FAME: Flexible, Scalable Analogy Mappings Engine

Shahar Jacob, Chen Shani, Dafna Shahaf
The Hebrew University of Jerusalem, Israel
{shahar.jacob, chenxshani, dshahaf}@cs.huji.ac.il

Abstract

Analogy is one of the core capacities of human cognition; when faced with new situations, we often transfer prior experience from other domains. Most work on computational analogy relies heavily on complex, manually crafted input. In this work, we relax the input requirements, requiring only names of entities to be mapped. We automatically extract common-sense representations and use them to identify a mapping between the entities. Unlike previous works, our framework can handle partial analogies and suggest new entities to be added. Moreover, our method's output is easily interpretable, allowing for users to understand why a specific mapping was chosen.

Experiments show that our model correctly maps 81.2% of classical 2x2 analogy problems (guess level=50%). On larger problems, it achieves 77.8% accuracy (mean guess level=13.1%). In another experiment, we show our algorithm outperforms human performance, and the automatic suggestions of new entities resemble those suggested by humans. We hope this work will advance computational analogy by paving the way to more flexible, realistic input requirements, with broader applicability.

1 Introduction

One of the pinnacles of human cognition is the ability to find parallels across distant domains and transfer ideas between them. This *analogous reasoning* process enables us to learn new information faster and solve problems based on prior experience (Minsky, 1988; Hofstadter and Sander, 2013; Holyoak, 1984; PJM, 1966).

The most seminal work in computational analogy is Gentner's Structure Mapping Theory (SMT) (Gentner, 1983) and its implementation, Structure Mapping Engine (SME) (Falkenhainer et al., 1989). In a nutshell, SMT assumes input from two domains: base and target. It maps between objects in a base domain and objects in a target domain according to common *relational structure*, rather than on object attributes.

For example, consider the Rutherford model of the hydrogen atom, where the atom was explained in terms of the (better-understood) solar system (Falkenhainer et al., 1989): a planet revolving around the sun is mapped to an electron revolving around the nucleus. The mapping is due to shared *relations* between objects (revolving around, being attracted to), not object attributes (round, small).

One of the main criticisms brought against SME and its follow-up work is their need for extensive hand-coded input – structured representations of both the entities and their relations (see Figure 1 for the input to the atom/solar system mapping).

Chalmers et al. (1992) argued that too much human creativity is required to construct this input, and the analogy is already effectively given in the representations: "A brief examination [...] shows that the discovery of the similar structure in these representations is not a difficult task. The representations have been set up in such a way that the common structure is immediately apparent. Even for a computer program, the extraction of such common structure is relatively straightforward."

Some follow-up works avoid hand-coding LISP-like representations, generating them from sketches (Forbus et al., 2011), qualitative simulators (Deghghani and Forbus, 2009), etc. However, they still require much knowledge engineering, and thus are hard to scale. Nowadays, when the web is full of information about potential domains to transfer ideas from (McNeil Jr and Odón, 2013), such representations do not tap into the potential of web-scale analogies for augmenting human creativity.

The method with the simplest input we are aware of is Latent Relation Mapping Engine (LRME) (Turney, 2008), which requires only two lists of entities to be mapped. Given two entities, they search for phrases containing both in a large corpus and

Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 16426–16442
December 6-10, 2023 ©2023 Association for Computational Linguistics

Solar System

```
(defEntity sun :type inanimate)
(defEntity planet :type inanimate)

(defDescription solar-system
    entities (sun planet)
    expressions
        (((mass sun) :name mass-sun)
         ((mass planet) :name mass-planet)
         ((greater mass-sun mass-planet) :name >mass)
         ((attracts sun planet) :name attracts)
         ((revolve-around planet sun) :name revolve)
         ((and >mass attracts) :name andı)
         ((cause andı revolve) :name cause-revolve)
         ((temperature sun) :name temp-sun)
         ((temperature planet) :name temp-planet)
         ((greater temp-sun temp-planet) :name >temp)
         ((gravity mass-sun mass-planet) :name force-gravity)
         ((cause force-gravity attracts) :name why-attracts)))
```

Rutherford atom

```
(defEntity nucleus :type inanimate)
(defEntity electron :type inanimate)

(defDescription rutherford-atom
    entities (nucleus electron)
    expressions
        (((mass nucleus) :name mass-n)
         ((mass electron) :name mass-e)
         ((greater mass-n mass-e) :name >mass)
         ((attracts nucleus electron) :name attracts)
         ((revolve-around electron nucleus) :name revolve)
         ((charge electron) :name q-electron)
         ((charge nucleus) :name q-nucleus)
         ((opposite-sign q-nucleus q-electron) :name >charge)
         ((cause >charge attracts) :name why-attracts)))
```

Figure 1: SME representation of the Solar system/Rutherford atom. Reproduced from Falkenhainer et al. (1989).

use them to generate simple patterns. For example, "a sun-centered solar system illustrates" gives rise to patterns such as "a X * Y illustrates". However, such patterns are extremely simple and brittle, and LRME requires exact string matches between the domains (so "revolve around" is different from "rotate around").

In this work, we develop FAME, a Flexible Analogy Mapping Engine. FAME's input requirements are minimal, requiring only two sets of entities. We apply state-of-the-art NLP and IR techniques to automatically infer commonsense relations between the entities using a variety of data sources, and construct a mapping between the domains. Importantly, we do not require identical phrasings of relations. Moreover, our output is interpretable, showing how the mapping was chosen.

Unlike previous works, we drop the strong *bijectivity* assumption and let the algorithm decide which entities to include in the mapping. Meaning, we allow for entities to remain unmapped. Our algorithm can also generate new *suggestions* for the non-mapped entities. This paves the road to algorithms that can handle even more limited input – for example, using domain *names* (solar system, atom) as input, or just a single mapped entity pairs (e.g., turn white blood cells into policemen and see how the analogy unfolds). Our contributions are:

- A novel, scalable, and interpretable approach for automatically mapping two domains based on commonsense *relational* similarities. Our algorithm handles partial mappings and suggests additional entities.
- We extend the work of Romero and Razniewski (2020) to discover salient knowledge about pairs of entities.
- Our model's accuracy is 81.2% on simple, 2x2 problem s(guess level=50%). On larger problems, it achieves 77.8% perfect mappings (guess level=13.1%). In another experiment, we outperform humans (90% vs. 70.2%) and demonstrate that our automatic suggestions resemble human suggestions. We release code and data.[1]

2 Problem Definition

An analogy is a mapping from a base domain $\mathcal{B}$ into a target domain $\mathcal{T}$. The mapping is based on *relations*, and not object attributes. Base objects are not mapped into objects that resemble them; rather, there is a common *relational structure*, and they are mapped to objects that play similar roles. We follow the formulation of Sultan and Shahaf (2022), brought here for completeness:

Entities and Relations. Let $\mathcal{B} = \{b_1, ..., b_n\}$ and $\mathcal{T} = \{t_1, ..., t_m\}$ be two sets of entities. For example: $\mathcal{B}$ = {sun, Earth, gravity, solar system, Newton}, $\mathcal{T}$ = {nucleus, electrons, electricity, atom, Faraday}.

