correct answer. Finally, the answer itself can reside in a preprocessed map of questions and answers, or within a corpus of curated documents from which the answer is retrieved, some background and examples can be found in Ogan et al. (2012); Wong et al. (2012) and references therein.

Each of these data require large investments of human effort to create and maintain. However, even if an automated approach to generating these data were available, there would be many instances where human review of the data would be required. In some cases, the accuracy of the system may be insufficient without human review, and in others, there may be regulatory requirements for human review.

We propose a method for automated construction and maintenance of the data required for such an agent system. Our approach combines machine learning, unsupervised topic modeling, and information retrieval given source corpora containing candidate questions and answers. Further, to address the cases where human review is required, we describe a method for including the AI agent in the construction, verification, and maintenance of its own data resources by requesting review from a human expert. Our approach aims to minimize the time required of the human expert while ensuring accuracy in the system. For some related work see Clark et al. (2003); Nass, Steuer, and Tauber (1994).

1.2 Generation of Knowledge

The agent system relies on multiple corpora for knowledge. Potential user questions represent one corpus. Within each user question, a set of variations for matching the associated user question is required, and these collectively represent a second corpus. The corpus of answers can either be specifically coded to correspond to match questions, or may be an unstructured corpus from which answers are retrieved.

Here, we consider a case from a customer service center providing an existing unstructured corpus containing historical customer service tickets as well as an unstructured corpus of reference documents intended for human reference. From these corpora, our method generates the required corpus of potential user questions, the variations of those questions, and their answers. For cases where probabilistic answers are acceptable, the AI agent can use these to directly in answering user questions. It is critical for maintenance of these data to monitor updates the reference corpus and to re-evaluate the question and answer entries that could be affected by the updates. In cases where human expert supervision is required, our method identifies the best expert and requests their supervision via the AI agent interface. Here the role of the AI agent reduces the cost for the human expert in identifying new and trending questions, researching them, and verifying their answers.

2 Methodology

2.1 Dialog System

For natural language understanding, we employ a multilayer convolutional neural network for multi-label classification which identifies the most likely question entry in a deterministic dialog tree, (Krizhevsky, Sutskever, and Hinton (2012)). The CNN labels are trained on variations of each question. We describe the initial generation of these labels below. The deterministic dialog tree maps via its associated CNN label to an associated answer. If an answer has not yet been associated with that label, the system employs a search strategy to retrieve documents that are likely to contain the answer from the reference corpus (see below for the direction of these documents to the human expert).

2.2 Knowledge Base

Our method requires the existence of a reference corpus and a corpus of historical inquiries. The reference corpus typically consists of unstructured text in the form of reference documents intended for human interpretation. These could be in the form of policy documents, web pages, or reference emails. In our customer service center case, the historical inquiry corpus consists of previous tickets from customer call and email inquiries. Our method aims to use the data housed and obscured in these corpora to automate the generation of the CNN label for natural language understanding as well to identify the correct answer to a given user question.

2.2.1 Topic Modeling

The historical inquiry corpus is the source used to generate question variations. Our method uses Latent Dirichlet Allocation (LDA) topic modeling (Blei, Ng, and Jordan (2003)) to infer the topics that are represented in the all the tickets. We apply LDA using the Mallet toolbox (McCallum (2002)) in a recursive manner where the tickets are first divided into several general topics to find sub-topics in each partition. Hierarchical clustering and topic models for documents have been looked at extensively in the literature with different applications, see Blei, Griffiths, and Jordan (2010); Bot et al. (2005); Wang et al. (2011). The LDA topic labels themselves can be used as CNN labels, as they encapsulate a group of relevant tickets into a set of sorted keywords. The representative labels in the CNN are not required to be a natural language question themselves, but they provide a descriptive label for the question variations associated with that label. Next, we apply the LDA inference model to label all sentences in the tickets. In this way, we deconvolve the multiple topics of each ticket to the level of their constituent sentences. Next, we hierarchically cluster these sentences in the space of their LDA topic vectors. Many of the resulting clusters correspond the LDA topics directly, and a few correspond to convolutions of those topics. Critically, this clustering approach allows us to identify sentences that represent clear examples of the question posed in the topic while simultaneously sifting out off-topic (e.g. greetings) and perfunctory language (e.g. metadata tags) to separate clusters. The relevant clusters of sentences then represent natural language variations for the training of the CNN. As an example in Fig. 1 we see a collection of around 260 sentences from 10 topics clustered into 15 different clusters. Sentences are represented as rows and columns show different topics. Red corresponds to higher relevance and blue is low relevance to that topic.

The LDA model is also used in conjunction with an inverted index of the reference corpus to retrieve documents that likely contain answers to the user question. Thus we can populate the required elements of a new dialog tree entry: the question label, its variations, and an answer. However, in some cases, we may prefer human verification of these elements.

2.2.2 Expert-sourcing

Given access to a human expert, our method leverages the AI agent as a social actor seeking expertise assistant in the same manner that a human agent working a customer service center would seek expertise in the case of ticket escalation. The AI agent can present the expert (for example via email or an interactive prompt or web interface) with a request stating the identification of a new question that needs an answer. In addition to the top-scoring variations identified in the topic modeling, the agent can retrieve the original context of these sentences from their ticket in the historical inquiry corpus. To minimize research time required by the expert, the AI agent can also provide any documents retrieved from the reference corpus. Thus the expert's task is reduced to formulation of a bona fide answer to the question represented. This process closes the loop of generating trusted answers to the questions provided to the customer with minimal investment of human supervision. An schematic presentation of the process is shown in Fig. 2.

