Extending Analogical Generalization with Near-Misses

Matthew D. McLure
Qualitative Reasoning Group
Northwestern University
2133 Sheridan Road
Evanston, IL, 60208, USA
mclure@u.northwestern.edu

Scott E. Friedman
Smart Information Flow Technologies (SIFT)
Minneapolis, MN, USA
friedman@sift.net

Kenneth D. Forbus
Qualitative Reasoning Group
Northwestern University
2133 Sheridan Road
Evanston, IL, 60208, USA
forbus@northwestern.edu

Abstract

Concept learning is a central problem for cognitive systems. Generalization techniques can help organize examples by their commonalities, but comparisons with non-examples, *near-misses,* can provide discrimination. Early work on near-misses required hand-selected examples by a teacher who understood the learner's internal representations. This paper introduces Analogical Learning by Integrating Generalization and Near-misses (ALIGN) and describes three key advances. First, domain-general cognitive models of analogical processes are used to handle a wider range of examples. Second, ALIGN's analogical generalization process constructs multiple probabilistic representations per concept via clustering, and hence can learn disjunctive concepts. Finally, ALIGN uses unsupervised analogical retrieval to find its own near-miss examples. We show that ALIGN out-performs analogical generalization on two perceptual data sets: (1) hand-drawn sketches; and (2) geospatial concepts from strategy-game maps.

Introduction

Learning concepts from examples is a core capability for cognitive systems. While many approaches learn over feature vectors, this work involves learning from more expressive relational representations, similar to inductive logic programming (Muggleton and De Raedt 1994; Cleuziou, Martin, and Vrain 2003). We focus here on similarity-based supervised learning, where labeled examples are available. Winston (1970) introduced the idea of a near-miss, a negative example that is very similar to a positive example. The small number of differences – ideally one – simplified the learner's search for necessary conditions for category membership. Winston's system used analogical matching to compare the structured representations of the positive and negative examples in order to find the difference(s). Winston's system had some important limitations: it used a domain-specific analogical matcher; it required the teacher to know the internal representations of the concept; it required the teacher to label near-misses; it learned one representation per concept and thus could not handle disjunctive concepts; and it was only tested on blocks-world scenes.

Since that time, the state of the art in knowledge representation and analogical reasoning has improved to the point where a more general model capable of using near-misses can be formulated over more general representations. This paper describes Analogical Learning by Integrating Generalization and Near-misses (ALIGN), which is built on Gentner's (1983) structure-mapping theory of analogy and similarity. Computational models of human analogical matching, retrieval, and generalization (summarized below) are combined and extended to yield three important advantages: (1) ALIGN learns both characteristic properties (i.e., descriptions generalized over positive examples) and discriminative properties (i.e., strict membership criteria) of categories; (2) ALIGN can learn disjunctive categories; and (3) ALIGN automatically identifies near-misses via analogical retrieval, so the teacher does not need to provide them or know the underlying knowledge representations.

We provide empirical results to support these claims, using ALIGN to learn spatial concepts from hand-sketched examples in a sketch understanding system. We use sketched materials for two reasons. First, visual and spatial information is of great importance in domains such as science, engineering, and art, and typically involves rich, relational representations. Second, encoding spatial representations automatically from raw ink reduces tailorability, as opposed to hand-generated representations.

We start by summarizing the analogical processing models that ALIGN uses, and the sketch understanding system we used to automatically encode examples for the experiments. We then describe ALIGN and two experimental analyses: (1) ALIGN learns concepts from sketches of everyday objects; and (2) ALIGN learns geospatial concepts from circled examples on a strategy-game map. We close with related and future work.

Background

Computational models of Structure-Mapping

Structure-Mapping is a psychological theory of analogy and similarity that has been used to explain how people compare visual stimuli (Markman and Gentner 1996; Sagi, Gentner, and Lovett 2012), learn abstract categories (Gentner and Namy 1999), and learn contrastive categories via difference detection (Smith and Gentner 2014). It defines analogical comparison as a process of aligning two structured representations, a base and a target, guided by mapping constraints. The matching process produces *correspondences* that specify which elements (i.e., entities and expressions) in the base go with which elements in the target. Structure-mapping also generates *candidate inferences* that project unshared structure (i.e., non-corresponding expressions) from base to target or vice-versa.