Let $\mathcal{R}$ be a set of relations. A relation is a set of ordered entity pairs with some meaning. The exact representation is purposely vague, as we do not restrict ourselves to strings, embeddings, etc. Intuitively, relations should capture notions like "revolve around".

In our example, relations between $\mathcal{B}$ and $\mathcal{T}$ include the *Earth* revolve around the *Sun*, like *electrons* orbit the *nucleus*; the *Earth* creates a force field of *gravity*, similar to *electrons* creating *electric force* fields; the *Sun* and the *Earth* are part of

[1] https://github.com/shaharjacob/FAME

$\mathcal{B}$	**Mapping**	$\mathcal{T}$
Sun	$\rightarrow$	Nucleus
Earth	$\rightarrow$	Electrons
Gravity	$\rightarrow$	Electric force
Solar system	$\rightarrow$	Atom
Newton	$\rightarrow$	Faraday

Table 1: Illustration of a relational analogy between the solar system and the atom.

the *solar system*, as the *nucleus* and *electrons* are part of the *atom*; *Newton* discovered *gravity*, as *Faraday* is credited with discovering *electric force*.

Note that relation is an asymmetric function, as the pairs are ordered; e.g., Newton discovered gravity, but gravity did not discover Newton.

Slightly abusing notation, we denote the *set* of relations that hold between two entities e_1, e_2 as $\mathcal{R}(e_1, e_2) \subseteq 2^{\mathcal{R}}$. For example, $\mathcal{R}(Earth, Sun)$ contains {revolve around, attracted to}, etc. For clarity, we sometimes use $\mathcal{R}_B$, $\mathcal{R}_T$ to emphasize that the entities belong to the $\mathcal{B}$, $\mathcal{T}$ domain.

Similarity. Let *sim* be a similarity metric between two *sets* of relations, $sim : 2^{\mathcal{R}} \times 2^{\mathcal{R}} \rightarrow [0, \infty)$. Intuitively, when applied to singletons, we want our similarity metric to capture how relations are like each other. For example, "revolve around" is similar to "orbit" and (to a lesser degree) "spiral". When applied to sets of relations, we want *sim* to be higher if the two sets *share many distinct* relations. For example, {revolve around, attracted to} should be more similar to {orbit, drawn into} than to {revolve around, orbit} (as the last set does not include any relation similar to attraction). In Section 3.2 we present our *sim* implementation.

Given one pair from $\mathcal{B}$ and one from $\mathcal{T}$, we define similarity in terms of their relations. Since $\mathcal{R}$ is asymmetric, we consider both directions:

$$sim^*(b_1, b_2, t_1, t_2) = sim(\mathcal{R}_B(b_1, b_2), \mathcal{R}_T(t_1, t_2)) + sim(\mathcal{R}_B(b_2, b_1), \mathcal{R}_T(t_2, t_1))$$

Objective. Our goal is to output a mapping $\mathcal{M} : \mathcal{B} \rightarrow \mathcal{T} \cup \perp$ such that no two $\mathcal{B}$ entities are mapped to the same $\mathcal{T}$ entity (Table 1). Mapping into $\perp$ means the entity was not mapped to any entity in the $\mathcal{T}$ domain.

We look for the mapping $\mathcal{M}^*$ that captures the best inter-domain analogical structure similarity by maximizing the relational similarity:

$$\arg\max_{\mathcal{M}} \sum_{j=1}^{n-1} \sum_{i=j+1}^{n} sim^*(b_j, b_i, \mathcal{M}(b_j), \mathcal{M}(b_i))$$

Note: if b_i or b_j maps to $\perp$, sim^* is defined to be 0.

3 Analogous Matching Algorithm

We wish to find the best mapping from $\mathcal{B}$ to $\mathcal{T}$. We first extract relations between entity pairs from the same domain (Section 3.1). Then, we compute similarity between entity pairs that could be mapped (Section 3.2). Finally, we build the mapping (Section 3.3).

3.1 Relation Extraction

Automatically extracting relations is a key part of our algorithm, as it eliminates the need for extensive manual curation of the input. We focus on *commonsense* relations (e.g., the Earth *revolves around* the Sun), as opposed to situational relations (e.g., the book is on the table). This broadly falls under open information extraction (OIE), the task of generating a structured representation of the information in a text. There has been a lot of work in this area, especially attempts to automate the construction of commonsense datasets (Etzioni et al., 2008, 2004; Yates et al., 2007; Lenat et al., 1985; Sap et al., 2019).

Given two entities, we automatically extract relations from multiple sources:

ConceptNet. A commonsense dataset, containing about 1.5M nodes (Liu and Singh, 2004). For each entity, we receive a list of (predicate, entity), which we filtered to match the second entity (single or plural form). The predicates serve as our relations.

Open Information Extraction. A database automatically extracted from a large web corpus (Etzioni et al., 2008). It contains over 5B triplets of the form (subject, predicate, object). We searched for a match between both entities in the (subject, object) fields, and used the predicates as our relations.

GPT-3 (text-davinci-001).[2] We used a generative pretrained large language model (LM) as a knowledge base in a few-shot manner (Petroni et al., 2019; Brown et al., 2020b). We input a prompt of four analogies, e.g., "Q: What are the relations between gravity and Newton?, A: Newton discovered gravity. A: Newton invented gravity." (see Section

[2]GPT-3 is the only data source that is not freely available. All queries needed for this paper accumulated to less than $50.

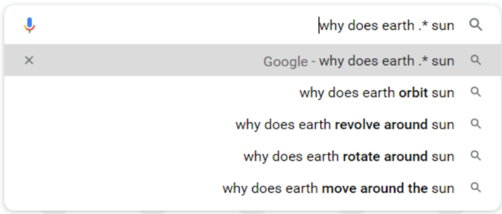


Figure 2: Quasimodo++. Example regex used to extract suggestions from Google ("<question> <entity1> .* <entity2>"). We use questions such as "Why does", "Why did" and "How does".

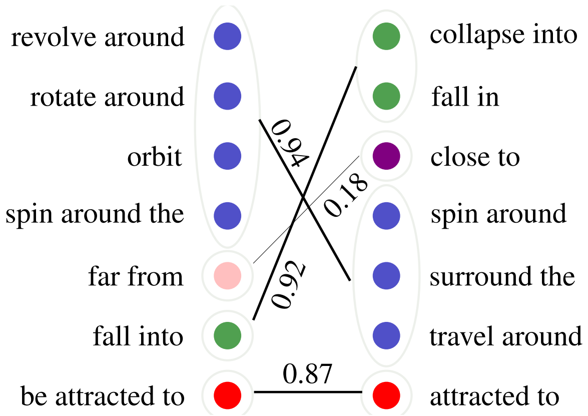


Figure 3: Left: partial relations of *Earth:sun*. Right: partial relations of *electron:nucleus*. This is the result of the maximum weighted match on the clusters. Colors correspond to clusters.

A.2.3 for the full prompt). GPT-3 outputs up to three sentences per query. We kept only sentences of the form <entity> <text> <entity>, treating the <text> as the relation.

Quasimodo. A commonsense knowledge base that focuses on salient properties of objects (Romero and Razniewski, 2020). It contains more than 3.5M triplets of (subject, predicate, object). It considers *questions* instead of statements. For instance, if people search for an answer to "Why is the sky blue?", this implies that the sky is blue. Whenever our two entities appeared in the (subject, object) fields, we extracted their predicates as relations.