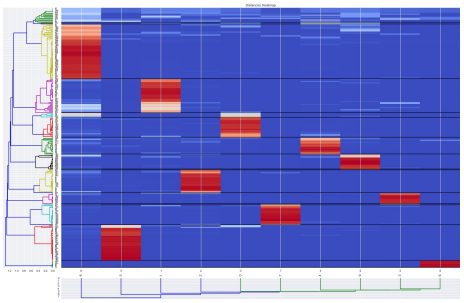


Figure 1: Hierarchical Clusters of Sentences from 10 Topics

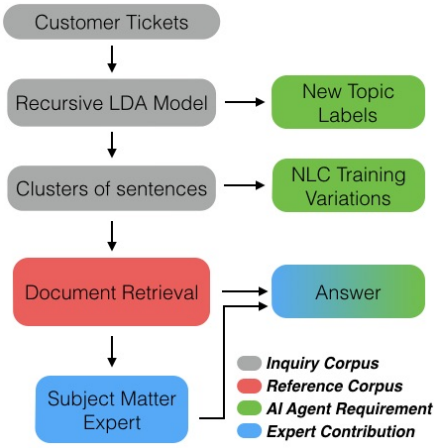


Figure 2: Schematic of the information retrieval process

3 Discussion

We have described a process of radically reducing investment of human time required to generate and maintain the data necessary for implementing an AI agent for question and answer. We hope to refine and generalize this work in a number of ways, including implementing optimal algorithms for selection of topic models and cluster, novel strategies for information retrieval from the reference corpus, event detection for trending questions in a continuously updated inquiry corpus, and evaluation of a system resulting from the method in comparison to manually constructed and curated knowledge bases.

References

Blei, D. M.; Griffiths, T. L.; and Jordan, M. I. 2010. The nested Chinese restaurant process and Bayesian nonparametric in-

ference of topic hierarchies. *Journal of the ACM* 57(2):1–30.

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research* 3(4-5):993–1022.

Bot, R. S.; Wu, Y.-f. B.; Chen, X.; and Li, Q. 2005. Generating better concept hierarchies using automatic document classification. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, CIKM '05, 281–282. New York, NY, USA: ACM.

Clark, P.; Chaudhri, V.; Mishra, S.; Thoméré, J.; Barker, K.; and Porter, B. 2003. Enabling domain experts to convey questions to a machine: A modified, template-based approach. In *Proceedings of the 2Nd International Conference on Knowledge Capture*, K-CAP '03, 13–19. New York, NY, USA: ACM.

Fader, A.; Zettlemoyer, L.; and Etzioni, O. 2014. Open question answering over curated and extracted knowledge bases. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, 1156–1165. New York, NY, USA: ACM.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In Pereira, F.; Burges, C. J. C.; Bottou, L.; and Weinberger, K. Q., eds., *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc. 1097–1105.

McCallum, A. K. 2002. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

Nass, C.; Steuer, J.; and Tauber, E. R. 1994. Computers are social actors. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, 72–78. ACM.

Ogan, A.; Finkelstein, S.; Mayfield, E.; D'Adamo, C.; Matsuda, N.; and Cassell, J. 2012. Oh Dear Stacy!: Social interaction, elaboration, and learning with teachable agents. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 39–48. ACM.

Tan, P.-N.; Blau, H.; Harp, S.; and Goldman, R. 2000. Textual data mining of service center call records. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '00, 417–423. New York, NY, USA: ACM.

Wang, D.; Zhu, S.; Li, T.; Chi, Y.; and Gong, Y. 2011. Integrating document clustering and multidocument summarization. *ACM Trans. Knowl. Discov. Data* 5(3):14:1–14:26.

Wong, W.; Cavedon, L.; Thangarajah, J.; and Padgham, L. 2012. Mixed-initiative conversational system using question-answer pairs mined from the web. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, 2707–2709. New York, NY, USA: ACM.

KR^3L: An Architecture for Knowledge Representation, Reasoning and Learning in Human-Robot Collaboration

Mohan Sridharan
Electrical and Computer Engineering
The University of Auckland, New Zealand
m.sridharan@auckland.ac.nz

Abstract

This paper describes an architecture that combines the complementary strengths of declarative programming, probabilistic graphical models, and reinforcement learning. Reasoning with different descriptions of incomplete domain knowledge and uncertainty is based on tightly-coupled representations at two different resolutions. For any given goal, non-monotonic logical inference with the coarse-resolution domain representation provides a plan of abstract actions. Each abstract action is implemented as a sequence of concrete actions by reasoning probabilistically over a relevant part of the fine-resolution representation, committing high probability beliefs to the coarse-resolution representation. Unexplained plan step failures trigger relational reinforcement learning for incremental and interactive discovery of domain axioms. These capabilities are illustrated in simulated domains and on a physical robot in an indoor domain.

1 Introduction

Consider a robot assisting humans in locating and moving objects to specific places in an office with multiple rooms. While the robot typically needs considerable domain knowledge to perform these tasks, it is difficult for humans to provide comprehensive domain knowledge. The robot may be equipped with some commonsense knowledge, e.g., "books are usually in the library", and exceptions to this knowledge, e.g., "cookbooks are in the kitchen". In addition, the robot's actions are non-deterministic, and any information extracted from sensor inputs is likely to be incomplete and unreliable. To assist humans in such domains, the robot thus has to represent knowledge, reason, and learn, at both the sensorimotor level and the cognitive level. This objective maps to some fundamental challenges in knowledge representation, reasoning, and learning. For instance, the robot has to encode and reason with commonsense knowledge such that the semantics are readily accessible to humans, while also quantitatively modeling the uncertainty in sensing and actuation. Furthermore, for computational efficiency, the robot has to tailor sensing and actuation to tasks at hand, incrementally and interactively revising its knowledge over time. As a step towards addressing these challenges, the architecture described in this paper combines the knowledge representation and non-monotonic logical reasoning capabilities of declarative programming, with the uncertainty modeling capabilities of probabilistic graphical models, and the incremental and interactive learning capability of reinforcement learning. Key features of this architecture are:

- An action language describes transition diagrams of the domain at two resolutions, with the fine-resolution diagram being a *refinement* of the coarse-resolution diagram.
- For any given goal, non-monotonic logical reasoning with the coarse-resolution commonsense knowledge provides a tentative plan of abstract actions.
- Each abstract action is implemented probabilistically as a sequence of fine-resolution concrete actions, by *zooming* to the relevant part of the fine-resolution diagram and constructing suitable data structures.
- Unexplained plan step failures trigger incremental and interactive discovery of previously unknown domain axioms using relational reinforcement learning.