The Structure-Mapping Engine (SME)

The Structure-Mapping Engine (Falkenhainer, Forbus, and Gentner 1989) is a computational model of structure-mapping theory. SME operates via a local-to-global process, initially constructing local match hypotheses in parallel, followed by a serial phase that greedily coalesces islands of matches into globally consistent mappings, guided by a scoring process based on structure-mapping principles. Each mapping contains a set of correspondences and a *similarity score* that captures the overall similarity. The similarity score is normalized to the closed interval [0, 1] by dividing by the mean of the scores of the self-matches of base and target. Each mapping can contain candidate inferences, from base to target, and reverse candidate inferences, from target to base. Candidate inferences can include *analogy skolems*, which are entities hypothesized via projection because they lack a correspondent in the other case.

MAC/FAC

MAC/FAC (Forbus, Gentner, and Law 1995) is a model of similarity-based retrieval built on SME. The inputs are (1) a *probe* case and (2) a set of cases from which to retrieve, called a *case library*. MAC/FAC retrieves up to three similar cases from the case library based on similarity to the probe. The algorithm has two stages. The first stage (MAC) computes in parallel coarse similarity estimates between the probe and every case in the case library using content vectors, a redundant non-structured representation computed from the cases. Each dimension in a content vector is proportional to the number of statements using that predicate in the original case, thus the dot product estimates the number of local matches that SME will find between the probe and each case. This scales well to large case libraries. The cases with the highest dot product with the probe are passed to the second stage, FAC. FAC uses SME to compare the probe to each MAC result, in parallel. The case with highest SME similarity score, plus up to two more if very close to the top scorer, are returned as the reminding(s).

Sequential Analogical Generalization Engine (SAGE)

The Sequential Analogical Generalization Engine (SAGE) is a model of analogical generalization built on SME, and the successor to SEQL (Kuehne et al. 2000). SAGE maintains a *generalization context* for each concept, where incoming training examples are incrementally clustered to form *generalizations* (two or more analogically mapped examples) and *unassimilated examples* (clusters of size one). SAGE generalizations are themselves cases, but each expression in a generalization is assigned a probability based on the frequency with which it corresponded to expressions in co-clustered examples.

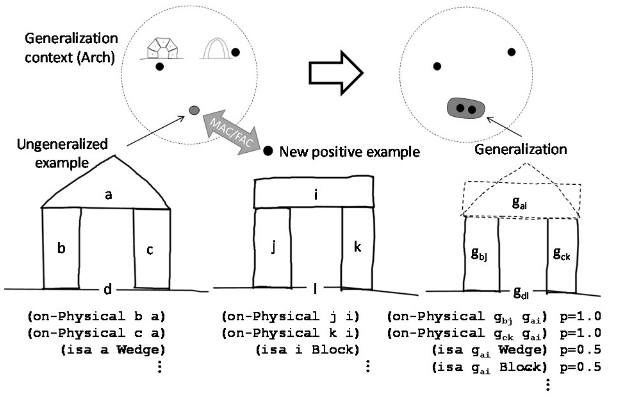


Figure 1: An example is added to the "Arch" generalization context and generalized with a previously unassimilated example.

The SAGE algorithm consists of two stages: (1) *select* at most one existing cluster that is sufficiently similar to the incoming example, and (2) *merge* the incoming example into the cluster, if sufficiently similar. The select process uses MAC/FAC, with the new example as the probe and the generalization context as the case library. If the normalized similarity score of the best reminding is greater than the *assimilation threshold* of the generalization context, SAGE merges the example; otherwise, the new example is added as an unassimilated example.

The merge process uses the correspondences in the SME mapping to (1) update the probabilities associated with each existing expression in the generalization, and (2) add new expressions from the example to the generalization. When non-identical entities correspond in the mapping, SAGE replaces them with *generalized entities* (i.e., symbols that represent multiple symbols from examples). A merge is illustrated in Figure 1, where corresponding expressions in the resulting generalization have p = 1.0, and non-corresponding expressions have a p = 0.5. SME uses these probabilities to bias the local scores on match