Quasimodo++. A relation extraction method that we develop, inspired by Quasimodo. Quasimodo was constructed using questions about a single entity; we extended it to questions exploring relations between *pairs* of entities. We used Google's query auto-completion to tap into the query logs, asking questions containing both desired entities, such as "How does Earth * Sun", "How is Earth * Sun", and "Why does Sun * Earth" for every pair of entities (see Figure 2 for an example). The exact regular expressions we used can be found in Section A.1.

We presented here the knowledge sources we implemented. We note that our algorithm is easy to extend to new sources and that we expect that its robustness will increase with coverage.

3.2 Scoring Entity Pairs

We wish to calculate $sim^*(b_i, b_j, t_k, t_p)$ for $b_{i,j} \in \mathcal{B}$, $t_{k,p} \in \mathcal{T}$, $1 \leq i < j \leq n$, $1 \leq k \neq p \leq m$.

In Section 2 we specified desiderata of *sim*, especially that it is higher if the two sets share many distinct relations. We now present our implementation of *sim*.

Without loss of generality, let us consider $sim(\mathcal{R}_B(b_1, b_2), \mathcal{R}_T(t_1, t_2))$. We first extract all relations $\mathcal{R}_B(b_1, b_2), \mathcal{R}_T(t_1, t_2)$. Next, we calculate the score between each relation in $\mathcal{R}_B(b_1, b_2)$ and each relation in $\mathcal{R}_T(t_1, t_2)$. We create a complete bipartite graph where the left side nodes are the relations of $\mathcal{R}_B(b_1, b_2)$, and the right side nodes are the relations of $\mathcal{R}_T(t_1, t_2)$ (Figure 3). The edge weights (w) are the cosine similarity of the nodes' sBERT embedding (Reimers and Gurevych, 2019).

We remove non-informative relations by extracting the top-frequent n-grams ($n = \{1, 2, 3, 4\}$) from Wikipedia and setting their score to zero. Edges that did not reach a threshold (chosen using hyper-parameter search, see Section 3.3) were set to zero.

Next, we cluster similar relations on each side (e.g., "revolve around" and "circle around") to avoid double-counting. We use hierarchical agglomerative clustering based on the cosine embedding similarity (threshold $= 0.5$; see Section 3.3). The weight of edges between two clusters is the maximal weight of an edge between their nodes (see Figure 3; colors correspond to clusters).

Finally, we apply Maximum-Weight Bipartite Matching on the clusters (see Section 3.3). The similarity score *sim*$(\mathcal{R}_B(b_1, b_2), \mathcal{R}_T(t_1, t_2))$ is defined as the sum of the remaining edges.

3.3 Building a Mapping

Using the score mappings between pairs, we can compose larger mappings. We use beam-search, starting from the most promising pair-mappings found in Section 3.2. In each iteration, we expand the 20 most promising partial mappings, testing each possible mapping between single entities of $\mathcal{B}$ and $\mathcal{T}$ (that are consistent with the current partial

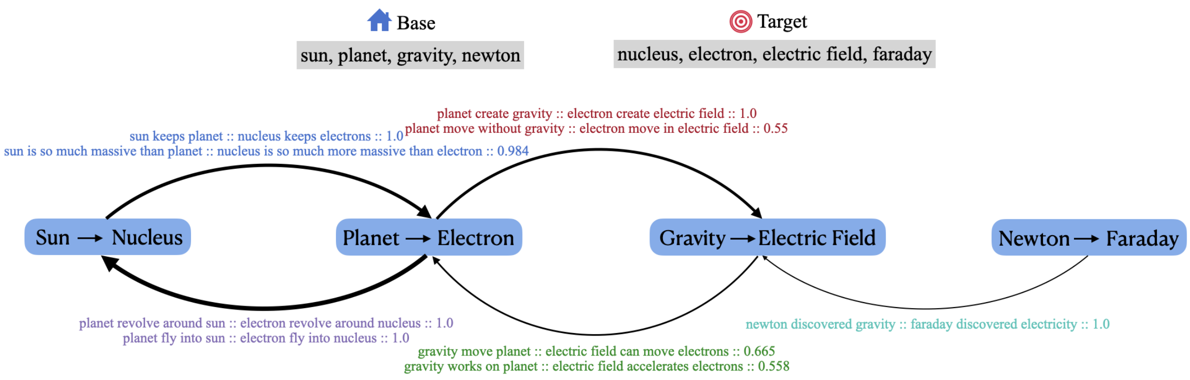


Figure 4: A snippet from our UI. Top: Input. Bottom: The mapping our algorithm found (output), is represented as a graph. Nodes correspond to mappings between single entities (e.g., sun to nucleus). Each edge is annotated with some of the shared relations between the mapped pairs corresponding to its endpoints and their similarity score. For the sake of visualization, we show at most two relations for each edge. Edge weight corresponds to strength.

mapping – i.e., a $\mathcal{B}$ entity cannot map to multiple $\mathcal{T}$ entities). When expansions stop increasing the score, we stop the search and select the mapping with the highest score.

Figure 4 shows a snippet from our UI. Input appears on the top. FAME's output mapping is represented as a graph: nodes correspond to single entity mappings (e.g., Sun to nucleus). Edges represent the shared relational structure. Each edge contains some of the shared relations between the mapped pairs corresponding to its endpoints (e.g., "more massive than") and their similarity score (note the edges are directional). To ease visualization, we show at most two relations per edge. The thickness of an edge corresponds to its weight.

A note on the solution space. In previous works $n = m$ and $\mathcal{M}$ is a *bijective* function. Meaning, $\mathcal{M}$ is both injective (one-to-one; each element in the target is the image of at most one element in the source) and surjective (onto; all the target terms are covered). In other words, no entity is left unmapped. In that case, the solution space's cardinality is $n!$.

We allow for $n \neq m$ and for entities to remain unmapped. Without loss of generality let $n \leq m$. The cardinality is then $\left(\sum_{i=0}^{n} \binom{n}{i} \frac{m!}{(m-i)!}\right) - (n \cdot m)$, where i is the number of matched entities. We subtract $n \cdot m$ because we do not allow for a mapping of size 1; our algorithm starts by mapping pairs and then adds single-entity mapping at each iteration of the beam search.

This relaxation of the bijective constraint drastically increases the space; for $n = 7$, $n! = 5,040$, while our space is of size $130,922$.

Hyper-Parameter Search. We constructed a new dataset to set our model's hyper-parameters (See Appendix A). The dataset contains 36 analogical mapping problems created by ten volunteers, not from our research team. We showed them example analogies and asked them to generate new ones. An expert from our team verified their output, discarding 4 analogies. Domain size varied between 3 to 5 (average size=3.4).

On the problems generated by the volunteers, FAME achieves 83.3% perfect mappings (the whole mapping is correct). If we consider single mappings separately, it achieves 89.4% accuracy.

4 Entity Suggestion

One of the main limitations of previous analogical mapping algorithms is their inability to automatically expand analogies. This is especially interesting in our case, as we allow for unmapped entities; thus, suggesting new entities could identify potential mapping candidates for the unmapped entities.