In our architecture, we translate the coarse-resolution representation to an Answer Set Prolog (ASP) program, and construct a partially observable Markov decision process (POMDP) for probabilistic planning. The architecture has been demonstrated to support reasoning with violation of defaults, noisy observations, and unreliable actions in large, complex domains [Colaco and Sridharan, 2015; Zhang *et al.*, 2015; Sridharan *et al.*, 2016; Zhang *et al.*, 2014]. Here, we summarize the technical contributions, and some results of experimental trials in simulation and on a mobile robot moving objects to specific places in an office domain.

2 Related Work

Knowledge representation, reasoning, and learning are well-researched areas in robotics and AI. Logic-based representations and probabilistic graphical models have been used to control sensing, navigation and interaction for robots and agents [Bai *et al.*, 2014; Galindo *et al.*, 2008]. Formulations based on probabilistic representations (by themselves) make it difficult to perform commonsense reasoning, whereas classical planning algorithms and logic programming tend to require considerable prior knowledge of the domain and the agent's capabilities. For instance, theories of reasoning about action and change, and the non-monotonic logical reasoning

ability of ASP have been used by an international community for controlling the behavior of one or more robots [Balduccini *et al.*, 2014; Saribatur *et al.*, 2014]. However, ASP does not support probabilistic representation of uncertainty, whereas a lot of information extracted from sensors and actuators is represented probabilistically.

Researchers have designed architectures that combine deterministic and probabilistic algorithms for task and motion planning [Kaelbling and Lozano-Perez, 2013], or combine a probabilistic extension of ASP with POMDPs for human-robot dialog [Zhang and Stone, 2015]. Recent work used a three-layered organization of knowledge and reasoning, and combined first-order logic with probabilistic reasoning for open world planning [Hanheide *et al.*, 2015]. Some popular formulations that combine logical and probabilistic reasoning include Markov logic network [Richardson and Domingos, 2006], Bayesian logic [Milch *et al.*, 2006], and probabilistic extensions to ASP [Baral *et al.*, 2009; Lee and Wang, 2015]. However, algorithms based on first-order logic do not provide the desired expressiveness—they do not support non-monotonic logical reasoning, and it is not always possible to express degrees of belief quantitatively, e.g., by adding probabilities to logic statements. Algorithms based on logic programming do not support all the desired capabilities such as incremental revision of (probabilistic) information, and reasoning with a large probabilistic component.

Many tasks that require an agent to learn from repeated interactions with the environment have been posed as Reinforcement Learning (RL) problems [Sutton and Barto, 1998]. As a step towards addressing fundamental challenges such as scaling and transfer of learned knowledge, relational RL (RRL) combines relational representations with regression for generalization [Dzeroski *et al.*, 2001]. Existing approaches, however, use RRL for planning, and generalization is limited to a single MDP for a specific planning task [Driessens and Ramon, 2003; Gartner *et al.*, 2003; Tadepalli *et al.*, 2004].

As a step towards addressing the challenges described above, we have developed architectures that couple declarative programming, probabilistic graphical models, and reinforcement learning [Sridharan *et al.*, 2016; 2015; Zhang *et al.*, 2015]. Here, we describe the architecture, and illustrate its capabilities in simulation and in the context of a physical robot assisting in an office domain.

3 Architecture Description

Figure 1 shows the components of our architecture. We illustrate the components using the following examples.

Office Domain: Consider a robot that is assigned the goal of moving specific objects to specific places in an office. This domain contains:

- Sorts such as *place*, *thing*, *robot*, and *object*, with *object* and *robot* being subsorts of *thing*. Sorts *textbook*, *printer* and *kitchenware*, are subsorts of *object*. Also, sorts for object attributes *color*, *shape*, and *size*.
- Four specific places: *office*, *main_library*, *aux_library*, and *kitchen*. We assume that these places are accessible from each other without the need to navigate any corridors, and that doors between these places are open.
- An instance of sort *robot*, called rob_1. Also, a number of instances of subsorts of the sort *object* in specific places.

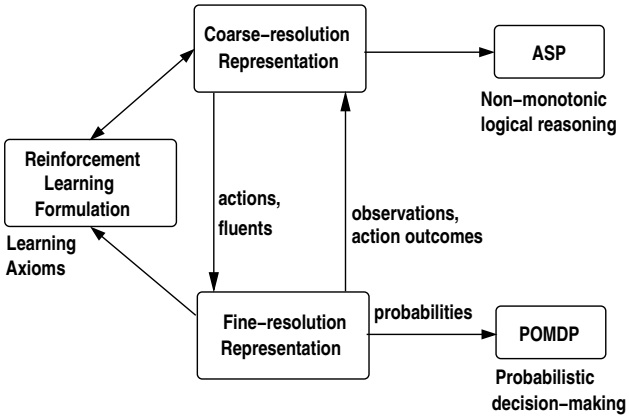


Figure 1: Architecture integrates the complementary strengths of declarative programming, probabilistic graphical models, and reinforcement learning, for knowledge representation, reasoning, and learning, with qualitative and quantitative descriptions of knowledge and uncertainty.

In this domain, coarse-resolution reasoning considers the location of objects in places, while fine-resolution reasoning considers the location of objects in grid cells in these places. As an additional example used to illustrate the relational reinforcement learning, consider:

Blocks World (BW): a tabletop domain where the robot's objective is to stack blocks characterized by different colors, shapes, and sizes, in specific configurations on a table. The sorts of the BW domain include elements such as *block*, *place*, *color*, *shape*, *size*, and *robot*. A scenario with four blocks of the same size corresponds to ≈ 70 states under a standard RL/MDP formulation [Dzeroski *et al.*, 2001]. In this domain, the robot may not know, for instance, that no block can be placed on a prism-shaped block.