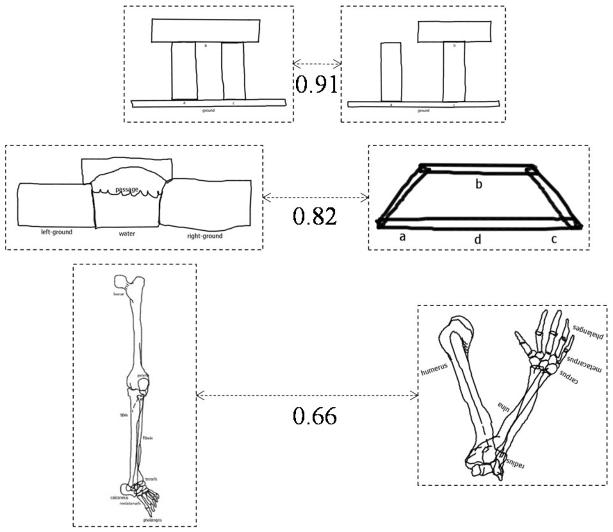


Figure 2: Three pairs of sketched objects with decreasing normalized similarity scores, from top to bottom. All three pairs are potential near-misses because their labels differ.

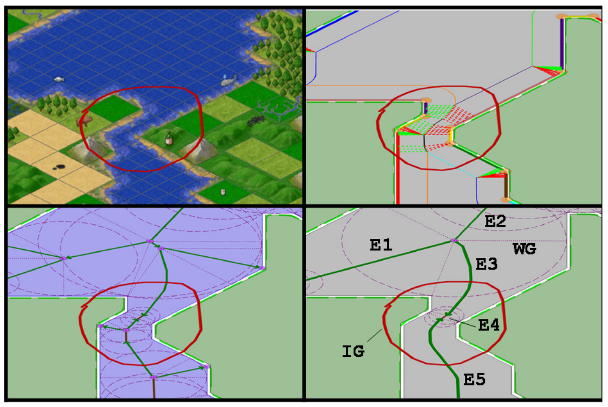
hypotheses. As SAGE assimilates more examples into a generalization, some expressions will retain high probability, while others will diminish. Consequently, generalizations in SAGE express a probability distribution over their expressions, separating characteristic (high probability) structure from incidental (low-probability) structure. SAGE uses a *probability threshold* to prune low-probability expressions from the generalization and maintain tractability.

SAGE has been used to learn and classify musical genres (Dehghani and Lovett 2006), spatial prepositions (Lockwood, Lovett, and Forbus 2008), and sketched every-day objects (Lovett, Dehghani, and Forbus 2007; McLure, Friedman, and Forbus 2012), suggesting that it provides a domain-general foundation for ALIGN.

CogSketch

Sketching is a powerful way to express visual and spatial ideas. CogSketch (Forbus et al. 2011) is an open-domain sketch understanding system. Users draw digital ink, which they segment into visual entities (glyphs) and label with concepts drawn from a large (OpenCyc-derived) knowledge base.[1] Conceptual labeling is a practical vehicle for information that is often conveyed with language during human-to-human sketching (e.g., "this is a block"). CogSketch constructs visual representations from what is drawn (e.g., that two glyphs have intersecting edges) and spatial representations from both the visual and conceptual information (e.g., that the objects depicted by the glyphs are touching). Figure 2 shows six examples. CogSketch's extensive vocabulary of representations is motivated by human spatial cognition, including relations

[1] www.cyc.com/platform/opencyc



```
(isa SG SelectionGlyph)     (isa E4 StraightEdge)
(isa WG WaterGlyph)         (narrowRadius E4)
(bisects WG SG)             (lengthShort E4)
(medialAxisEdge WG E4)      (clockwiseNeighborAtJct E1 E2)
(contains SG E4)            (sourceJct (elementsConnected E1 E2))
(intersectsInk SG E3)       (obtuseJct (elementsConnected E1 E2))
(isa E4 SinkEdge)           (straightJct (directedConnection E3 E4))
            ⋮
```

Figure 3: A water glyph (top-left) is decomposed into its skeleton edges (bottom right) by using a grassfire algorithm (top-right) to compute a medial-axis transform, which is pruned and segmented according to its radius function.

for adjacency, relative position and size, and topological relationships. It is capable of analyzing the same sketch at multiple levels of description (i.e., edges, objects, and groups).