For example, let $\mathcal{B}$ = {Sun, Earth, gravity, Newton} and $\mathcal{T}$ = {nucleus, electron, electricity}. The correct mapping is Sun $\rightarrow$ nucleus, Earth $\rightarrow$ electron, gravity $\rightarrow$ electricity, leaving Newton with no mapping. Our goal is to suggest candidate entities that preserve the relational structure.

Intuitively, we look at the relations Newton shares with other $\mathcal{B}$ entities (e.g., discovered gravity), and try to see which $\mathcal{T}$ entity plays a corresponding role (i.e., who discovered electricity?).

More formally, suppose we wish to find candidates t^* for mapping to b_n. We first extract the relations of $R_b(b_i, b_n)$, $\forall i \in [n]$ (denoted as R_{b_i}).

Sources	Near	Far	Extended
All	**85%**	**77.5%**	**77.8%**
All-ConceptNet	**85%**	**77.5%**	**77.8%**
All-Open IE	**85%**	67.5%	58.3%
All-Quasimodo	**85%**	**77.5%**	72.2%
All-Quasimodo++	80%	72.5%	72.2%
All-GPT-3	57.5%	50%	66.7%

Table 2: Ablation study on the 2x2 near and far problems and our extended set, leaving out knowledge sources. Results show the importance of the generative LM approach (GPT-3.5) as a knowledge source. Open Information Extraction also contributes much, especially for the complex analogies (2x2-far and extended).

We then iterate over all relations $r \in R_{b_i}$ and use the pair $\{\mathcal{M}(b_i), r\}$ to extract suggestions for t^*.

We use Open Information Extraction, Quasimodo, and Quasimodo++. While our method was previously used to extract *relations* given a pair of two entities, we now use it to extract *entities* given a pair of {entity, relation}. This entails filtering on the predicate field in our commonsense datasets and changing the queries in Quasimodo++.

As suggestions tend to be noisy, we cluster all extracted entities (similarly to the clustering from Section 3.2). We remove clusters of size < 2.

For each suggestion cluster, we rerun our analogous matching algorithm with a representative entity from that cluster (the closest to the cluster's center of mass). We pick the cluster whose representative resulted in the mapping with the highest score. As the commonsense datasets we work with operate mostly on string matching, small changes (e.g., Benjamin Franklin/Ben Franklin) could sometimes result in slightly different results. Thus, we perform one final round, with *all* entities from our chosen cluster, and pick the highest score mapping.

5 Evaluation

In this section, we evaluate FAME. We test its ability to identify the correct mapping (Section 5.1), and compared it to both related works (Section 5.2) and human performance (Section 5.3).

5.1 Performance on Analogy Problems

2x2 problems. One of the things that might have held computational analogy back is the lack of high-quality, large-scale datasets. Most datasets are small and focus on classical 2x2 problems ($A : B :: C : D$), similar to SAT questions.

We start by testing FAME on this standard type of analogies. We use 80 problems from Green et al. (2010), split into 40 near and 40 far analogies (e.g., for "answer:riddle", near analogy is "solution:problem", far analogy is "key:lock"). While the dataset is small, we believe it is still interesting to explore. Our algorithm managed to perfectly map 85% of near analogies and 77.5% of far ones. Random guess baseline is 33.3% (see Section 3.3).

Extended problems. Encouraged by the results of the 2x2 problems, we explore more complex problems. We decided to extend the Green far analogies (which are harder than the near ones). We had three experts go over the dataset together and brainstorm potential extensions. On four problems, the experts did not manage to agree on any additional mappings, leaving us with 36 extended problems (average domain size 3.3).

Our algorithm perfectly mapped 77.8% of the extended problems. Random baseline is 13.1% on average. As we relax the bijection assumption, FAME's average guess level is even lower – 2.2% (see Section 3.3). If we look beyond the top-rated solution, our algorithm has the correct solution in its top-2 guesses 83.3% of the time and 91.7% for top-3.

Error analysis. We found 3 main causes of error:

- **Coverage** (for example, we could not find a relation between "hoof" and "hoofprint"). This prompted us to ablate the knowledge sources FAME uses (Table 2). Results show the importance of the generative LM approach. Open IE is also important, especially for the more complex analogies (far and extended). Some sources, such as ConceptNet, did not seem to contribute much.
- **Noisy relations** that are either peculiar or plain wrong (e.g., "a footballer can iron").
- **Embedding similarity** (for example, "produce" and "is produced by" have a high similarity score). This is exacerbated by **ambiguity** (e.g., the word "pen" referred to "pigpen" and not to the writing instrument).

5.2 Comparison to Related Work

SME line of work. We had difficulty comparing FAME to SME (Falkenhainer et al., 1989) and its extensions, due to their complex input requirements. LRME (Turney, 2008) is closest to our setting, but no code or demo is available. Thus, we compare to their published results on a set of 20 problems.

LRME's entities include nouns, verbs, and adjectives. Since FAME expects noun phrases, we filtered out all other input terms (one problem has only a single noun, so we are left with 19 problems). It is hard to compare in this setup (and unfortunately, authors did not report which partial mappings were correct). Still, LRME's accuracy was 75%, whereas FAME achieved 84.2%.

While the size of the problems is smaller when restricted to nouns, we believe the noun-only setting is harder. The verbs and adjectives often provide hints that significantly constrain the search space. For example, in problem A6 (Turney, 2008) (mapping a projectile to a planet) there is one adjective in each domain (parabolic, elliptical). Those adjectives can only apply to one or two of the nouns (i.e., you cannot have parabolic earth, air, or gravity), effectively giving away the noun mapping.

As a side note, we also believe that our noun-only input is a cleaner problem setting, as it is often easier to automatically identify the entities in a domain than to identify the attributes and verbs relevant for the analogy. In the words of LRME's authors, "LRME is not immune to the criticism of Chalmers et al. (1992), that the human who generates the input is doing more work than the computer that makes the mapping." We believe FAME is a step in the right direction in this regard.

Pretrained LMs. In the absence of a baseline, we turn to a generative pretrained large LM known to have impressive commonsense abilities – GPT-3.5 (text-davinci-002). We used 4 random examples from the hyper-parameter search dataset. After some experimentation with prompt engineering, we chose two variants (see A.2.3).

The results are summarized in Table 3. GPT-3.5 does well on the 2x2 datasets (Green et al., 2010). However, both datasets appear on the web, and perhaps GPT-3.5 was exposed to them during training (data leakage). In particular, we found some of the answers via a simple web search (Figure A.6).

Moreover, GPT-3.5's performance drops on the extended set, where problems are complex and do not appear on the web. Interestingly, it does not even manage to return a valid mapping in some of the cases. This exercise improves our understanding of FAME's strengths and weaknesses.