Action Language: The transition diagrams of our architecture's coarse-resolution and fine-resolution domain representations are described in an *action language* AL [Gelfond and Kahl, 2014]. AL has a sorted signature containing three *sorts*: *statics* (domain properties whose truth values cannot be changed by actions), *fluents* (domain properties whose values can be changed by actions) and *actions* (elementary actions that can be executed in parallel). AL allows three types of statements: causal laws, state constraints and executability conditions.

3.1 Coarse-Resolution Planning and Diagnosis

The coarse-resolution domain representation has a system description $\mathscr{D}_H$ and histories with defaults $\mathscr{H}$. $\mathscr{D}_H$ consists of a sorted signature (Σ_H) that defines the names of objects, functions, and predicates available for use, and axioms to describe the coarse-resolution transition diagram τ_H. Examples of sorts in the example domain are

place, *thing*, and *robot*. The fluents and actions are defined in terms of their arguments, e.g., in our domain, $loc(thing, place)$ and $in\_hand(robot, object)$ are some inertial fluents[1], and $move(robot, place)$, $grasp(robot, object)$, $putdown(robot, object)$, and $put(object, object)$ are some actions. Examples of axioms include causal laws such as:

$$move(R, Pl) \textbf{ causes } loc(R, Pl)$$
$$grasp(R, Ob) \textbf{ causes } in\_hand(R, Ob)$$

state constraints such as:

$$\neg loc(Ob, Pl_1) \textbf{ if } loc(R, Pl_2),\ Pl_1 \neq Pl_2$$
$$loc(Ob, Pl) \textbf{ if } loc(R, Pl),\ in\_hand(R, Ob)$$

and executability conditions such as:

$$\textbf{impossible } move(R, Pl) \textbf{ if } loc(R, Pl)$$
$$\textbf{impossible } grasp(R, Ob) \textbf{ if } loc(R, Pl_1),\ loc(Ob, Pl_2),\ Pl_1 \neq Pl_2$$
$$\textbf{impossible } grasp(R, Ob) \textbf{ if } in\_hand(R, Ob)$$

The recorded history of a dynamic domain is usually a record of (a) fluents observed to be true at a time step $obs(fluent, boolean, step)$, and (b) the occurrence of an action at a time step $hpd(action, step)$. Our architecture *expands on this view by allowing histories to contain (prioritized) defaults describing the values of fluents in their initial states*. For instance, the default "textbooks are typically in the main library. If a textbook is not there, it is in the auxiliary library. If the textbook is not there either, it is in the office" can be represented elegantly as:

$$\textbf{initial default } loc(X, main\_library) \textbf{ if } textbook(X)$$
$$\textbf{initial default } loc(X, aux\_library) \textbf{ if } textbook(X),\ \neg loc(X, main\_library)$$
$$\textbf{initial default } loc(X, office) \textbf{ if } textbook(X),\ \neg loc(X, main\_library),\ \neg loc(X, aux\_library)$$

This coarse-resolution domain representation is transformed into a program $\Pi(\mathscr{D}_H, \mathscr{H})$ in CR-Prolog that incorporates consistency restoring (CR) rules in ASP [Gelfond and Kahl, 2014]. ASP is based on stable model semantics and non-monotonic logics, and includes *default negation* and *epistemic disjunction*, e.g., unlike $\neg a$ that states *a is believed to be false*, *not a* only implies that *a is not believed to be true*, and unlike "$p \vee \neg p$" in propositional logic, "*p or* $\neg p$" is not a tautology. ASP can represent recursive definitions, defaults, causal relations, and constructs that are difficult to express in classical logic formalisms. The ground literals in an *answer set* obtained by solving Π represent beliefs of an agent associated with Π; statements that hold in all such answer sets are program consequences. Algorithms for computing the entailment of CR-Prolog programs, and for planning and diagnostics, reduce these tasks to computing answer sets of CR-Prolog programs. Π consists of causal laws of $\mathscr{D}_H$, inertia axioms, closed world assumption for defined fluents, reality checks, and records of observations, actions, and defaults, from $\mathscr{H}$. Every default is turned into an ASP rule and a CR rule that allows the robot to assume, under exceptional circumstances, that the default's conclusion is false, so as to restore program consistency—see [Sridharan *et al.*, 2015; Zhang *et al.*, 2014] for formal definitions of states, entailment, and models for consistent inference.

In addition to planning, the architecture supports reasoning about exogenous actions to explain the unexpected (observed) outcomes of actions [Balduccini and Gelfond, 2003]. For instance, to reason about a door between two rooms being locked unexpectedly (e.g., by a human), we introduce exogenous action $locked(door)$ and add the axioms:

$$is\_open(D) \leftarrow open(R, D),\ \neg ab(D)$$
$$ab(D) \leftarrow locked(D)$$

where a door is considered *abnormal*, i.e., $ab(D)$, if it has been locked, say by a human. Actions and suitable axioms are included for other situations in a similar manner. We also introduce an *explanation generation* rule and a new relation *expl* as follows:

$$occurs(A, I) \mid \neg\, occurs(A, I) \leftarrow exogenous\_action(A),\ I < n$$
$$expl(A, I) \leftarrow action(exogenous, A),\ occurs(A, I),\ not\ hpd(A, I)$$

where *expl* holds if an exogenous action is hypothesized but there is no matching record in the history. We also include *awareness* axioms and *reality check* axioms:

$$\%\ \text{awareness axiom}$$
$$holds(F, 0)\ or\ \neg\, holds(F, 0) \leftarrow fluent(basic, F)$$
$$occurs(A, I) \leftarrow hpd(A, I)$$
$$\%\ \text{reality checks}$$
$$\leftarrow obs(fluent, true, I),\ \neg\, holds(fluent, I)$$
$$\leftarrow obs(fluent, false, I),\ holds(fluent, I)$$

The awareness axioms guarantee that an inertial fluent's value is always known, and that reasoning takes into account actions that actually happened. The reality check axioms cause a contradiction when observations do not match expectations, and the explanation for such unexpected symptoms can be reduced to finding (and extracting suitable statements from) the answer set of the corresponding program [Gelfond and Kahl, 2014]. The new knowledge is included in the ASP program and used for subsequent inference. This approach provides *all* explanations of an unexpected symptom. The other option is to use a CR rule instead of the explanation generation rule:

$$occurs(A, I) \stackrel{+}{\leftarrow} exogenous\_action(A),\ I < n$$

where the robot assumes the occurrence of an exogenous action, under exceptional circumstances, to restore consistency.

