# Scaling Creative Inspiration with Fine-Grained Functional Aspects of Ideas

Tom Hope
tomh@allenai.org
Allen Institute for AI
The University of Washington

Ronen Tamari
Hebrew University of Jerusalem

Hyeonsu Kang
Carnegie Mellon University

Daniel Hershcovich
University of Copenhagen, Denmark

Joel Chan
University of Maryland

Aniket Kittur
Carnegie Mellon University

Dafna Shahaf
dshahaf@cs.huji.ac.il
Hebrew University of Jerusalem

## ABSTRACT

Large repositories of products, patents and scientific papers offer an opportunity for building systems that scour millions of ideas and help users discover inspirations. However, idea descriptions are typically in the form of unstructured text, lacking key structure that is required for supporting creative innovation interactions. Prior work has explored idea representations that were either limited in expressivity, required significant manual effort from users, or dependent on curated knowledge bases with poor coverage. We explore a novel representation that automatically breaks up products into fine-grained functional aspects capturing the purposes and mechanisms of ideas, and use it to support important creative innovation interactions: functional search for ideas, and exploration of the design space around a focal problem by viewing related problem perspectives pooled from across many products. In user studies, our approach boosts the quality of creative search and inspirations, substantially outperforming strong baselines by 50-60%.

## 1 INTRODUCTION

Human creativity often relies on detecting *structural matches* across distant ideas and *adapting* them by transferring mechanisms from one domain to another [16, 17, 37, 38]. For example, microwave ovens were discovered by *repurposing* radar technology developed during World War II. Teflon, today chiefly used in non-stick cookware, was first used in armament development. Recognizing the

potential of this innovation process, major organizations such as NASA and Procter & Gamble actively engage in searching for opportunities to adapt existing technologies for new markets [27].

Online repositories of millions of products, scientific papers, and patents present an opportunity to augment and scale this core process of innovation. The large scale and diversity of these repositories is important, because inspiration can be found in unexpected places – for example, a car mechanic recently invented a simple device to ease childbirths by adapting a trick for extracting a cork stuck in a wine bottle, which he discovered in a YouTube video [1].

However, the predominant way human problem-solvers currently interact with these repositories — via standard search engines — does not tap into their potential for augmenting and scaling human ingenuity. Core to this limitation is the representation of ideas in the form of unstructured textual descriptions. This representation hinders creative innovation interactions that require traversing multiple levels of granularity and abstraction around a focal problem, to "break out" of fixation on the details of a specific problem by exploring the design space and viewing novel perspectives on problems and solutions [15, 23, 60, 65].

Toward addressing this challenge, our vision in this paper is to develop a novel representation of ideas that can support **exploration and abstraction of fine-grained functional aspects in large-scale idea repositories** – aspects such as the purposes and mechanisms of products. More specifically, our goal is to obtain a representation having two key capabilities: (I) The representation would be able to automatically disentangle raw descriptions into **fine-grained functional "units"** that support search and discovery of products that match on certain functions but not others. (II) This representation should also allow navigating the landscape of ideas at **different resolutions** — enabling users to "zoom" in and out at desired levels of abstraction of a given problem and connect to inspirations in seemingly distant areas.

As an example, consider an inventor looking for a way to wash clothes without water (e.g., in space, or where water is scarce). Figure 1 illustrates our vision. Breaking down product descriptions into fine-grained functions (*capability I* above) could allow an automated system to find ideas that match on certain purpose aspects

(*washing clothes*) but not certain mechanism aspects (*usage of water*). This could lead to solutions like cleaning mechanisms based on dry ice or chemical coating.

Zooming out and abstracting the problem to a more general framing (*capability II*) might lead to broader ideas for the problem of *cleaning* such as techniques for removing dirt or odor – each resulting in novel problem perspectives and inspirations. In Figure 1, each node represents a cluster of documents with a similar purpose and the user can explore neighboring clusters to find inspirations (e.g., dry shampoo). This can also expand the innovator's conception of the problem space itself, such as the assumption that clothes should be cleaned and reworn (vs. biodegradable).