CogSketch interacts with the strategy game Freeciv to encode qualitative structured representations for regions on the map. CogSketch renders the map and creates glyphs for units, cities, and *terrain blobs* (i.e., polygons that outline groups of tiles that share some property). The user selects a region by drawing a glyph on the map (e.g., the red circle in Figure 3). When the user assigns a conceptual label to the selection glyph, CogSketch automatically encodes a case. The encoding scheme used in our geospatial classification task (illustrated in Figure 3) begins by detecting and recording topological relationships between the selection glyph and overlapping land or water terrain blobs. Any blob glyph that overlaps the selection glyph is decomposed into a skeleton based on the Medial Axis Transform (MAT). Each skeleton is further segmented at points corresponding to qualitative changes in the radius function (i.e., the distance from each point to its closest points on the exterior) – a strategy inspired by shock graphs (Siddiqi et al. 1999). The result for each blob glyph is a network of edges directed from its wider sections to its narrower sections. The scheme encodes qualitative properties over these edges including length, radius function, curvature, and various aspects of connectivity. Finally, topological relationships between each edge and the selection glyph are encoded.

CogSketch has been used to model visual problem solving (Lovett, Forbus, and Usher 2010), including cultural differences (Lovett and Forbus 2011). It has also been used as a platform for sketch-based educational software (Yin et al. 2010). These experiments suggest that the representations it produces are both realistic and useful.

ALIGN

Similarity can be deceptive. Isthmuses are similar to straits, bays are similar to peninsulas, and bridges are similar to arches. In this section, we describe how ALIGN leverages these deceptive similarities as opportunities to learn and represent category boundaries. We begin by discussing how ALIGN detects and exploits deceptive similarities, then we describe how it represents and revises its hypotheses, and finally we describe how it classifies examples with a mix of similarity and hypothesis-testing.

Detecting & Exploiting Near-Misses with Analogy

Given a labeled training example (e.g., of an *arch*), ALIGN uses MAC/FAC to retrieve similar training examples with *different* labels (e.g., *bridge*) or a *null* label.[2] High-similarity pairs of examples with different labels above a *similarity threshold* (equal to SAGE's assimilation threshold) are *near-misses*. Using the best SME mapping between the positive (*arch*) and negative (*bridge* or *null*) example, ALIGN uses the candidate inferences (i.e., projected expressions across cases) to produce hypotheses about category boundaries.

ALIGN distinguishes between two types of candidate inferences (CIs) that produce different hypotheses:

1. *Positive-to-negative CIs* (PNCIs) project relations from the positive example in terms of the negative example's entities. ALIGN converts these to *inclusion hypotheses*: necessary criteria for asserting category membership.

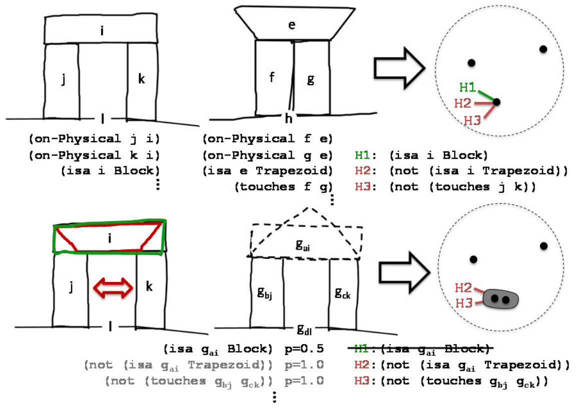

i
j
k
l
e
f
g
h
H1
H2
H3
(on-Physical j i)
(on-Physical k i)
(isa i Block)
(on-Physical f e)
(on-Physical g e)
(isa e Trapezoid)
(touches f g)
H1: (isa i Block)
H2: (not (isa i Trapezoid))
H3: (not (touches j k))
i
j
k
l
g_{ai}
g_{bj}
g_{ck}
g_{dl}
H2
H3
(isa g_{ai} Block) p=0.5
(not (isa g_{ai} Trapezoid)) p=1.0
(not (touches g_{bj} g_{ck})) p=1.0
~~H1: (isa g_{ai} Block)~~
H2: (not (isa g_{ai} Trapezoid))
H3: (not (touches g_{bj} g_{ck}))

Figure 4: A near-miss pair and hypothesis revision

2. *Negative-to-positive CIs* (NPCIs) project relations from the negative example, in terms of the positive example's entities. ALIGN converts these to *exclusion hypotheses*: sufficient criteria for blocking category membership.