[1] Inertial fluents obey the laws of inertia and can be changed directly by actions, while defined fluents are not subject to inertia axioms and cannot be changed directly by an action.

the set of CR rules with the smallest cardinality is considered to be the *minimal* explanation. The architecture also includes a similar approach (with CR rules) to reason about partial scene descriptions, e.g., properties of objects and events, extracted from sensor inputs such as camera images. Given ideal descriptions of domain objects, and partial descriptions extracted from sensor input, candidate explanations are sets of CR rules that can be triggered to explain the descriptions, the set with lowest cardinality is the minimal explanation—see [Colaco and Sridharan, 2015] for more details.

3.2 Fine-Resolution Probabilistic Planning

For any given goal, the answer set obtained by ASP-based coarse-resolution inference includes a sequence of abstract actions. Each such action a^H in state σ of τ_H is executed by probabilistic reasoning at a fine-resolution. This reasoning includes three steps:

1. Define the fine-resolution version of the coarse-resolution transition diagram and randomize it.
2. Zoom to the part of the randomized fine-resolution transition diagram that is relevant to the execution of a^H.
3. Construct a POMDP from the zoomed part, solve it to obtain a policy, and use policy to execute a sequence of concrete actions.

The fine-resolution system description $\mathscr{D}_L$ has a sorted signature Σ_L and axioms that describe transition diagram τ_L. Unlike the coarse-resolution representation, the fine-resolution representation implicitly includes a history of observations and actions—the current state is assumed to be the result of all information obtained in previous time steps. Σ_L inherits the sorts, fluents, actions, and axioms from the coarse resolution signature and introduces new ones (or revised versions) that are viewed as components of their coarse-resolution counterparts. For instance, sorts *room* and *cell* are subsorts of *place*, while new fluent $loc(thing, cell)$ represents the cell location of things in the domain. Since action execution is considered to be non-deterministic in the fine-resolution representation, we introduce new fluents to keep track of observations, e.g., $observed(fluent, value, outcome)$, with $outcomes = \{true, false, undet\}$, keeps track of the observed values of specific fluents. New actions are also introduced, e.g., $test(robot, fluent, value)$ is used to test a fluent for a specific value. In addition, we define new statics to describe relations between the new sorts, and new axioms that describe the relations between the coarse-resolution elements and their fine-resolution counterparts. We specify a sequence of steps that defines the fine-resolution transition diagram as a *refinement* of the coarse-resolution diagram such that, for every state transition $T = \langle \sigma, a^H, \sigma' \rangle$ in τ_H, there is a path in τ_L from state s compatible with σ, to some state compatible with σ'—see [Sridharan and Gelfond, 2016] for details.

The certainty of the robot's observations and the effects of the actions executed are only known with some degree of probability. We model this uncertainty by *randomizing* $\mathscr{D}_L$, i.e., by replacing the deterministic causal laws in $\mathscr{D}_L$ by non-deterministic ones and modifying the signature to declare each affected fluent as a random fluent. The randomized system description $\mathscr{D}_{LR}$ is used in semi-supervised experimental trials to collect statistics and compute the probabilities of action outcomes and reliability of observations. Reasoning probabilistically over $\mathscr{D}_{LR}$ can result in incorrect behavior and be computationally intractable for complex domains. To execute any given abstract action a^H in state σ of τ_H, the architecture thus *zooms* to $\mathscr{D}_{LR}(T)$, the part of $\mathscr{D}_{LR}$ that is relevant to the execution of a^H—see [Sridharan and Gelfond, 2016] for details.

Next, $\mathscr{D}_{LR}(T)$ is used to construct a POMDP defined by the tuple $\langle S^L, A^L, Z^L, T^L, O^L, R^L \rangle$ for a specific goal state. The first three elements are the set of states, actions, and the values of observable fluents. The next two elements are the transition function $T^L : S^L \times A^L \times S'^L \rightarrow [0,1]$, which defines the probabilistic state transitions, and the observation function $O^L : S^L \times A^L \times Z^L \rightarrow [0,1]$, which defines the probability of observing the values of observable fluents by executing knowledge producing actions in specific states—these actions do not change the physical state. Functions T^L and O^L describe a probabilistic transition diagram over the belief state using the statistics collected experimentally. The reward specification $R^L : S^L \times A^L \times S'^L \rightarrow \Re$ is used to encode the relative cost or *utility* of taking specific actions in specific states, based on the goal state that is to be achieved. Since the true state is partially observable, planning computes a *policy* $\pi : b_t \rightarrow a_{t+1}$ that maximizes the cumulative reward over a planning horizon to map belief states, i.e., probability distributions over the states, to actions. The POMDP tuple is constructed using data structures that allow the use of existing (approximate) POMDP solvers. Plan execution uses the policy to repeatedly choose an action in the current belief state, and updates the belief state after executing that action and receiving an observation:

$$b_{t+1}(s_{t+1}) \propto O(s_{t+1}, a_{t+1}, o_{t+1}) \sum_s T(s, a_{t+1}, s_{t+1}) \cdot b_t(s)$$

Policy execution is terminated by a transition to a terminal state. In our case, this transition occurs because the probability of one of the states is very high, or because none of the states are very likely and there is no value in executing the policy further—the latter case is interpreted as the inability to execute a^H. The corresponding action outcomes are added as statements to the history in the coarse-resolution description—see [Sridharan *et al.*, 2015] for details about constructing and solving the POMDP.