In this paper, we develop a scalable computational model of ideas that brings us closer to this vision. We train a neural network to extract spans of text describing purposes and mechanisms in product texts, and use them to build a span embedding representation that allows aspect-based matching between ideas. We then use this representation of *individual* ideas to automatically mine connections between problems and solutions across *entire repositories* and build a "functional network" that resembles functional ontologies used in engineering and design ideation [39, 51], which are typically hand-crafted and limited in scale.

Our approach could facilitate many applications in creative innovation due to its ability to decompose idea texts into fine-grained functional aspects, and to surface related problem perspectives at multiple levels of abstraction — two fundamental drivers of creativity support. In this paper we instantiate the approach in two prototype systems, probing its value regarding each of these capabilities:

- **Functional aspect-based search for alternative uses.** One important use of our novel representation is to enhance the expressivity of search engines over idea repositories. This way, our representation could support expressive search for *alternative, atypical uses of products* to identify potentially high-value adaptation opportunities.
- **Exploring alternative levels of problem perspectives.** Recent work [42] showed that problem-solvers are often interested in exploring different reformulations of the problem when searching for inspiration. Our fine-grained span representations facilitate mining of recurring functional relations, such as purposes that are often mentioned together or mechanisms associated with purposes. This level of detail enables us to map the landscape of ideas with a network of purposes and mechanisms, allowing us to automatically traverse neighboring problems and solutions around a focal problem and surface novel inspirations to users.

Previous work highlighted the importance of functional representations for supporting ideation [12, 42, 52, 60, 65, 100], but these methods require significant *manual effort* from the user, rely on resources with limited coverage, or have limited expressivity (we discuss this work in more detail in §2). We seek to advance the state of the art by developing a novel representation that is both expressive and scalable, and exploring the applications it unlocks. We believe our representation may serve as a useful building block for novel creativity support tools that can help users find and recombine the inspirations latent in unstructured idea repositories at a scale previously impossible.

**In summary, in this work we contribute**:

- A novel computational representation of ideas with fine-grained functional aspects for purposes and mechanisms.
- Empirical demonstrations of the flexibility and utility of the representation for computational support of core creative tasks: (1) searching for *alternative, atypical product uses* for potential adaptation opportunities; and (2) creating a *functional concept graph* that enables innovators to explore the design space around a focal problem. Through two empirical user studies we demonstrate that our representation significantly outperforms both previous work and state-of-the art embedding baselines on these tasks. We achieve Mean Average Precision (MAP) of 87% in the alternative product uses search, and 62% of our inspirations for design space exploration are found to be *useful and novel* – a relative boost of 50-60% over the best-performing baselines.

## 2 RELATED WORK: IDEATION SYSTEMS AND COGNITIVE MECHANISMS

The contributions in this paper relate broadly to previous work on systems that use structured representations for supporting ideation, and studies that seek to understand the cognitive process of creativity and its implications. We provide a brief discussion of these two themes.

**Cognitive Theories of Creative Thinking with Prior Ideas.** A core aspect of creative thinking that distinguishes it from regular problem solving is the need for divergent thinking [21, 85, 87] - to construct and explore diverse ideas that are quite different from the obvious path of ideation. This process of divergence is shaped by prior knowledge [97] — such as past ideas, external stimulation, and examples — in ways that sometimes hinders creativity through mechanisms like fixation [54], and sometimes leads to creative breakthroughs [92].

