Scaling up Analogy with Crowdsourcing and Machine Learning

Joel Chan[1], Tom Hope[2], Dafna Shahaf[2] and Aniket Kittur[1]

[1] Human-Computer Interaction Institute
Carnegie Mellon University, Pittsburgh PA 15213, USA
joelchuc@cs.cmu.edu, nkittur@cs.cmu.edu,
[2] School of Computer Science and Engineering
Hebrew University of Jerusalem, Jerusalem, Israel
tom.hope@mail.huji.ac.il, dshahaf@cs.huji.ac.il

Abstract. Despite tremendous advances in computational models of human analogy, a persistent challenge has been scaling up to find useful analogies in large, messy, real-world data. The availability of large idea repositories (e.g., the U.S. patent database) could significantly accelerate innovation and discovery in a way never previously possible. Previous approaches have been limited by relying on hand-created databases that have high relational structure but are very sparse (e.g., predicate calculus representations). Traditional machine-learning/information-retrieval similarity metrics (e.g., LSA) can scale to large, natural-language datasets; however, while these methods are good at detecting surface similarity, they struggle to account for structural similarity. In this paper, we propose to leverage crowdsourcing techniques to construct a dataset with rich "analogy-tuning" signals, used to guide machine learning models towards matches based on relations rather than surface features. We demonstrate our approach with a crowdsourced analogy identification task, whose results are used to train deep learning algorithms. Our initial results suggest that a deep learning model trained on positive/negative example analogies from the task can find more analogous matches than an LSA baseline, and that incorporating behavioral signals (such as queries used to retrieve an analogy) can further boost its performance.

Keywords: Analogy, crowdsourcing, machine learning

1 Introduction

Invention by analogy (i.e., transferring ideas from other domains that are structurally similar to a target problem) is a powerful way to create new innovations. For example, a car mechanic invented a new low-cost way to ease difficult childbirth by drawing an analogy to a cork extraction method in wineries (inserting and inflating a small plastic bag in the bottle) [12]. This award-winning device has the potential to change lives worldwide, particularly women in developing countries with limited medical resources.

The recent growth of online innovation repositories represents an unparalleled opportunity for invention by analogy. These repositories contain hundreds of thousands (Quirky, OpenIDEO) or millions (the U.S. patent database, the Web) of ideas that have the potential to be applied to other structurally similar domains. However, the scale of these repositories presents a challenge to a person's ability to find useful analogies.

Computational systems could greatly accelerate innovation by mining analogies from these vast repositories. Indeed, decades of research on computational models of human analogy-making have yielded tremendous advances in the ability of computational systems to explain and simulate human-like analogical reasoning. Yet, a persistent challenge has been **scaling up computational analogy systems to reliably find useful analogies in large, messy, real-world data**. Existing approaches are limited by either relying on hand-created databases that have high relational structure but are small, domain-specific, and costly to keep updated [14, 17], or on machine learning approaches that can scale to large datasets but have difficulty encoding and matching relations [5, 16].

In this paper, we propose a hybrid approach which combines crowdsourcing with machine learning to develop a scalable approach to finding analogies in large idea repositories. A key insight is that we aim to externalize and capture the mental processes that humans use to find and evaluate analogies to serve as training data for a machine learning model. The intuition is that instead of trying to build a complete human-generated dataset or a machine learning model driven only from existing data, the rich behavioral traces of how people query for analogies can "tune" a more scalable computational approach towards matches based on relations rather than surface features.

We illustrate our approach through a crowdsourced analogy identification task where people query a repository and find analogical matches to a target. These matches and queries are used as training data (and for feature selection) for deep learning algorithms. Our initial results suggest that a deep learning model trained on positive/negative example analogies from the task can find more analogous matches than an LSA baseline, and that incorporating behavioral signals (such as queries entered) can further boost its performance.