Constructing and solving a POMDP can become computationally inefficient for complex domains, e.g., rooms with many cells connected to many other rooms, even with sophisticated POMDP solvers. To address this problem, we have explored reasoning in ASP at a finer resolution (e.g., areas in places instead of places), with selective grounding of the variables [Colaco and Sridharan, 2015]. Our prior work has also explored hierarchical decompositions of POMDPs for reliable and efficient operation [Sridharan *et al.*, 2010].

3.3 Reinforcement Learning

Consider the task of stacking books in the *main_library* in our illustrative domain, and assume that the axiom "larger books cannot be stacked on smaller books" is not known to the robot. Generating and executing plans that do not take this

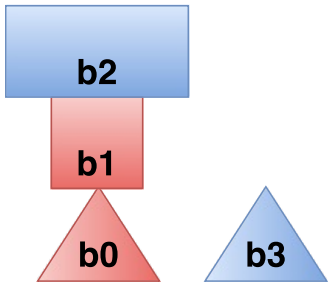


Figure 2: Illustrative example of a planned goal state that the robot cannot achieve.

axiom into account will result in the robot failing to accomplish the desired objective of stacking the books. Similarly, in the BW domain, a robot that does not know the axiom "no block can be placed on a prism-shaped block", and asked to stack any three of the four blocks placed on a table, may attempt to reach the goal configuration shown in Figure 2. The first action in the corresponding plan: $move(b_1, b_0)$ followed by $move(b_2, b_1)$, will result in a failure that cannot be explained. Our architecture supports incremental discovery of unknown axioms governing domain dynamics, by integrating relational reinforcement learning (RRL). The current beliefs of the robot, and the system descriptions at the two resolutions, are used to formulate the task of incrementally discovering domain axioms as an RL problem. Ideally, the state should be estimated using the fine-resolution representation and the corresponding (POMDP) belief states. However, to explore the feasibility of RRL for discovering axioms, our current RRL implementation abstracts away the uncertainty in perception and consider the corresponding MDP. Furthermore, we currently focus on discovering executability conditions for actions in the coarse-resolution description, i.e., axioms that encode the conditions under which each specific action cannot be executed.

Axioms in the coarse-resolution ASP program eliminate impossible state transitions in the RL formulation. In the RL formulation, the goal (achieving which provides high rewards) is set to be the state that resulted in the unexplained plan step failure. Over repeated episodes of Q-learning, the relative values of different state-action pairs (i.e., the Q-values) are computed. Once the Q-values converge, this approach can identify specific ground actions that should not be attempted. However, these axioms may conflict with existing axioms, or include specific instances of more general axioms. Conflicts can be identified as inconsistencies in the answer set of the corresponding ASP program. To discover the general axioms, we first support generalization within the MDP, using state-action pairs visited in a set of episodes to construct a binary decision tree—each path from the root to a leaf corresponds to a state-action pair, and individual nodes are specific fluents. This tree is used to provide a policy for the subsequent episode(s). When Q-learning is terminated, this tree relationally represents the robot's experiences. The second step simulates similar errors (to the one that triggered RRL) and considers the corresponding MDPs. The Q-value of a state-action pair is now the the weighted average of the values across different MDPs—the weight is inversely proportional to the distance to the goal state based on the optimal policy for the MDP. These similar MDPs may be chosen using the information encoded in the ASP program. The third step identifies candidate axioms by constructing training samples based on specific actions and the corresponding related fluents encoded in the binary decision tree. These training samples are used to construct a decision tree in which each path from the root node to a leaf is a candidate executability condition. The final step considers all candidate axioms for different actions, and uses K-means algorithm to cluster these candidates based on their value. The axioms that fall within the cluster with the largest mean are considered to represent generalized axioms, and are added to the ASP program to be used in the subsequent steps—see [Sridharan *et al.*, 2016] for details about our RRL approach.

4 Summary of Experimental Results

This section summarizes some experimental results in simulation and on physical robots to demonstrate the capabilities of the architecture—for more information, please see [Colaco and Sridharan, 2015; Sridharan *et al.*, 2015; Sridharan and Gelfond, 2016; Sridharan *et al.*, 2016]. The simulator uses models that represent objects using probabilistic functions of features extracted from images, and models that reflect the robot's motion and perception capabilities.

First, consider an execution scenario in which the robot is in the $office$, and it is assigned the goal of moving a specific textbook tbk to the $office$. Based on default knowledge (about the location of textbooks), the robot creates a plan of abstract actions:

$$move(rob_1, main\_library),\ grasp(rob_1, tbk)$$
$$move(rob_1, office),\ putdown(rob_1, tbk)$$

where the robot rob_1 will have to search for tbk in the $main\_library$ before grasping it. Each action is executed probabilistically by constructing and solving the corresponding POMDP, as described above.

Next, consider the comparison of the proposed architecture (henceforth "PA") with just using POMDPs ("POMDP-1") in simulation trials. In these trials, the objective of the robot was to move specific objects (with unknown locations) to specific places in the domain. Note that POMDP-1 includes a hierarchical decomposition to make the task of solving the POMDPs computationally tractable [Zhang *et al.*, 2013]. The POMDP solver is given a fixed amount of time to compute action policies. An object's location in a cell is assumed to be known with certainty if the probabilistic belief (of the object's existence in the cell) exceeds a threshold (0.85). The robot's ability to successfully complete the task is shown in Figure 3 as a function of the number of cells in the domain; each data point is the average of 1000 trials, and each room has four cells. As the number of cells increases, it becomes computationally difficult to generate good POMDP action policies that, in conjunction with incorrect observations, significantly impacts the ability to complete the trials. PA focuses the

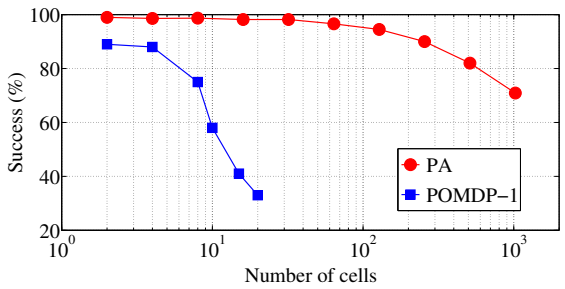


Figure 3: With limited policy computation time, PA is much more accurate than POMDPs as the number of cells increases.