E-KAR dataset. Chen et al. (2022) recently released a relevant dataset, E-KAR, for rationalizing analogical reasoning. The dataset consists of multiple-choice problems from civil service exams in China. For example, for the source triplet "tea:teapot:teacup", the correct answer is "talents:school:enterprise". The reasoning is that both teapot and teacup are containers for tea. After the tea is brewed in the teapot, it is transported into the teacup. Similarly, both school and enterprise are organizations. After talents are educated in school, they are transported into enterprise.[3]

Algorithm	Near	Far	Extended
FAME	85.0%	77.5%	**77.8%**
GPT-3.5 ":"	**92%**	**80%**	44%
GPT-3.5 "–>"	88%	**80%**	58%

Table 3: Comparison of FAME and GPT-3.5. GPT-3.5 does well on the 2x2 datasets (far and near). We note that data leakage is a concern. GPT-3.5's performance sharply drops on the extended problem set, where problems are bigger and do not appear on the web.

The E-KAR test set has no labels, so we used their validation set (N=119) to test FAME. As our task is different, we only took source entities (as $\mathcal{B}$) and entities from the correct answer (as $\mathcal{T}$). We filtered questions without nouns, resulting in N=101.

FAME found the right mapping 68.3% of the time. A closer examination of FAME's mistakes revealed that $\sim 75\%$ of them occurred due to relation *types* that are not at all covered by our framework: either ternary relations (soldier:doctor:military doctor $\rightarrow$ car:electric vehicle:electric car; the last term is a combination of the first two) or relations based on sharing some attribute (so "both containers for holding tea" is mapped to "both are organizations"). Some of the attribute-based mappings work at the whole-set level, so each entity on $\mathcal{B}$ could map to each entity on $\mathcal{T}$ (yellow:red:white $\rightarrow$ sad:happy:angry). Thus, we conclude there is a big gap between FAME and E-KAR's assumptions.

5.3 Comparison to People

We compare FAME with *human thinking* in a 2-phase experiment.[4] In the closed-world phase, the participants received ten structure mapping problems, in which they were asked to match instances from $\mathcal{B}$ to $\mathcal{T}$. The domains included between 3-5 entities (Table A.4). Participants were instructed to map each $\mathcal{B}$ entity into exactly one $\mathcal{T}$ entity.

[3]Interestingly, the authors of this paper thought that the "passengers:bus:taxi" answer was the correct one, based on containment and size relations.

[4]The experiment received ethics committee approval. See full instructions in Section A.4.

In the open-world phase, participants received five *mapped* problems, but one entity was left blank (Table A.5). Participants were instructed to fill in the blank with an entity that preserves the analogy.

Participants. We recruited 304 participants using social media. The compensation was a chance to win one of three $30 vouchers. 76.6% of our participants were between the ages 18-35 and 17.2% are between 36-45 (self-reported).

Closed-world mapping. FAME missclassified one problem compared to gold standard ($A9$, Table A.4), achieving 90% accuracy (human baseline was 70.2%; see full distribution in Table A.4).

Problem $A6$ has the lowest human accuracy (35.5%), and is also the largest one ($|\mathcal{B}| = |\mathcal{T}| = 5$). A closer examination of its confusion matrix reveals that while FAME correctly mapped *water* to *heat* and *pressure* to *temperature*, 15% of people switched the two. This might be due to the strong semantic pairing of *water* and *temperature*. FAME is immune to this, as it relays on *relations*.

On average, each participant mapped the problem the same as FAME 78% of the times. Overall, FAME outperforms humans, and most of the disagreement is due to human errors.

Open-world entity suggestion. We presented participants with five *mapped* problems where one entity was left blank (Table A.5) and asked them to fill in the black while preserving the analogy.

For all five problems, an entity from FAME's top two completions appeared in humans' top three completions (Table A.6). Meaning, our algorithm's top suggestions are similar to humans'. Only in one example ($B5$) one of the top two algorithm's completions appeared *third* in humans' (in the rest it is first or second). We suspect that this confusion in $B5$ occurred because *gravity* and *Newton* reminded participants of the term *apple*.

Figure 5 shows a word cloud for answers to problem $B1$. While most responses are quite similar, some participants returned creative and appropriate solutions (e.g., treasure chest, jewelry box, car).

Figure 5: Word cloud of human completions for $B1$ (Table A.6). While most responses were from the same semantic domain, some were creative and appropriate (e.g., treasure chest, jewelry box, car).

6 Related Work

Computational analogy-making dates back to the 1960s (Evans, 1964; Reitman, 1965). Analogy-making approaches are broadly categorized as symbolic, connectionist, and hybrid (French, 2002; Mitchell, 2021; Gentner and Forbus, 2011).

Symbolic approaches usually represent input as structured sets of logic statements. Our work falls under this branch, as well as SME (Falkenhainer et al., 1989) and its follow-up work. LRME (Turney, 2008) is the closest to our work, as it automatically extracts the relations. Unlike FAME, LRME requires exact matches of relations across different domains. We also focus on nouns only, making the problem harder, and relax the bijection assumption, allowing for automatically extending analogies.

NLP. Analogy-making received relatively little attention in NLP. The best-known task is word analogies, often used to measure embeddings' quality (inspired by Word2Vec's *"king - man + woman = queen"* example (Mikolov et al., 2013)). Follow-up work explored embeddings' linear algebraic structure (Arora et al., 2016; Gittens et al., 2017; Allen and Hospedales, 2019) or compositional nature (Chiang et al., 2020), neglecting relational similarity. A recent work on analogies between procedural texts (Sultan and Shahaf, 2022) did study relational similarity, but extracted the relations from the input texts, with no commonsense augmentations.

Recently, there have been efforts to study LMs' analogical capabilities (Ushio et al., 2021; Brown et al., 2020a). Findings indicate they struggle with abstract and complex relations and results depend strongly on LM's architecture and parameters.

Kittur et al. (2019) combined NLP and crowds for product analogies without explicitly modeling entities and relations, but instead automatically extracting *schemas* of the product.

7 Conclusions and Future Work

Detecting deep structural similarity across distant domains and transferring ideas between them is central to human thinking. We presented FAME, a novel method for analogy making. Compared to previous works, FAME is more expressive, scalable, robust and interpretable. It also allows partial matches and automatic entity suggestions to extend the analogies.

FAME correctly maps 81.2% of classical 2x2 analogy problems. On larger problems, it achieves 77.8% perfect mappings (mean guess level=13.1%). FAME also outperforms humans in solving analogy mapping problems (90% vs. 70.2%). Interestingly, our automatic suggestions of new entities resemble those suggested by humans.

In future work, we plan to improve coverage and extend our framework to more than just binary relations, as sometimes the key to an analogy is a relation involving more than two objects. In addition, we plan to improve our similarity measure, to address both context (to solve ambiguity) and the difference between active and passive relations. We plan to explore different forms of input, such as algorithms that take as input very partial domains, perhaps even just domain *names* (e.g., solar system, atom) and populate the domains with entities, or algorithms incorporating *user feedback*.

To conclude, we hope FAME will pave the way for analogy-making algorithms that require less-restrictive inputs and can scale up and tap into the vast amount of potential inspiration the web offers, augmenting human creativity.

8 Ethical Considerations & Limitations

While FAME can assist humans by inspiring non-trivial solutions to problems, it has been shown that humans struggle with detecting caveats in presented analogies (Holyoak et al., 1995). For example, the cardiovascular system is often taught to medical students in terms of water supply system (Swain, 2000). However, this analogy might also confuse them, as it ignores important differences between water and blood (e.g., blood clots). Thus, while our output is interpretable, it might still mislead people, and it is important to alert the users to this possibility.