2 Related Work

Computational Analogy Systems. We argue that a crucial reason behind the difficulty in scaling up computational analogy lies in a trade-off between accuracy and scale in existing approaches. On the one hand, models that have been the most successful at approaching human-level performance in analogical matching—such as Hummel and Holyoak's LISA analogy engine [11], Klenk and Forbus's [14] Companion for AP Physics problems, and Vattam and colleagues' [17] Design Analogy to Nature Engine (DANE)—rely heavily on hand-created databases that have high relational structure. Creating such databases involves extensive knowledge engineering efforts. Vattam and colleagues [17] estimate that converting a single (complex) biological system into a formal representation requires between forty and one hundred person-hours of work. Consequently, models that rely on hand-coded relational representations have yet to be successfully applied to large, open repositories like the U.S. Patent Database.

Conversely, a number of machine learning approaches exist that make minimal assumptions about the input data, and in particular do not require explicitly coded relational representations. Examples include word embedding models like Word2Vec [16], vector-space models like Latent Semantic Indexing [5], and probabilistic topic modeling approaches like Latent Dirichlet Allocation [1]. While these approaches scale well to large datasets, they have difficulty encoding *relational* similarity. One possible reason is that these approaches tend to rely on co-occurrence patterns between words that describe higher-level "concepts"; however, relational categories have very sparse and diverse term distributions [8]

Note that some approaches [6] try to capture structure by focusing on particular types of words (e.g., verbs). However, parts-of-speech alone are not enough to capture structural relations, and these methods suffer from a lot of noise. Consequently, these systems tend to have low precision of analogical matches, shifting the burden onto the user to sift through large amounts of false positives. To illustrate, Fu and colleagues [7] found that, despite their approach producing structures that experts found sensible, their "far" analogies actually were perceived as "too far", and hurt instead of helped creative output.

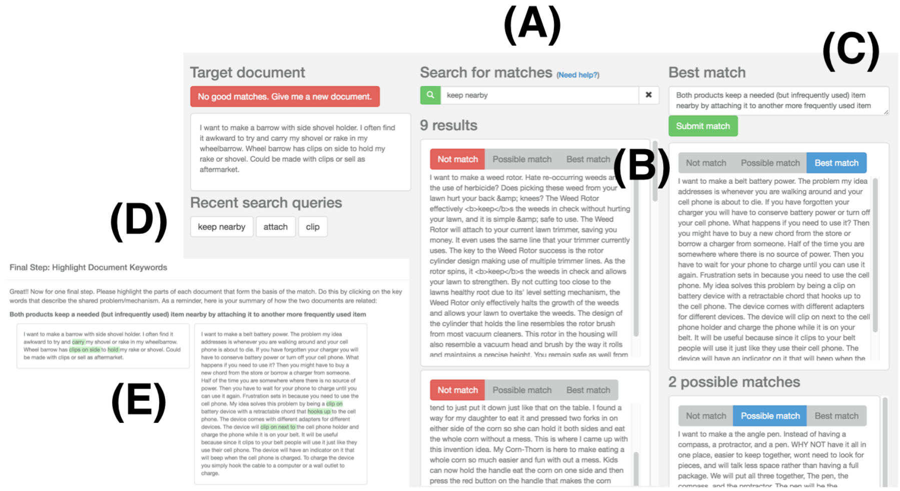
Crowdsourcing and Machine Learning. We believe a possibly fruitful way forward lies in hybrid approaches that combine crowdsourcing and machine learning. We are inspired by related efforts in human-computer interaction that combine these technologies to crowdsource complex cognition at scale, in particular clustering related items from rich and messy text sources [2, 9]. However, these methods are not aimed at finding analogical clusters, which requires supporting deep relational similarity rather than surface similarity.

3 Computational Analogy at Scale with Crowdsourcing and Machine Learning

We frame the problem of finding analogies in a large dataset as a hybrid human and machine-learning problem. We propose to use crowdsourcing to obtain rich "analogy-tuning" signals for machine learning models by capturing the process by which people query for and evaluate analogies. By doing so we aim to collect not just positive/negative examples of analogies, but also implicit and explicit behavioral traces ranging from the queries people use to look for analogies to the keywords they believe are discriminative. We believe these behavioral traces are vital for bridging the gap between scalable machine learning approaches and the structured representations of prior approaches. This machine-learning approach is related to other efforts in case-based-reasoning that use machine-learning to reduce the need for knowledge engineering [10]. To illustrate the potential of this approach, we present a system that seeds a deep learning model with analogy-relevant signals harvested from a crowdsourced analogy querying task.