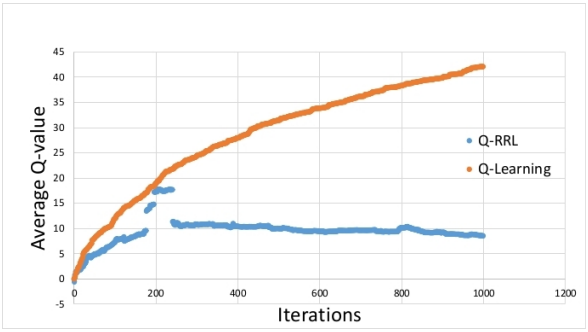


Figure 4: Comparing the rate of convergence of Q-RRL with that of Q-learning—Q-RRL converges much faster.

robot's attention on relevant rooms and cells to improve computational efficiency while still maintaining high accuracy—for larger domains, there is a drop in accuracy but the impact is much less pronounced.

The time taken by PA to generate a plan was also computed as a function of the domain size (characterized as the number of rooms and objects). PA generates appropriate plans for domains with a large number of rooms and objects. Using only the knowledge relevant to the goal significantly reduces the planning time in comparison with using all the domain knowledge available. This relevant part of the domain knowledge can be identified using the relations encoded in the coarse-resolution description. We also compared PA with POMDP-1 on a wheeled robot deployed on multiple floors of an office building. POMDP-1 takes 1.64 as much time as PA to move specific objects to specific places; this 39% reduction in execution time is statistically significant. Furthermore, we instantiated and evaluated our architecture in a different domain, e.g., of a robot waiter assisting in seating people and delivering orders in a restaurant. Results indicated that a purely probabilistic approach takes twice as much time as PA to locate and move objects to specific places. Videos of experimental trials can be viewed online: `http://youtu.be/8zL4R8te6wg`, `https://vimeo.com/136990534`

Next, to evaluate the robot's ability to discover previously unknown executability conditions, we designed multiple simulated trials in which the robot had to arrange objects in specific configurations. Some axioms were hidden from the robot, resulting in failure when certain intermediate configurations were reached. Rewards were provided by the simulator based on the success or failure of the plan. The robot successfully identified actions that could not be executed, and added suitable axioms to the coarse-resolution system description. For instance, in the BW domain, Figure 4 shows the rate of convergence of the average Q-values obtained using Q-RRL (i.e., our approach for relational reinforcement learning) is much better than that of Q-learning. It does not matter whether the actual average Q-values of Q-Learning are higher or lower than those of Q-RRL. In the BW domain, the robot also successfully discovered that no object should be stacked on a prism-shaped object:

$$\textbf{impossible}\ \ move(A,D)\ \ \textbf{if}\ \ has\_shape(D,prism)$$

In a similar fashion, in the context of stacking books in the office domain, the robot discovered that bigger books should not be stacked on smaller ones:

$$\textbf{impossible}\ put(B_1,B_2)\ \ \textbf{if}\ \ bigger(B_1,B_2),\ textbook(B_1),\ textbook(B_2).$$

Including such axioms in the ASP program enables the robot to generate plans that can be successfully executed to achieve the desired goal state, e.g., stacking of blocks or books in a desired configuration. For additional experimental results of evaluating our RRL approach, see [Sridharan *et al.*, 2016].

5 Conclusions

This paper described an architecture that combines the complementary strengths of declarative programming, probabilistic graphical models, and relational reinforcement learning (RRL). Tentative plans created by reasoning with commonsense knowledge in the coarse-resolution are implemented in the fine-resolution using probabilistic algorithms, adding statements to the coarse-resolution history. The incremental and interactive discovery of previously unknown domain axioms is formulated as an RRL problem. Experimental results indicate that the architecture supports reasoning and learning at the sensorimotor level and the cognitive level, and scales well to complex domains. These capabilities are important for robots collaborating with humans. Future work on the architecture will explore tighter coupling of the logical and probabilistic reasoning, and extensive trials on robots collaborating with humans in different domains.

Acknowledgments

The architecture summarized in this paper is based on collaboration with Michael Gelfond, Jeremy Wyatt, Shiqi Zhang, Zenon Colaco, Rashmica Gupta, and Prashanth Devarakonda. This work was supported in part by the US Office of Naval Research Science of Autonomy award N00014-13-1-0766. All opinions and conclusions described in this paper are those of the author.

References

[Bai *et al.*, 2014] Haoyu Bai, David Hsu, and Wee Sun Lee. Integrated Perception and Planning in the Continuous Space: A

POMDP Approach. *International Journal of Robotics Research*, 33(8), 2014.

[Balduccini and Gelfond, 2003] Marcello Balduccini and Michael Gelfond. Diagnostic Reasoning with A-Prolog. *Theory and Practice of Logic Programming*, 3(4-5):425–461, 2003.

[Balduccini *et al.*, 2014] Marcello Balduccini, William C. Regli, and Duc N. Nguyen. An ASP-Based Architecture for Autonomous UAVs in Dynamic Environments: Progress Report. In *International Workshop on Non-Monotonic Reasoning (NMR)*, Vienna, Austria, July 17-19, 2014.

[Baral *et al.*, 2009] Chitta Baral, Michael Gelfond, and Nelson Rushton. Probabilistic Reasoning with Answer Sets. *Theory and Practice of Logic Programming*, 9(1):57–144, January 2009.