ALIGN associates hypotheses with the positive example, and represents hypotheses in terms of the positive example's entities. For PNCIs, this requires a translation step that replaces the entities mentioned (from the negative side) with their corresponding (positive) entities. In Figure 4 (top), H1 is an inclusion hypothesis for the category *arch* resulting from the PNCI `(isa e Block)`, and H3 is an exclusion hypothesis resulting from the NPCI `(touches f g)`.

ALIGN also represents skolems in its hypotheses. A skolem in a PNCI is replaced with the positive entity from which it was projected, e.g. `(AnalogySkolemFn m)` translates to `m`. In contrast, NPCIs with skolems result in more complex exclusion hypotheses, because they need to

```
CL = {} ;; a case library for training examples
;; S = similarity threshold, $M_{YX}$ = reverse mapping of $M_{XY}$
TrainOn(example P, label $L_P$) ;; label may be null
  if $L_P$
    $GC_P$ = Label→GeneralizationContext ($L_P$)
    {$G_P$, $M_{GP}$} = SageAdd (P, $GC_P$, S) ;; if unassimilated,
                                            ;; $G_P$=P & $M_{GP}$=null
  for each {N, $M_{NP}$} in MacFacRetrieve(P,CL)
    $L_N$ = Example→Label(N)
    {$G_N$, $M_{GN}$} = Example→Generalization (N)
    if $L_P \neq L_N$ and NormScore($M_{NP}$) > S ;; a near-miss
      GenerateNewCriteria($L_P$, $G_P$, P, N, $M_{NP}$, $M_{PG}$)
      GenerateNewCriteria($L_N$, $G_N$, N, P, $M_{PN}$, $M_{NG}$)
  CL = CL + P

GenerateNewCriteria(label L, gen G, pos P, neg N,
                    mapping $M_{NP}$, mapping $M_{PG}$)
  if L and not BlockedByCriteria(N, G, $M_{GP}$, $M_{PN}$)
    $H_{PL}$ = ExtractCriteria(P, L, $M_{NP}$)
    AddCriteria(G, GeneralizeCriteria($H_{PL}$, $M_{PG}$))

TestOn(unlabeled example U)
  D = {} ;; discarded remindings
  for 1 to 3
    R = MacFacRetrieve(U, (CL – D))
    for each {E, $M_{EU}$} in SortByGenSimilarity(R, U)
      $L_E$ = Example→Label(E)
      {$G_E$, $M_{GE}$} = Example→Generalization(E)
      if BlockedByCriteria(U, $G_E$, $M_{GE}$, $M_{EU}$)
        D = D + E
      else return $L_E$
  return Example→Label(LeastBlocked(D))
```

Figure 5: ALIGN's top-level training and testing procedures.

[2] In this paper, categories are mutually exclusive, and some training and testing examples have a null label (i.e., they are confusers).

specify criteria about *extra* entities that aren't represented in the positive case while remaining embedded in the case. For these NPCIs, ALIGN replaces each skolem with a variable, and constrains each variable so that it cannot bind to any entity from the positive example that entered into the near-miss mapping. Thus, a NPCI such as `(touches j (AnalogySkolemFn x))` is translated to capture the hypothesis that *there cannot be something extra that touches j,* as follows (*cannot* is implicit because it is an exclusion hypothesis):

```
(and (different i ?skolem)
     (different j ?skolem)
     (different k ?skolem)
     (different l ?skolem)
     (touches j ?skolem))
```

ALIGN discards near-miss pairs that are *explained away* by existing hypotheses associated with the positive example.

In practice, hypotheses produced from a single comparison can be over-specific, and must be revised to be practically useful. We describe this process next.

Revising Hypotheses with Analogical Generalization

ALIGN maintains one SAGE generalization context for every (non-*null*) label encountered, and each training example is added to the generalization context corresponding to its label, if any. If SAGE assimilates the example into a generalization, its associated hypotheses are generalized to refer to the entities in the generalization. Generalized hypotheses are pruned if they include conditions not true of all examples assimilated into the generalization (i.e., expressions with $p < 1.0$ for inclusion hypotheses, and $p > 0$ for exclusion hypotheses). Figure 4 (bottom) demonstrates pruning an inclusion hypothesis.