3.1 Crowdsourcing Task for Collecting Analogy-Tuning Signals

The goal of the crowdsourcing component is to obtain rich behavioral data (e.g., positive/negative examples of analogies, query sequences, keywords) that signal



Target document

I want to make a barrow with side shovel holder. I often find it awkward to try and carry my shovel or rake in my wheelbarrow. Wheel barrow has clips on side to hold my rake or shovel. Could be made with clips or sell as aftermarket.

Recent search queries

keep nearby | attach | clip

Search for matches

keep nearby

9 results

I want to make a weed rotor. Hate re-occurring weeds an the use of herbicide? Does picking these weed from your lawn hurt your back &amp; knees? The Weed Rotor effectively <b>keep</b>s the weeds in check without hurting your lawn, and it is simple &amp; safe to use. The Weed Rotor will attach to your current lawn trimmer, saving you money. It even uses the same line that your trimmer currently uses. The key to the Weed Rotor success is the rotor cylinder design making use of multiple trimmer lines. As the rotor spins, it <b>keep</b>s the weeds in check and allows your lawn to strengthen. By not cutting too close to the lawns healthy root due to its' level setting mechanism, the Weed Rotor only effectively halts the growth of the weeds and allows your lawn to overtake the weeds. The design of the cylinder that holds the line resembles the rotor brush from most vacuum cleaners. This rotor in the housing will also resemble a vacuum head and brush by the way it rolls and maintains a precise height. You remain safe as well from

tend to just put it down just like that on the table. I found a way for my daughter to eat it and pressed two forks in on either side of the corn so she can hold it both sides and eat the whole corn without a mess. This is where I came up with this invention idea. My Corn-Thorn is here to make eating a whole corn so much easier and fun with out a mess. Kids can now hold the handle eat the corn on one side and then press the red button on the handle that makes the corn

Best match

Both products keep a needed (but infrequently used) item nearby by attaching it to another more frequently used item

I want to make a belt battery power. The problem my idea addresses is whenever you are walking around and your cell phone is about to die. If you have forgotten your charger you will have to conserve battery power or turn off your cell phone. What happens if you need to use it? Then you might have to buy a new chord from the store or borrow a charger from someone. Half of the time you are somewhere where there is no source of power. Then you have to wait for your phone to charge until you can use it again. Frustration sets in because you need to use the cell phone. My idea solves this problem by being a clip on battery device with a retractable chord that hooks up to the cell phone. The device comes with different adapters for different devices. The device will clip on next to the cell phone holder and charge the phone while it is on your belt. It will be useful because since it clips to your belt people will use it just like they use their cell phone. The device will have an indicator on it that will beep when the

2 possible matches

I want to make a the angle pen. Instead of having a compass, a protractor, and a pen. WHY NOT have it all in one place, easier to keep together, wont need to look for pieces, and will talk less space rather than having a full package. We will put all three together, The pen, the compass, and the protractor. The pen will be the

Final Step: Highlight Document Keywords

Great!! Now for one final step. Please highlight the parts of each document that form the basis of the match. Do this by clicking on the key words that describe the shared problem/mechanism. As a reminder, here is your summary of how the two documents are related:

Both products keep a needed (but infrequently used) item nearby by attaching it to another more frequently used item

I want to make a barrow with side shovel holder. I often find it awkward to try and carry my shovel or rake in my wheelbarrow. Wheel barrow has clips on side to hold my rake or shovel. Could be made with clips or sell as aftermarket.

I want to make a belt battery power. The problem my idea addresses is whenever you are walking around and your cell phone is about to die. If you have forgotten your charger you will have to conserve battery power or turn off your cell phone. What happens if you need to use it? Then you might have to buy a new chord from the store or borrow a charger from someone. Half of the time you are somewhere where there is no source of power. Then you have to wait for your phone to charge until you can use it again. Frustration sets in because you need to use the cell phone. My idea solves this problem by being a clip on battery device with a retractable chord that hooks up to the cell phone. The device comes with different adapters for different devices. The device will clip on next to the cell phone holder and charge the phone while it is on your belt. It will be useful because since it clips to your belt people will use it just like they use their cell phone. The device will have an indicator on it that will beep when the cell phone is charged. To charge the device you simply hook the cable to a computer or a wall outlet to charge.