[Colaco and Sridharan, 2015] Zenon Colaco and Mohan Sridharan. What Happened and Why? A Mixed Architecture for Planning and Explanation Generation in Robotics. In *Australasian Conference on Robotics and Automation (ACRA)*, Canberra, Australia, December 2-4, 2015.

[Driessens and Ramon, 2003] Kurt Driessens and Jan Ramon. Relational Instance-Based Regression for Relational Reinforcement Learning. In *International Conference on Machine Learning (ICML)*, pages 123–130. AAAI Press, 2003.

[Dzeroski *et al.*, 2001] Saso Dzeroski, Luc De Raedt, and Kurt Driessens. Relational Reinforcement Learning. *Machine Learning*, 43:7–52, 2001.

[Galindo *et al.*, 2008] Cipriano Galindo, Juan-Antonio Fernandez-Madrigal, Javier Gonzalez, and Alessandro Saffioti. Robot Task Planning using Semantic Maps. *Robotics and Autonomous Systems*, 56(11):955–966, 2008.

[Gartner *et al.*, 2003] Thomas Gartner, Kurt Driessens, and Jan Ramon. Graph Kernels and Gaussian Processes for Relational Reinforcement Learning. In *International Conference on Inductive Logic Programming (ILP)*, pages 140–163. Springer, 2003.

[Gelfond and Kahl, 2014] Michael Gelfond and Yulia Kahl. *Knowledge Representation, Reasoning and the Design of Intelligent Agents*. Cambridge University Press, 2014.

[Hanheide *et al.*, 2015] Marc Hanheide, Moritz Gobelbecker, Graham Horn, Andrzej Pronobis, Kristoffer Sjoo, Patric Jensfelt, Charles Gretton, Richard Dearden, Miroslav Janicek, Hendrik Zender, Geert-Jan Kruijff, Nick Hawes, and Jeremy Wyatt. Robot Task Planning and Explanation in Open and Uncertain Worlds. *Artificial Intelligence*, 2015.

[Kaelbling and Lozano-Perez, 2013] Leslie Kaelbling and Tomas Lozano-Perez. Integrated Task and Motion Planning in Belief Space. *International Journal of Robotics Research*, 32(9-10):1194–1227, 2013.

[Lee and Wang, 2015] Joohyung Lee and Yi Wang. A Probabilistic Extension of the Stable Model Semantics. In *AAAI Spring Symposium on Logical Formalizations of Commonsense Reasoning*, March 2015.

[Milch *et al.*, 2006] Brian Milch, Bhaskara Marthi, Stuart Russell, David Sontag, Daniel L. Ong, and Andrey Kolobov. BLOG: Probabilistic Models with Unknown Objects. In *Statistical Relational Learning*. MIT Press, 2006.

[Richardson and Domingos, 2006] Matthew Richardson and Pedro Domingos. Markov Logic Networks. *Machine Learning*, 62(1-2):107–136, February 2006.

[Saribatur *et al.*, 2014] Zeynep Saribatur, Esra Erdem, and Volkan Patoglu. Cognitive Factories with Multiple Teams of Heterogeneous Robots: Hybrid Reasoning for Optimal Feasible Global Plans. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014.

[Sridharan and Gelfond, 2016] Mohan Sridharan and Michael Gelfond. Using Knowledge Representation and Reasoning Tools in the Design of Robots. In *IJCAI Workshop on Knowledge-based Techniques for Problem Solving and Reasoning (KnowProS)*, New York, USA, July 10, 2016.

[Sridharan *et al.*, 2010] Mohan Sridharan, Jeremy Wyatt, and Richard Dearden. Planning to See: A Hierarchical Approach to Planning Visual Actions on a Robot using POMDPs. *Artificial Intelligence*, 174:704–725, 2010.

[Sridharan *et al.*, 2015] Mohan Sridharan, Michael Gelfond, Shiqi Zhang, and Jeremy Wyatt. A Refinement-Based Architecture for Knowledge Representation and Reasoning in Robotics. Technical report, Unrefereed CoRR abstract: `http://arxiv.org/abs/1508.03891`, August 2015.

[Sridharan *et al.*, 2016] Mohan Sridharan, Prashanth Devarakonda, and Rashmica Gupta. Discovering Domain Axioms Using Relational Reinforcement Learning and Declarative Programming. In *ICAPS Workshop on Planning and Robotics (PlanRob)*, London, UK, June 13-14, 2016.

[Sutton and Barto, 1998] R. L. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, USA, 1998.

[Tadepalli *et al.*, 2004] Prasad Tadepalli, Robert Givan, and Kurt Driessens. Relational Reinforcement Learning: An Overview. In *Relational Reinforcement Learning Workshop at the International Conference on Machine Learning*, 2004.

[Zhang and Stone, 2015] Shiqi Zhang and Peter Stone. CORPP: Commonsense Reasoning and Probabilistic Planning, as Applied to Dialog with a Mobile Robot. In *AAAI Conference on Artificial Intelligence*, pages 1394–1400, Austin, USA, 2015.

[Zhang *et al.*, 2013] Shiqi Zhang, Mohan Sridharan, and Christian Washington. Active Visual Planning for Mobile Robot Teams using Hierarchical POMDPs. *IEEE Transactions on Robotics*, 29(4):975–985, 2013.

[Zhang *et al.*, 2014] Shiqi Zhang, Mohan Sridharan, Michael Gelfond, and Jeremy Wyatt. Towards An Architecture for Knowledge Representation and Reasoning in Robotics. In *International Conference on Social Robotics (ICSR)*, pages 400–410, Sydney, Australia, October 27-29, 2014.

[Zhang *et al.*, 2015] Shiqi Zhang, Mohan Sridharan, and Jeremy Wyatt. Mixed Logical Inference and Probabilistic Planning for Robots in Unreliable Worlds. *IEEE Transactions on Robotics*, 31(3):699–713, 2015.