ALIGN uses generalized hypotheses to block labels and explain away new near-misses. An extra analogical translation step is required to project these hypotheses from the generalization to an unlabeled or negative example, via the positive (assimilated) example. The degree to which a failed hypothesis blocks a label is weighted using the size of the generalization it is associated with, with the intuition being that hypotheses that persist after more generalization operations are more accurate, all else being equal.

Since SAGE can produce multiple clusters from the positive examples of a single concept, ALIGN can maintain alternative sets of hypotheses for a given concept, resulting in a disjunctive category structure. Structural similarity (via analogical retrieval) determines which disjunct(s) get tested against a given unlabeled example.

Classification via Analogy

To classify an unlabeled example, ALIGN retrieves similar labeled examples from memory. Those retrieved examples that have been assimilated during learning are exchanged for their associated generalizations. Starting with the most similar case, ALIGN tests each of the case's associated hypotheses by:

1. Translating the hypothesis to refer to entities in the unlabeled example, using the entity correspondences of the best mapping.
2. Testing whether the translated condition holds. If an exclusion hypothesis holds or an inclusion hypothesis fails to hold (skolems count as failures), do not infer that label.

If these hypotheses are consistent with the new example, ALIGN classifies the new example with the retrieved label. Otherwise, ALIGN iterates to the next retrieved example. This is important for distinguishing similar categories.

Because we assume that the set of concepts is collectively exhaustive, ALIGN conducts repeated retrievals during classification until it finds an unblocked label. If no unblocked label is found after 3 retrievals, then the label of the most highly supported retrieval is chosen. Support is determined by subtracting the cumulative weight of the inconsistent hypotheses from that of the consistent hypotheses. Since *null*-labeled retrievals cannot have associated hypotheses, they have neutral (0) support.

Evaluation

To evaluate the extension of analogical generalization with near-misses, we ran two classification tasks under different conditions: (1) *ALIGN* is the full system described above; (2) *Prototypes* is *ALIGN* without near-miss analysis; and (3) *Examples* is *ALIGN* without near-misses and *also* without analogical generalization, so classification is reduced to similarity-based retrieval over the library of training examples. For moderately high similarity thresholds (0.6-0.9), we expected Prototypes to outperform Examples and ALIGN to outperform both.

Arches: 8	Bridges: 4	*False arches:* 8
Skeletal arms: 4	Triangles: 4	*False triangles:* 4
Skeletal legs: 4	Quadrilaterals: 4	*False quads:* 4

Table 1: A dataset of sketched concepts. "False" categories were negative examples of all categories.

Experiment 1

ALIGN was compared with the Prototypes and Examples conditions in a classification task involving objects sketched in CogSketch. There were 44 sketches distributed over 6 mutually exclusive labels, where 16 of the sketches had *null* labels (i.e., no positive labels) but were potential near-misses for some of the concepts (e.g., *false arches* are nearly arches) as summarized in Table 1. The cases produced by CogSketch each contained, on average, 4.5

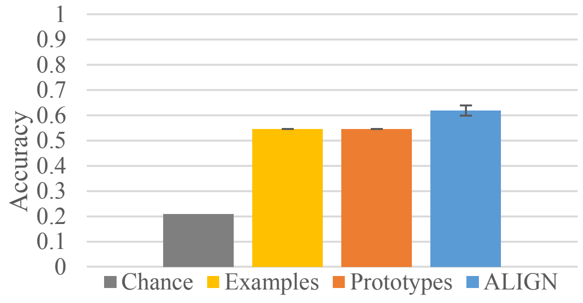


Figure 6: Classification performance of ALIGN compared to Examples and Prototypes.

unique entities and 31 facts. The task was to choose one or none of the 6 labels, with chance at 21% accuracy. Performance was evaluated via 4-fold cross-validation. Statistical significance was measured using a one-tailed, paired t-test. The similarity threshold for ALIGN was set to 0.8. This was also the assimilation threshold used for the Prototypes condition.

Results

Figure 6 shows the results of Experiment 1. The Prototypes and Examples approaches performed identically at 55% accuracy, well above chance ($p < 0.01$). ALIGN demonstrated a 7% improvement over these conditions ($p < 0.01$).