Fig. 1. Search workflow. **(A)** Enter queries to **search** for analogous documents, and mark "possible"/"best" matching documents. **(B) Screen** matches (e.g., promote "possible" to "best" match, directly reject "possible"/"best" match), **(C) Describe** analogy to target document. **(D)** View/return to **prior queries** (optional). **(E)** After submitting best match, **highlight** keywords in both documents that explain the analogy.

the core relational structures of documents. Guided by the psychological insight that comparison is a powerful way to get people to attend to the core structural features of a description [15], we decided to embed the task of providing analogy-tuning signals within a realistic task of finding analogies.

Workers use a simple search interface to find product descriptions that are analogous to a seed product. Figure 1 depicts the interface and the four main components of the task: 1) **searching** for matches (A), 2) **screening/processing** matches (B), 3) **describing** the analogy (C), and 4) **highlighting** keywords (E).

This approach yields a rich set of signals that we can use for our machine learning models. For example:

- What **queries** are used, in what **sequence**?
- Which documents are tagged as **possible matches**?
- Which documents are tagged as **best matches**?
- Which documents are **implicitly rejected** (i.e., ignored in the search result list, despite appearing before matches)?
- Which documents are **directly rejected**?
- How is the best match **described** as being analogous to the seed document?
- Which **key terms** are highlighted?

Importantly, this task context enables us to harvest all these rich signals from a natural search task that is easy and familiar. Additionally, the task design could guide the instrumentation of an interface in a user-facing computational analogy system to yield similar signals for ongoing refinement of the underlying models.

Deployment. We test our approach with a corpus of product descriptions from Quirky.com, an online crowdsourced product innovation website. Quirky is representative of the kinds of datasets we are interested in, because it is large (at the time of writing, it hosts upwards of 10,000 product ideas), unstructured (ideas are described in natural language), and covers a variety of domains (invention categories), which makes cross-domain analogies possible. The following example is representative of the length and "messiness" of product ideas in this dataset:

> **Proximity wristband:** *Control of childs in public places is very difficult and stressing. Parents fear that their childs may go too far unexpectedly for any reason, and that actually happens no matter how careful they are, especially in crowdy places. Because childs always run and move. Parents can't relax and are obliged to keep their eyes continuously on their childs. Furthermore, the consequences of a child going too far from his parents may be very dangerous. A wristband is put on the wrist of the child. The wristband has a radio connection (bluetooth) with one of the parents' smartphone, that has been previously matched with the wristband. Parents may activate/disactivate the alarm on the wristband by tapping on the App installed on their smartphone. When the alarm is activated, the App detects the distance between the 2 radio connected devices (the wristband and the smartphone). If the distance gets higher than the maximum value (changeable in the settings of the App) than a speaker integrated in the bracelet emits a loud alarm and the smartphone starts ringing. The inside of the wristband hosts a circular conductive element that loses its metallic continuity if the wristband opens for any reason, so if this circuit is opened the wristband is programmed to emit the alarm and alert the parents.*

We crowdsourced analogy finding within a set of 400 randomly sampled Quirky products. Three hundred and ninety-four workers from Amazon Mechanical Turk collected analogies for 227 "seed" documents (median of 1 unique analogy per seed, range of 1-10 unique analogies). Median completion time for each seed was 10 minutes, and pay was $6/hr (or $1 per completed seed). Workers could complete as many seeds as they wanted.

An example behavioral trace sequence from our data illustrates how the act of comparison pushes people to focus on structure (along with the rich data we can mine from the behavioral traces). Worker X received the "proximity wristband" as a seed. She initially started with the query "alarm", and tagged as a possible match a "voltage plug" product that automatically alerts the user if there are voltage problems for a given power outlet. She also rejected non-analogous results like a "smart doorbell chime". Dissatisfied with the results, she entered a new query ("wristband"), but didn't find any useful matches. Finally, she entered "proximity" as a query, and tagged a product about a "digital dog fence" as a best match, explaining that both products are about "Proximity, keeping object within a set distance, the object has it attached."

3.2 Task 1: Semantic Similarity from Traces

An important challenge of working with natural language is finding appropriate semantic similarities. Many existing similarity metrics, like Word2vec [16] do poorly on verb and adjective similarity, which are central for structural similarity.