Another issue is the fact that FAME's coverage highly depends on external resources (ConceptNet, Google AutoComplete, etc.). This might be particularly problematic when applied to low-resource languages. As the relations we look for are common-sense relations, rather than cultural or situational ones, using automatic translation might ameliorate the problem.

Lastly, we also note these resources evolve over time, and thus if one is interested in reproducibility, it is necessary to save snapshots of the extracted relations.

Acknowledgements

We thank the reviewers for their insightful comments. This work was supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant no. 852686, SIAM).

In memory of the more than one thousand victims of the horrific massacre carried out by Hamas terrorists on October 7th, 2023.

References

Carl Allen and Timothy Hospedales. 2019. Analogies explained: Towards understanding word embeddings. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 223–231. PMLR.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2016. A latent variable model approach to pmi-based word embeddings. *Transactions of the Association for Computational Linguistics*, 4:385–399.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020a. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020b. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

David J Chalmers, Robert M French, and Douglas R Hofstadter. 1992. High-level perception, representation, and analogy: A critique of artificial intelligence methodology. *Journal of Experimental & Theoretical Artificial Intelligence*, 4(3):185–211.

Jiangjie Chen, Rui Xu, Ziquan Fu, Wei Shi, Zhongqiao Li, Xinbo Zhang, Changzhi Sun, Lei Li, Yanghua

Xiao, and Hao Zhou. 2022. E-kar: A benchmark for rationalizing natural language analogical reasoning. *arXiv preprint arXiv:2203.08480*.

Hsiao-Yu Chiang, Jose Camacho-Collados, and Zachary Pardos. 2020. Understanding the source of semantic regularities in word embeddings. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 119–131, Online. Association for Computational Linguistics.

Morteza Dehghani and Ken Forbus. 2009. Qcm: A qp-based concept map system. In *the 23nd International Workshop on Qualitative Reasoning (QR09)*, pages 16–21.

Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S Weld. 2008. Open information extraction from the web. *Communications of the ACM*, 51(12):68–74.

Oren Etzioni, Michael Cafarella, Doug Downey, Stanley Kok, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S Weld, and Alexander Yates. 2004. Web-scale information extraction in knowitall: (preliminary results). In *Proceedings of the 13th international conference on World Wide Web*, pages 100–110.

Thomas G Evans. 1964. A heuristic program to solve geometric-analogy problems. In *Proceedings of the April 21-23, 1964, spring joint computer conference*, pages 327–338.

Brian Falkenhainer, Kenneth D Forbus, and Dedre Gentner. 1989. The structure-mapping engine: Algorithm and examples. *Artificial intelligence*, 41(1):1–63.

Kenneth Forbus, Jeffrey Usher, Andrew Lovett, Kate Lockwood, and Jon Wetzel. 2011. Cogsketch: Sketch understanding for cognitive science research and for education. *Topics in Cognitive Science*, 3(4):648–666.

Robert M French. 2002. The computational modeling of analogy-making. *Trends in cognitive Sciences*, 6(5):200–205.

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

Dedre Gentner and Kenneth D Forbus. 2011. Computational models of analogy. *Wiley interdisciplinary reviews: cognitive science*, 2(3):266–276.

Alex Gittens, Dimitris Achlioptas, and Michael W Mahoney. 2017. Skip-gram- zipf+ uniform= vector additivity. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 69–76.

Adam E Green, David JM Kraemer, Jonathan A Fugelsang, Jeremy R Gray, and Kevin N Dunbar. 2010. Connecting long distance: semantic distance in analogical reasoning modulates frontopolar cortex activity. *Cerebral cortex*, 20(1):70–76.

Douglas R Hofstadter and Emmanuel Sander. 2013. *Surfaces and essences: Analogy as the fuel and fire of thinking*. Basic books.

Keith J Holyoak. 1984. Analogical thinking and human intelligence. *Advances in the psychology of human intelligence*, 2:199–230.

Keith J Holyoak, Paul Thagard, and Stuart Sutherland. 1995. Mental leaps: analogy in creative thought. *Nature*, 373(6515):572–572.

Aniket Kittur, Lixiu Yu, Tom Hope, Joel Chan, Hila Lifshitz-Assaf, Karni Gilon, Felicia Ng, Robert E Kraut, and Dafna Shahaf. 2019. Scaling up analogical innovation with crowds and ai. *Proceedings of the National Academy of Sciences*, 116(6):1870–1877.

Douglas B Lenat, Mayank Prakash, and Mary Shepherd. 1985. Cyc: Using common sense knowledge to overcome brittleness and knowledge acquisition bottlenecks. *AI magazine*, 6(4):65–65.

Hugo Liu and Push Singh. 2004. Conceptnet—a practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226.

Donald G McNeil Jr and Mr Odón. 2013. Car mechanic dreams up a tool to ease births. *The New York Times*, 13.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Marvin Minsky. 1988. *Society of mind*. Simon and Schuster.

Melanie Mitchell. 2021. Abstraction and analogy-making in artificial intelligence. *Annals of the New York Academy of Sciences*, 1505(1):79–101.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*.

PJM. 1966. Models and analogies in science.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Walter Ralph Reitman. 1965. Cognition and thought: an information processing approach.

Julien Romero and Simon Razniewski. 2020. Inside quasimodo: Exploring construction and usage of commonsense knowledge. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 3445–3448.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. 2019. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3027–3035.

Oren Sultan and Dafna Shahaf. 2022. Life is a circus and we are the clowns: Automatically finding analogies between situations and processes. *Proceedings of the 2022 conference on empirical methods in natural language processing (EMNLP)*.

David P Swain. 2000. The water-tower analogy of the cardiovascular system. *Advances in Physiology Education*, 24(1):43–50.

Peter D Turney. 2008. The latent relation mapping engine: Algorithm and experiments. *Journal of Artificial Intelligence Research*, 33:615–655.

Asahi Ushio, Luis Espinosa Anke, Steven Schockaert, and Jose Camacho-Collados. 2021. Bert is to nlp what alexnet is to cv: Can pre-trained language models identify analogies? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3609–3624.

Alexander Yates, Michele Banko, Matthew Broadhead, Michael J Cafarella, Oren Etzioni, and Stephen Soderland. 2007. Textrunner: open information extraction on the web. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 25–26.

A Implementation Details

We fine-tune our model using 36 problems described in Section 3.3.

We used the pre-trained model *msmarco-distilbert-base-v4* which is based on sBERT (Reimers and Gurevych, 2019). We set the similarity threshold (the similarity between two relations) to be **0.2** (range checked: 0-0.6). We set the number of top n-grams which was filtered (the top frequencies n-grams in Wikipedia) to **500**. The clustering distance threshold is set to **0.5** (range checked: 0.3-0.9). The number of clusters we consider when computing the sum is set to **3** (range checked: 1-maximum number of clusters). We set the beam search size to **20** (range checked: 1-40). All of these parameters describes in Section 3.

We provide access to our anonymous repository can be found[1]. We note that the usage of Docker is not supported in this version for the purpose of maintaining anonymity. However, the algorithmic content is available.