Experiment 2

This experiment evaluated classification performance in a task involving geospatial concepts. Examples were circled on a Freeciv map using CogSketch as described above. The dataset consisted of 60 examples evenly distributed over 6 mutually exclusive and collectively exhaustive categories: *Isthmus*, *Peninsula*, *Strait*, *Bay*, *Archipelago* and *Island*. The cases produced by CogSketch each contained, on average, 8 unique entities and 60 facts. 10-fold cross-validation was used. ALIGN was compared to the Prototypes approach over a range of similarity thresholds. At a threshold of 1, both reduce to Examples.

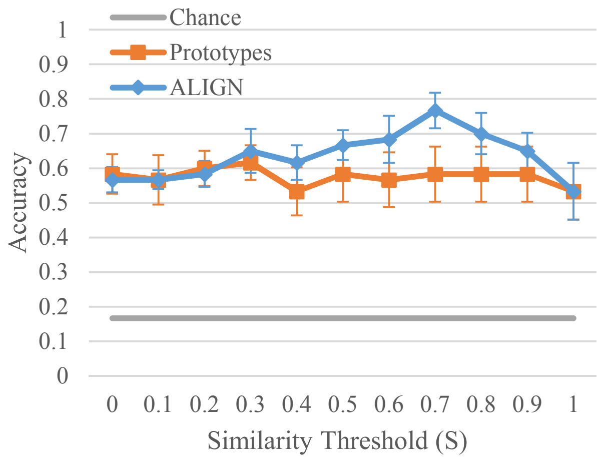


Figure 7: Classification accuracy on a geospatial dataset.

Results

As shown in Figure 7, ALIGN achieved a peak accuracy of 77%, significantly outperforming the Prototypes peak performance of 62% ($p < 0.05$). The Prototypes condition where similarity threshold (S) is 1 is by definition the Examples condition, since $S = 1$ effectively prevents generalization and near-miss analysis. Here, Examples performed significantly worse than ALIGN with an accuracy of 53% ($p < 0.05$), which was not significantly different from the peak of Prototypes.

Discussion

This data demonstrates that extending analogical generalization with near-misses can improve classification for spatial concepts represented as structured, qualitative descriptions. The interaction between the similarity threshold and the marginal benefit of near-misses suggests that there is a "sweet spot" for comparison; when the threshold is too low, more distant near-miss comparisons are generated, resulting in more hypotheses, most of which are incidental differences and should be pruned. When the similarity threshold is too high, the generalizations in SAGE end up smaller in size and more numerous, providing less basis for pruning bad hypotheses.

Using analogical generalization alone (Prototypes) did not significantly improve performance over simple retrieval (Examples) as we expected. The identical performance in Experiment 1 is likely related to the size of the categories; out of a group of four bridges, at least three must be sufficiently similar for two to generalize in the training set and a third to match with the resulting prototype during classification. Also recall that Experiment 1 contains examples without any positive label, which are not given a chance to generalize. Thus, their incidental similarities to other categories are not suppressed.

Related Work

ALIGN is capable of learning with relatively few examples. It requires orders of magnitude fewer examples than existing connectionist models (e.g., Elman 1998; Krizhevsky, Sutskever, and Hinton 2012). We believe ALIGN makes more realistic demands.

The IDeAL system (Bhatta and Goel 1997) learned by analogy over structured representations similar to those used by ALIGN, but it was applied to a design problem-solving task instead of recognition. Its retrieval mechanism indexed previous designs by their function, whereas retrieval in ALIGN uses domain-general similarity-based retrieval. IDeAL learned generic teleological mechanisms by comparing a correct solution provided by a teacher to designs retrieved from memory – similar to near-miss discovery in ALIGN – but it relied on design-specific background knowledge to abstract the appropriate aspects from the design pair. Hypothesis production in ALIGN is domain-general, so instead it relies on finding similar positive examples to properly refine its hypotheses.

Cleuziou, Martin and Vrain (2003) described an inductive logic programming (ILP) approach to learning disjunctive concepts that bore similarities to ALIGN. Positive examples were clustered into *subconcepts* according to a similarity measure, and a separate conjunctive definition was learned for each subconcept based on the negative examples. Its representations, similarity metric, and hypothesis generation were all based in first order logic; the rules learned contained logical variables and were tested deductively, in contrast with the hypotheses here, which are embedded in learned models and tested via analogical projection. Winston (1982, 1986) also explored learning rules from analogies, producing logical rules via generalizing on one example. His if-then rules and censors were functionally similar to our inclusion and exclusion hypotheses, respectively.