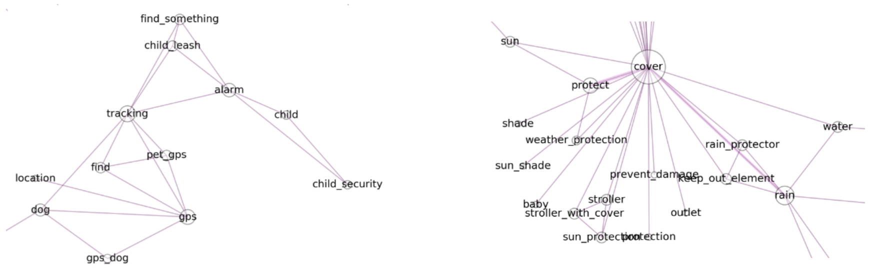


Fig. 2. Fragments of the query graph from our study. Nodes correspond to queries, edges correspond to two queries that were used to find the same analogy.

We now use behavioral traces to expose similarity. We constructed a graph from queries entered in our study: Nodes correspond to queries; there is an edge if the queries resulted in the same analogy. In other words, if one user found an analogy using the query "cover", and another used "protect", we add an edge.

Figure 2 shows fragments from this graph. By construction, the query graph exposes a lot of the desired semantic similarity. Semantically similar verbs often form dense clusters (e.g., protect/defend/shield), and related terms (gps/location/find) also tend to be a short distance away. Traversing the query graph can also reveal analogies: for example, an analogy between a dog gps and a child tracking device (left), and between products that protect from rain, water, and sun (right). This graph could be used to guide feature selection in a machine learning model.

3.3 Task 2: Learning Analogies

We now demonstrate how to frame analogy-finding as a machine learning problem. In this context, we are given a training set $\mathcal{D} = \{(\mathbf{x}_i^1, \mathbf{x}_i^2, y_i)\}$. $(\mathbf{x}_i^1, \mathbf{x}_i^2)$ are pairs of product descriptions. Label $y_i \in \{0, 1\}$ corresponds to whether the pair was tagged as an analogy ("best match") or as a non-analogy (ignored in search result, directly rejected). Our goal is learn a decision function for new pairs whose score reflects their "degree of analogy": $f(\theta, (\psi_\theta(\mathbf{x}_i^1), \psi_\theta(\mathbf{x}_i^2)))$, where $\psi_\theta(\cdot)$ is function embedding $\mathbf{x}_i$ into a shared feature space and θ are model parameters.

The document model we use to demonstrate our ideas is based on convolutional neural network (CNN) architecture that has shown state-of-the-art results in many NLP tasks [13, 4]. This distributional model learns to map texts to vector representations. The objective guides the model to learn a representation where texts tagged as analogous are close, and non-analogous texts are far.

We use a Siamese Network architecture [3], where two identical copies (same weights) of a function are applied to two inputs. Another layer measures distance between the two inputs, computing whether they are similar. Figure 3 shows the main components of the architecture. We represent each word as a low-dimensional, real-valued dense vector $\mathbf{w} \in \mathbb{R}^d$. Each $\mathbf{x}_i$ is thus a sequence of vectors $\mathbf{w}$, which together form a matrix $\mathbf{M}_i$ where column j represents the j^{th} word in the input sequence.

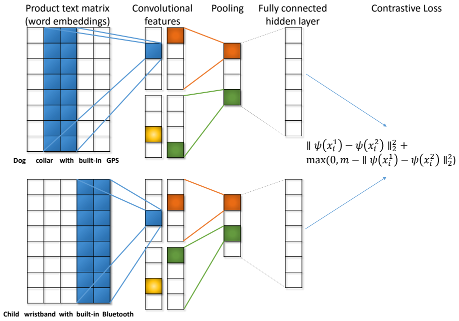

Product text matrix (word embeddings)
Convolutional features
Pooling
Fully connected hidden layer
Contrastive Loss
Dog collar with built-in GPS
Child wristband with built-in Bluetooth
$\| \psi(x_i^1) - \psi(x_i^2) \|_2^2 +$
$\max(0, m - \| \psi(x_i^1) - \psi(x_i^2) \|_2^2)$

Fig. 3. Siamese Network architecture. Words are embedded into a low-dimension representation and combined into a matrix (left). Convolutional "sliding window filters" (blue) are applied to the matrix. Multiple filters are applied (different colors), forming a feature map pooled to form an aggregated signal. The pooled representation goes through a fully connected layer. The same weights are applied for both inputs across all layers. Finally, distance between inputs is computed (Contrastive Loss).