Our design goals for this research are guided by past research on cognitive mechanisms that enable helpful interactions with past ideas. For example, research on insight problem-solving has uncovered the role of re-representation of past ideas by decomposing them into conceptual chunks and then recombining and/or repurposing them into new solutions [61, 73]. Similar patterns in terms of core helpful interactions with prior ideas have been observed in in situ studies of expert designers' dissection of past ideas into component aspects and features for repurposing and recombination into new ideas [26, 48, 49]. Another core process is analogical abstraction, where innovators think about and retrieve past ideas not in terms of their surface features, but in terms of deeper structural features or schemata, such as their underlying *purposes* (goals) and *mechanisms* [40, 41]. These abstracted "schemas" can then facilitate analogical transfer of ideas across domains that can lead to groundbreaking discoveries [19, 50, 60, 72, 79]. For example, the ancient Greeks studied the properties of sound waves by analogy to ripples in water; Nobel laureate Salvador Luria used abstract structural similarity between a slot machine and bacteria mutations to understand

Exploring the Purpose Graph                    Example Documents from Clusters
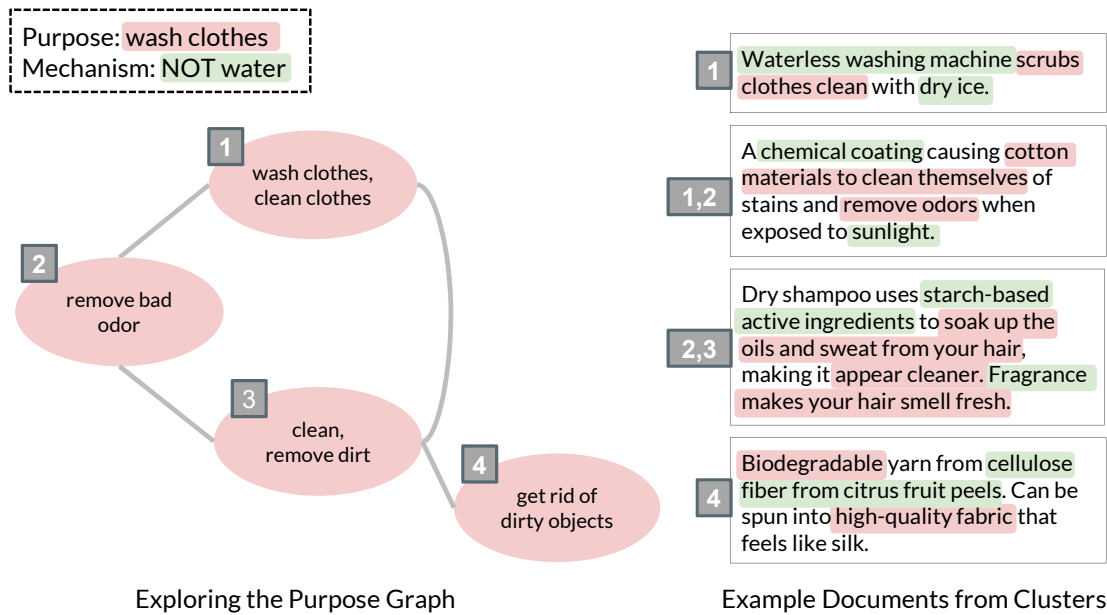
**Figure 1: Extracting fine-grained purposes and mechanisms at scale enables mapping the landscape of ideas. Suppose an inventor is looking for a way to wash clothes without water. On the left we see a snippet from the graph of purposes. Each node in the graph represents a cluster of similar purpose spans extracted from documents (labels are manually generated for illustration purposes). Edges reflect abstract structural similarity, capturing co-occurrence patterns of spans in the corpus (see Section 5.1 and Figure 9). On the right we see example documents containing purposes from the four clusters (corresponding cluster numbers appear in boxes). Purpose/mechanism spans in documents are shown in pink/green, respectively. One could find direct matches to the query – i.e., documents with purpose from cluster 1 and mechanism not "water" (e.g., waterless washing machine using dry ice), or explore neighboring purpose clusters, reformulating the problem as removing odor, removing dirt, or getting rid of the dirty clothes, each resulting in a different set of inspirations.**