Another analogous approach is K-Means+ID3 (Gaddam, Phoha, and Balagani 2007), which cascaded k-means clustering and ID3 decision trees for anomaly detection. While their approach used feature-vector data, it shared the idea that rule-based class boundaries can apply better when trained locally in some similarity space.

Conclusions and Future Work

This paper describes ALIGN, a concept learning model that extends analogical generalization with near-misses. It is capable of automatically finding its own near-miss examples, and we have shown evidence from two experiments that near-misses significantly improve performance over analogical generalization and retrieval alone. We plan on experimenting with ALIGN in other domains, as well as expanding the range of concepts in sketched domains.

Acknowledgements

This material is based upon work supported by the Air Force Office of Scientific Research under Award No. FA2386-10-1-4128.

References

Bhatta, S., & Goel, A., 1997. Learning generic mechanisms for innovative strategies in adaptive design. *The Journal of the Learning Sciences* 6(4), 367-396.

Cleuziou, G., Martin, L., & Vrain, C., 2003. Disjunctive learning with a soft-clustering method. In *ILP'03: 13th International Conference on Inductive Logic Programming*, 75–92.

Elman, J., 1998. Generalization, simple recurrent networks, and the emergence of structure. In *Proceedings of the Twentieth Annual Conference of the Cognitive Science Society*. Mahwah, NJ: Lawrence Erlbaum Associates.

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

Forbus, K., Gentner, D., and Law, K., 1995. MAC/FAC: A model of similarity-based retrieval. *Cognitive Science* 19, 141-205.

Forbus, K. D., Usher, J., Lovett, A., Lockwood, K. and Wetzel, J., 2011. Cogsketch: Sketch understanding for cognitive science research and for education. *Topics in Cognitive Science* 3.

Gaddam, S., Phoha, V., & Balagani, K., 2007. K-Means+ID3: A novel method for supervised anomaly detection by cascading k-means clustering and ID3 decision tree learning methods. *IEEE Transactions on Knowledge and Data Engineering* 19(3).

Gentner, D., & Namy, L., 1999. Comparison in the development of categories. *Cognitive Development* 14, 487-513.

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

Krizhevsky, A., Sutskever, I., and Hinton, G. E., 2012. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 1097-1105.

Kuehne, S., Forbus, K., Gentner, D. and Quinn, B., 2000. SEQL: Category learning as progressive abstraction using structure mapping. *Proceedings of CogSci. 2000.*

Lovett, A., Forbus, K., and Usher, J., 2010. A structure-mapping model of Raven's Progressive Matrices. *Proc. of CogSci 2010.*

Lovett, A., & Forbus, K., 2011. Cultural commonalities and differences in spatial problem solving: A computational analysis. *Cognition* 121, 281-287.

Markman, A. B., & Gentner, D., 1996. Commonalities and differences in similarity comparisons. *Memory & Cognition* 24(2), 235-249.

Muggleton, S., & De Raedt, L., 1994. Inductive logic programming: Theory and methods. *Journal of Logic Programming* 19-20, 629-679.

Sagi, E., Gentner, D., & Lovett, A., 2012. What difference reveals about similarity. *Cognitive Science* 36, 1019-1050.

Siddiqi, K., Shokoufandeh, A., Dickinson, S. J., and Zucker, S. W., 1999. Shock graphs and shape matching. *International Journal of Computer Vision* 30, 1-24.

Smith, L., & Gentner, D., 2014. The role of difference-detection in learning contrastive categories. *Proceedings of CogSci. 2014.*

Winston, P. H., 1970. Learning structural descriptions from examples. Ph.D. diss., MIT, Cambridge, MA.

Winston, P. H., 1982. Learning new principles from precedents and exercises. *Artificial Intelligence* 23(12).

Winston, P. H., 1986. Learning by augmenting rules and accumulating censors. In *Machine Learning: An Artificial Intelligence Approach* 2, 45-62. Morgan-Kaufman.

Yin, P., Forbus, K., Usher, J., Sageman, B. and Jee, B., 2010. Sketch Worksheets: A Sketch-based Educational Software System. *Proceedings of the 22nd Annual Conference on Innovative Applications of Artificial Intelligence*.