Next, our model learns to compose the **w** into higher-level semantic representations by applying transformations to $\mathbf{M}_i$. Each vector sequence is passed through a **convolutional layer**, applying a bank of "sliding window filters" to extract local features of small subsequences of words.

To learn non-linear patterns, convolutional layers are followed by elementwise **activation functions**. We use the ReLU function, $\max(0, x)$. The output of the activation function is passed through a **pooling layer**, which reduces dimensionality by aggregating information, capturing pertinent patterns and filtering noise. The pooling layer performs max-pooling, returning the largest value for each column. Finally, values pass through a **fully connected** layer which computes a linear transformation followed by ReLU non-linearity, combining local features into a global semantic view of the text.

This composition of functions, from embedding words to the final layer, yields our function $\psi_\theta(\cdot)$, mapping a text input $\mathbf{x}_i$ into a new vector representation. Crucially, in our model $\psi_\theta(\cdot)$ is shared for $(\mathbf{x}_i^1, \mathbf{x}_i^2)$, enabling the model to learn a symmetric representation that is invariant to the order in which the texts are provided. Finally, our objective function is the **Contrastive Loss**, defined as:

$$L(\psi_\theta(\mathbf{x}_i^1), \psi_\theta(\mathbf{x}_i^2)) = y_i L^+(\psi_\theta(\mathbf{x}_i^1), \psi_\theta(\mathbf{x}_i^2)) + (1 - y_i) L^-(\psi_\theta(\mathbf{x}_i^1), \psi_\theta(\mathbf{x}_i^2)),$$

where

$$L^+(\psi_\theta(\mathbf{x}_i^1), \psi_\theta(\mathbf{x}_i^2)) = ||\psi_\theta(\mathbf{x}_i^1) - \psi_\theta(\mathbf{x}_i^2))||_2^2$$
$$L^-(\psi_\theta(\mathbf{x}_i^1), \psi_\theta(\mathbf{x}_i^2)) = \max(0, m - ||\psi_\theta(\mathbf{x}_i^1) - \psi_\theta(\mathbf{x}_i^2))||_2^2.$$

L^+ penalizes positive pairs far apart, and L^- penalizes negative pairs (direct/implicit rejected pairs, from the crowd) closer than margin m. Learning is done with gradient descent, via backpropagation.

Results. We applied the model to the crowd-annotated Quirky data. We split our data into training and evaluation sets, each containing distinct sets of "seed" texts (to test the model's ability to generalize). The training set consists of about 12500 pairs. Positive labels were assigned directly by crowd workers, while negative labels mean the pairs were implicitly rejected by not being tagged despite being viewed by a worker. Our evaluation set comprised of roughly 3000 pairs.

The data was imbalanced, with about 10 negatives for each positive ($\approx 1,100$ positive pairs). To counter the imbalance, we use a weighted loss function. For the LSA baseline, we compute the Singular Value Decomposition (SVD) on the document-term matrix, with term weights given by the TF-IDF score. Cosine similarity is then computed for each pair. The table below shows proportion of analogies among pairs with highest predicted scores (precision@K). For example, looking at the top 2% predictions of our model (2% evaluation-set instances with lowest predicted distance; about 60 pairs), 64% were tagged as positive (vs. 46% for LSA). The overall proportion of positive labels in the test set was only 11%. Importantly, negative labels are derived implicitly, so many pairs with negative annotations could possibly be “mislabeled” and are actually positive.