bacterial replication [77]; and in computer science and optimization, analogies to processes in nature inspired algorithms such as simulated annealing [59], genetic algorithms [35], and momentum-based gradient descent [83]. This process of abstraction over past ideas is also an important contributor to the ability to reformulate problems [20, 23, 24, 31, 57]. For example, innovators tasked with a problem (find more room to store e-waste) might consider a related, more general goal (reducing environmental pollution) to inspire new solutions, or consider "sibling" formulations (create alternative materials that are biodegradable). This ability to reframe a problem using other problems that bear some abstract relation to it is known to be a powerful way to combat fixation and boost creativity [16, 31, 36–38]. This abstraction and reframing process has also been observed in studies of example curation [49, 56].

A core unifying thread across these mechanisms is the need for particular structured, yet flexible, representations of ideas in terms of their component aspects, such as analogical schemas, or conceptual chunks. In this paper, we focus on developing representations with these properties, starting with the decomposition of ideas into their component purposes and mechanisms. As we discuss in the next section, developing computational representations with these properties that can operate over large scale repositories of ideas remains a formidable open challenge.

**Utilizing Structured Representations for Ideation Systems.**
A main focus of creativity techniques and prototypes has been building computational systems for exploring the space of possible solutions to problems and alternative problem perspectives [31]. To do so, such systems often leverage structured knowledge representations for mapping the design space and linking across different problems. For example, the WordTree method [65] – a prominent design method in creative engineering design – directs designers to break their problem into subfunctions, and then use the WordNet database [75] to explore abstractions and related functional aspects. Likewise, a recent study [42] asked designers to select product aspects to abstract using WordNet and the Cyc ontology [63], which aimed to serve as a general-purpose repository of commonsense knowledge in structured form. These and other general-purpose knowledge bases (e.g., NELL [76] and DBpedia [29]) largely encode categorical knowledge (e.g., is-a, has-a) and rarely functional knowledge (e.g., used-for), and often suffer from poor coverage of concepts in real-world products [42]. Knowledge bases and ontologies that *do* focus on functions, behaviors, and structures [6, 51, 96] have primarily been hand-crafted and are therefore even more limited in coverage. Work attempting to scale up has shown promise but is limited in expressivity or interpretability, such as modeling patents in terms of verbs and nouns [32], using principal component analysis [25], or learning coarse aggregate vectors that capture

| Purpose | Mechanism | Untag |
| --- | --- | --- |

What everyone wants to have is a comfortable way to sleep while traveling. A neck pillow filled with soft material that supports your neck. It's unique because it has sensors to track your sleep.

**Figure 2: Crowdsourcing interface for fine-grained purposes and mechanisms.**

only one overall product purpose and mechanism [52] that cannot disentangle the different aspects of a product, unlike our work presented in this paper. While full abstraction of ideas currently remains a holy grail, here we investigate whether learning nuanced functional aspects might enable a limited form of abstraction useful for augmenting creativity and providing a first step towards true automated abstraction.

## 3 LEARNING A FINE-GRAINED FUNCTIONAL REPRESENTATION

Our goal in this section is to construct a representation that can support the creative innovation tasks and interactions discussed in the Introduction.

We propose to use *span representations* [62]. Given a product text description, we extract tagged spans of text corresponding to purposes and mechanisms (see Figure 2), and represent the product as a set of span embeddings.

More technically, we use a standard *sequence tagging* formulation, with $\mathcal{X}_N = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ a training set of $N$ texts, each a sequence of tokens $\mathbf{x}_i = (x_i^1, x_i^2, \ldots, x_i^T)$, and $\mathcal{Y}_N$ a corresponding set of label sequences, $\mathcal{Y}_N = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N\}$, $\mathbf{y}_i = \{y_i^1, y_i^2, \ldots, y_i^T\}$, where each $y^j$ indicates token $j$'s label (purpose/mechanism/other). In later sections, we represent each product $i$ as a set of purpose span embedding vectors and a set of mechanism span embedding vectors.