A.1 Quasimodo++ regular expressions

We use the following regex for our Quasimodo++: "<question> <prefix> <entity1> .* <entity2>". The questions we used are: {"why do", "why is", "why does", "why does it", "why did", "how do", "how is", "how does", "how does it", "how did"}. The prefix is optional and can be {"a", "an" and "the"}. We use both singular and plural forms of the entities.

A.2 GPT-3

A.2.1 Prompts used for relation extraction

The prompt used for GPT-3 is:
Q: What are the relations between a blizzard and snowflake?
A: A blizzard produces snowflakes.
A: A blizzard contains a lot of snowflakes.

Q: What are the relations between an umbrella and rain?
A: An umbrella protects from rain.
A: An umbrella provides adequate protection from rain.

Q: What are the relations between a movie and screen?
A: A movie displayed on a screen.
A: A movie can be shown on a screen.

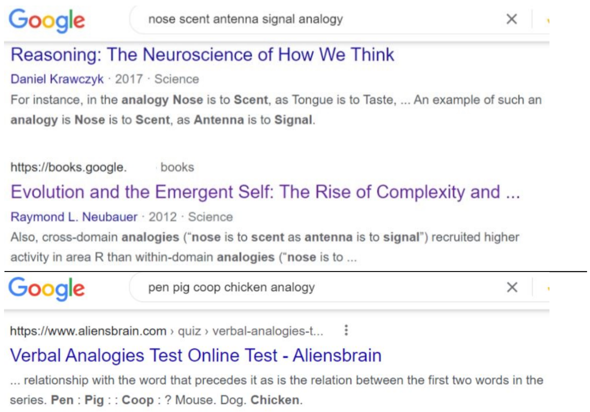


Figure 6: Looking for analogies from the original Green eval dataset online.

Q: What are the relations between Newton and gravity?
A: Newton discovered gravity.
A: Newton invented gravity.

Q: What are the relations between an electron and nucleus?
A: An electron revolves around the nucleus.
A: An electron is much smaller than the nucleus.
A: An electron attracts the nucleus.

Q: What are the relations between water and a pipe?
A: Water flows through the pipe.
A: Water passes through the pipe.

A.2.2 Prompts used for baseline comparison

After some experimentation with prompt engineering, we chose two variants of the prompt:

Q: Find an analogical mapping between the entities "eraser", "paper" and "pencil" and the entities "keyboard", "delete" and "screen".
A: eraser:pencil:paper::delete:keyboard:screen
(or)
A: eraser -> delete, pencil -> keyboard, paper -> screen

A.2.3 Possible leakage

Example answers for chosen analogies from Green eval dataset found via a simple web search can be found in Figure 6

A.3 Repository

To ease the access and usage of our code we use Docker. Its main goal is to shift the cross-platform installation burden from the user to the developer. Unfortunately, we cannot share our Docker due to anonymity concerns (username). We will include it in the non-anonymized version.

We provide a React based web interface, currently available only locally. This system is used to visualize the graphs created by the algorithm's mapping output. In addition, it visualizes the relations between entities, their similarity, and the clustering. This interface is useful for assisting in developing, debugging and understanding the algorithm's output. The demo is accessible using our repository[1].

A.4 Experiments

Snippets of the experimental setup (including instructions) can be found in Figures 7, 8.

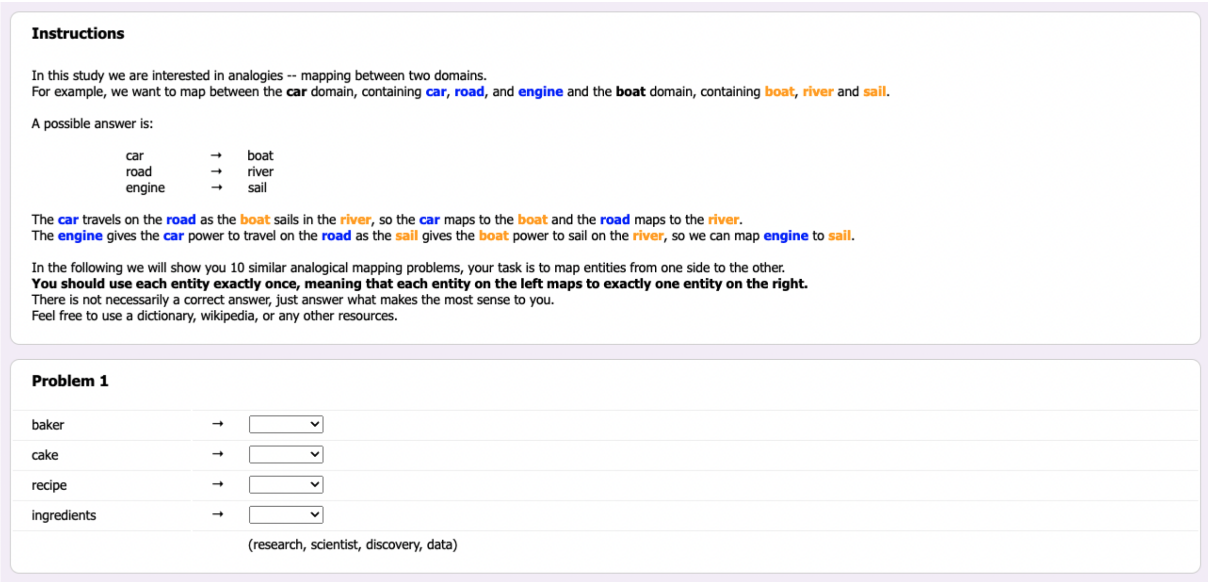
Table 4 depicts the ten analogical proportion problems used in the *structure mapping* experiment (closed-world mappings in Section 5.2). Accuracy denotes the percentage of human participants who mapped from $\mathcal{B}$ to $\mathcal{T}$ correctly. Results show this task is non-trivial even for humans.

Table 6 illustrates the experimental setup for the second phase of our experiment, in which participants received a solved mapping problem with one entity left out (open-World in Section 5.2).

Table 5 contains all solved analogy problems used in the second phase of the experiment (entity suggestion, see open-World in Section 5.2). Participants were given with the complete mapping, but with a missing entity (as presented here).

A.5 E-kar

Table 7 shows an example of a problematic problem from E-KAR dataset.



Instructions

In this study we are interested in analogies -- mapping between two domains.
For example, we want to map between the **car** domain, containing **car**, **road**, and **engine** and the **boat** domain, containing **boat**, **river** and **sail**.