Method	Top 2%	Top 5%	Top 10%	Top 15%	Top 25%
Siamese Net	0.64	0.54	0.39	0.32	0.28
Latent Semantic Analysis	0.46	0.40	0.34	0.29	0.25

In Table 1 we show examples of seed documents and their best predicted matches. Our model recovers both “purpose ” matches (e.g., gloves that hold nails, and a hammer handle to store tools) as well as “mechanism” (child / dog tracking devices). Comparing our model to LSA-based similarity, one can qualitatively observe that overall, LSA seems to focus more on surface similarity. For example, when starting from a wristband that monitors child proximity, LSA returns a baby recliner with sensors, a lawnmower connected to a wristband, and skateboard shoes; our model returns a pet tracker, a wallet finder and vehicle finder. Our model seems to “err" and return smart window blinds that detect when homeowners are away – possibly recognizing the semantic “analogy” between a child or pet wandering off and a homeowner being away.

3.4 Task 3: Incorporating the query

We now extend our machine learning setting, adding the query as additional input. Our training data is now $\mathcal{D} = \{(\mathbf{x}_i^1, \mathbf{x}_i^2, \mathbf{q}_i, y_i)\}$, where $\mathbf{q}_i$ is the user-entered query. We apply the same representation $\psi_\theta(\cdot)$ we use for $\mathbf{x}_i^1, \mathbf{x}_i^2$ to $\mathbf{q}_i$, embedding the query into a shared feature space. We seek to measure similarity between seed and target in this space. Importantly, similarity should be relative in some sense to the user's query. We do so by using the *projection* of the query vector onto the seed and target vectors – essentially “aligning" the query “concept" with each product text. We then compute the contrastive loss as before. More formally, we re-define the contrastive loss to incorporate the query as follows.

$L^+(\psi_\theta(\mathbf{x}_i^1), \psi_\theta(\mathbf{x}_i^2), \psi_\theta(\mathbf{q}_i)) = ||\frac{\psi_\theta(\mathbf{q}_i)\cdot\psi_\theta(\mathbf{x}_i^1)}{\psi_\theta(\mathbf{x}_i^1)\cdot\psi_\theta(\mathbf{x}_i^1)}\psi_\theta(\mathbf{x}_i^1) - \frac{\psi_\theta(\mathbf{q}_i)\cdot\psi_\theta(\mathbf{x}_i^2)}{\psi_\theta(\mathbf{x}_i^2)\cdot\psi_\theta(\mathbf{x}_i^2)}\psi_\theta(\mathbf{x}_i^2)||_2^2$

and similarly for L^-. In the table below we report results for this new model and compare it to the model without the query. Since many queries did not have matching targets we filter the data to a smaller subset with full information (6000 less rows). The smaller training set drops base model accuracy to .46 (from .64). Interestingly, the query information has an apparent positive effect, compensating for lack of data and boosting accuracy from .46 to .54. These preliminary results validate our intuition that machine learning models would benefit from a variety of human-generated analogy-relevant signals.

Table 1. Best matches examples. Descriptions shortened and separated by slashes.

Seed	Siamese Net top matches	LSA top matches
Proximity wristband for children with a bluetooth connection to parents' smartphone. App detects the distance, emits an alarm if child is too far.	A dog and cat collar that has built-in GPS tracking. // A wallet phone case that can be located with a key-chain remote control. // Programmable smart window blinds that can also detect when you are away. // A vehicle finder with a small portable electronic device transmitting a signal from parked car to smartphone.	Baby recliner - placing sensors on a baby being walked by parent, learn movement and incorporate into reclining chair. // Clamp on a lawnmower safety bar with a pin connected to a wristband. When user is too far pin is pulled out and the clamp opens. // Magnetic skateboard shoes with RF controller for turning on and off. Could be fabricated into a ring or wristband.
Window louvers that stop rain by funneling the rain out.	Programmable window blinds. // A window-mounted fan with sound dampening louvers. // Lamps that work off batteries for additional lighting without depending on electricity	A window-mounted fan with sound dampening louvers. // A glare-stopper device preventing energy-efficient windows from melting sides of buildings or vehicles // A car screen to help reduce excessive heat buildup when parking outside.
A flat magnet sewn into the posterior hand of a glove to hold nails or screws.	A golf glove with a magnetic strip around the wrist. // A hammer with a handle that stores nails and hex screwdrivers. // Tearproof gloves to work easily with tools.	A glove with adhesive contours allowing the user to pick up hair and lint in hard-to-reach places. // A can opener container device, with a magnet on both ends to hold the can in place. // A box that clamps onto a ladder and holds screws, nails and small items.

Method	Top 2%	Top 5%	Top 10%	Top 15%	Top 25%
Model with query	0.54	0.32	0.29	0.25	0.22
Model without query	0.46	0.33	0.23	0.20	0.16

4 Discussion and Conclusions

Overall, we believe this is a promising time to make traction on the problem of finding analogies in complex, messy data. There is a confluence of new work on crowdsourcing complex cognition, machine learning tools for unsupervised learning of semantics, and hybrid approaches combining crowds and machine judgments to get the best of both worlds. We believe there is promise in an approach that positions the process of crowdsourced analogical knowledge base creation as input to machine learning models, using explicit and implicit signals to augment the written text. In this paper we describe a prototype of this approach and early results describing its potential value.

We note that the approach described here does not tackle a number of important problems that we acknowledge as limitations. For example, we do not yet deal with the rich relational structure inherent in the source and target analogs. We also note that our intent is to find interesting and useful analogs in large data repositories, and as such we make no claims that the processes we describe match up with those that human cognition engages in during analogical retrieval and reasoning. However, our capture of the process by which people engage in while searching for analogies, including the queries and keywords they use and their perceived relevance judgments of their resulting matches, may prove valuable for further psychological research on the process of analogical retrieval.

References

1. D. M. Blei, A. Y. Ng, M. I. Jordan, and J. Lafferty. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, pages 993–1022, 2003.
2. J. Bragg and D. S. Weld. Crowdsourcing Multi-Label Classification for Taxonomy Creation. In *HCOMP'13*, 2013.
3. S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *CVPR'05*, volume 1. IEEE, 2005.
4. R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *ICML'08*. ACM, 2008.
5. S. Deerwester, S. T. Dumais, G. W. Furnas, and T. K. Landauer. Indexing by Latent Semantic Analysis. *JASIST*, 41(6):1990, 1990.
6. K. Fu, J. Cagan, K. Kotovsky, and K. L. Wood. Discovering Structure In Design Databases Through Functional And Surface Based Mapping. *JMD*, 135:031006, 2013.
7. K. Fu, J. Chan, J. Cagan, K. Kotovsky, C. Schunn, and K. Wood. The Meaning of Near and Far: The Impact of Structuring Design Databases and the Effect of Distance of Analogy on Design Output. *JMD*, 135(2):021007, 2013.
8. D. Gentner and K. J. Kurtz. Relational Categories. In *Categorization inside and outside the laboratory: Essays in honor of Douglas L. Medin*, APA decade of behavior series. American Psychological Association, Washington, DC, US, 2005.
9. N. Hahn, J. Chang, J. E. Kim, and A. Kittur. The Knowledge Accelerator: Big Picture Thinking in Small Pieces. In *CHI'16*, New York, NY, USA, 2016. ACM.
10. K. Hanney and M. T. Keane. The adaptation knowledge bottleneck: How to ease it by learning from cases. In D. B. Leake and E. Plaza, editors, *Case-Based Reasoning Research and Development*, number 1266 in Lecture Notes in Computer Science, pages 359–370. Springer Berlin Heidelberg, July 1997. DOI: 10.1007/3-540-63233-6_506.
11. J. E. Hummel and K. J. Holyoak. A symbolic-connectionist theory of relational inference and generalization. *Psychological Review*, 110(2):220–264, 2003.
12. D. G. M. Jr. Car Mechanic Dreams Up a Tool to Ease Births. *The New York Times*, Nov. 2013.
13. Y. Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.
14. M. Klenk and K. Forbus. Analogical model formulation for transfer learning in AP Physics. *Artificial Intelligence*, 173(18):1615–1638, Dec. 2009.
15. L. Kotovsky and D. Gentner. Comparison and Categorization in the Development of Relational Similarity. *Child Development*, 67(6):2797–2822, 1996.
16. T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient Estimation of Word Representations in Vector Space. *arXiv:1301.3781 [cs]*, Jan. 2013. arXiv: 1301.3781.
17. S. Vattam, B. Wiltgen, M. Helms, A. K. Goel, and J. Yen. DANE: Fostering Creativity in and through Biologically Inspired Design. In *Design Creativity 2010*. 2011.