A possible answer is:

car	→	boat
road	→	river
engine	→	sail

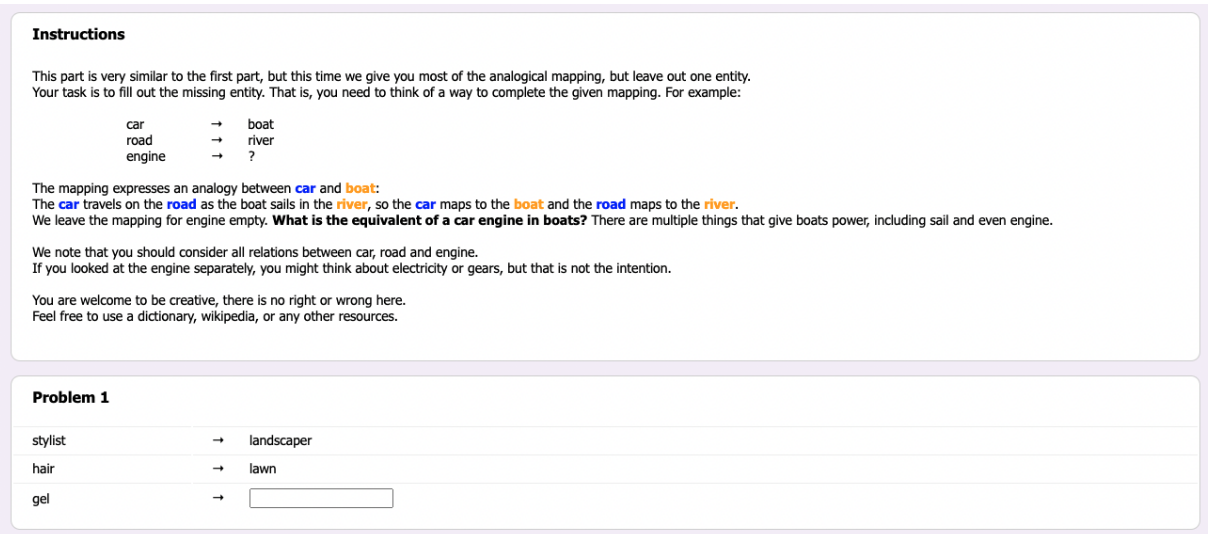
The **car** travels on the **road** as the **boat** sails in the **river**, so the **car** maps to the **boat** and the **road** maps to the **river**.
The **engine** gives the **car** power to travel on the **road** as the **sail** gives the **boat** power to sail on the **river**, so we can map **engine** to **sail**.

In the following we will show you 10 similar analogical mapping problems, your task is to map entities from one side to the other.
You should use each entity exactly once, meaning that each entity on the left maps to exactly one entity on the right.
There is not necessarily a correct answer, just answer what makes the most sense to you.
Feel free to use a dictionary, wikipedia, or any other resources.

Problem 1

baker	→	
cake	→	
recipe	→	
ingredients	→	
		(research, scientist, discovery, data)

Figure 7: Closed-World Mapping: Experiment instructions with the first question.



Instructions

This part is very similar to the first part, but this time we give you most of the analogical mapping, but leave out one entity.
Your task is to fill out the missing entity. That is, you need to think of a way to complete the given mapping. For example:

car	→	boat
road	→	river
engine	→	?

The mapping expresses an analogy between **car** and **boat**:
The **car** travels on the **road** as the boat sails in the **river**, so the **car** maps to the **boat** and the **road** maps to the **river**.
We leave the mapping for engine empty. **What is the equivalent of a car engine in boats?** There are multiple things that give boats power, including sail and even engine.

We note that you should consider all relations between car, road and engine.
If you looked at the engine separately, you might think about electricity or gears, but that is not the intention.

You are welcome to be creative, there is no right or wrong here.
Feel free to use a dictionary, wikipedia, or any other resources.

Problem 1

stylist	→	landscaper
hair	→	lawn
gel	→	

Figure 8: Open-World Entity Suggestion: Experiment instructions with the first question.

	$\mathcal{B}$	**Mapping**	$\mathcal{T}$	**Human Accuracy (Guess Level)**
A1	Baker Cake Recipe Ingredients	$\rightarrow$	Scientist Discovery Research Data	79.6% (4.2%)
A2	Eraser Pencil Paper	$\rightarrow$	Amnesia Memory Mind	71.7% (16.7%)
A3	Jacket Zipper Cold	$\rightarrow$	Wound Suture Infection	68.8% (16.7%)
A4	Train Track Steel	$\rightarrow$	Signal Wire Copper	74.0% (16.7%)
A5	Thoughts Brain Neurons	$\rightarrow$	Astronaut Space Stars	53.9% (16.7%)
A6	Water Pressure Bucket Pipe Rain	$\rightarrow$	Heat Temperature Kettle Iron Sun	35.5% (0.8%)
A7	Waves Water Shore Breakwater	$\rightarrow$	Sounds Air Ear Earplugs	65.1% (4.2%)
A8	Goal Soccer Grass Feet	$\rightarrow$	Basket Basketball Hardwood Hands	94.1% (4.2%)
A9	Seeds Fruit Bloom	$\rightarrow$	Ideas Product Success	64.5% (16.7%)
A10	Morning Breakfast Start Coffee	$\rightarrow$	Evening Dinner End Wine	95.1% (4.2%)

Table 4: The ten analogical proportion problems used in the *structure mapping* experiment. Accuracy denotes the percentage of human participants who mapped from $\mathcal{B}$ to $\mathcal{T}$ correctly. Note that each row under the $\mathcal{B}$ column is mapped to its $\mathcal{T}$ column. Problem's guess level appears in brackets below the accuracy. Results show this task is non-trivial even for humans.

$\mathcal{B}$	**Mapping**	$\mathcal{T}$
Electrons	$\rightarrow$	Earth
Electricity	$\rightarrow$	Gravity
Faraday	$\rightarrow$	Newton
Nucleus	$\rightarrow$	?

Table 5: Solved mapping problem with one missing $\mathcal{T}$ entity. Participants instructed to fill in the missing entity.

	$\mathcal{B}$		$\mathcal{T}$	**Algorithm**	**Humans**
	Answer		Key		
B1	Logic	$\rightarrow$	Mechanism		
	Riddle		**?**		
				Problem	**Lock** (58.9%)
				Lock	Door (11.8%)
				Feedback	Question (4.6%)
	Earth		Electrons		
	Gravity		Electricity		
B2	Newton	$\rightarrow$	Faraday		
	?		Nucleus		
				Sun	Earth's core (15.8%)
				Moon	Apple (13.2%)
				Mars	**Sun** (10.2%)
	Stylist		Landscaper		
B3	Hair	$\rightarrow$	Lawn		
	Gel		**?**		
				Fertilizer	**Fertilizer** (29.3%)
				Water	Lawn Mower (21.1%)
				Lime	Shears (10.2%)
	Chef		Baker		
	Meal		Cake		
B4	Pan	$\rightarrow$	Oven		
	Salt		**?**		
				Butter	**Sugar** (63.5%)
				Sugar	Flour (6.9%)
				Onion	Pepper (3.3%)
	Sun		Rain		
B5	Summer	$\rightarrow$	Winter		
	Sunscreen		**?**		
				Umbrella	**Umbrella** (51.0%)
				Birds	Coat (20.7%)
				Flooding	Cream (9.9%)

Table 6: Examples used in the second phase of the experiment. Participants were given with the complete mapping, but with a missing entity (as presented here). The algorithm top three completions are sorted according to certainty. Humans' top three completions are sorted according to their frequency in the experiment (in brackets).

$\mathcal{B}$	**Mapping**	$\mathcal{T}$
Ice	$\rightarrow$	Grass
Fog	$\rightarrow$	Tree

Table 7: "ice" and "fog" are different forms of the same substance, and both "ice" and "fog" are natural objects.". "grass" and "tree" are both plants, and "grass" and "tree" are both natural objects.