### 3.1 Data

We use real-world product idea descriptions taken from crowdsourced innovation website Quirky.com and used in [52], including 8500 user-generated texts describing inventions across diverse domains (e.g., kitchen products, health and fitness, clean energy). Texts typically include multiple purposes and mechanisms. Texts in Quirky use very *nonstandard language*, including grammatical and spelling errors (e.g., "Folds Up Perfect For Carrying. you can walk-on, put your mouth on and or hands on. numbers in any configuration 4 learning to De / Composing Numbers.").

**Annotation.** To create a dataset annotated with purposes and mechanisms, we collect crowdsourced annotations on Amazon Mechanical Turk (AMT). In the similar annotation task of [52] workers were reported to annotate long, often irrelevant spans. We thus guided workers to focus on shorter spans. To further improve quality and encourage more fine-grained annotations, we limited maximal span length that could be annotated, and disabled the
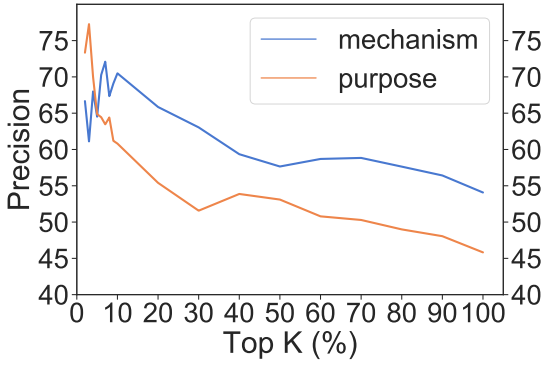
annotation of stopwords. Fig. 2 shows our tagging interface; rectangles are taggable chunks. For quality control, we required US-based workers with approval rate over 95% and at least 1000 approved tasks, and filtered unreasonably fast users. In total, we had 400 annotating workers. Workers were paid $0.1 per task. This rate was computed aiming for an hourly rate of $7, where completion time was estimated via a small-scale pilot study. However, in the full study we were surprised to find the median completion time was much higher, reaching 100 seconds. We note that this figure could be skewed (e.g., due to workers queuing of tasks or the ability to take breaks).

While a manual inspection of the annotations revealed they are mostly satisfactory, we observe two main issues: First, there are often **multiple correct annotations**. Second, workers provide **partial tagging** – in particular, if similar spans appear in different sentences, very few workers bother tagging more than one instance (despite instructions). These issues would have made computing evaluation metrics problematic. We thus decided to use the crowdsourced annotations as a *bronze-standard* for training and development sets only. For a reliable evaluation, we collected *gold-standard* test sets annotated by two CS graduate students. Annotators were instructed to mark *all* the relevant chunks, resulting in high inter-annotator agreement of 0.71. We collect 22316 annotated training sentences and 512 gold sentences, for a total of $238,399 tokens$ (tag proportions: 14.5% mechanism, 15.9% purpose, 69.6% other).

**A note on related annotated data.** There has been recent work on the related topic of information extraction from *scientific papers* by classifying sentences, citations, or phrases. Recent supervised approaches [9, 55, 66] use annotations which are often provided by either paper authors themselves, NLP experts, domain experts, or involve elaborate (multi-round) annotation protocols. Sequence tagging models are often trained and evaluated on (relatively) clean, succinct sentences [71, 101]. When trained on noisy texts, results typically suffer drastically [2]. Our corpus of product descriptions is significantly noisier than scientific papers, and our training annotations were collected in a scalable, low-cost manner by non-experts. Using noisy crowdsourced annotation for training and development only is consistent with our quest for a lightweight annotation approach that would still enable training useful models.

### 3.2 Extracting Spans

After collecting annotations, we can now train models to extract the spans. We explore several models likely to have sufficient power to learn our proposed novel representation, with the goal of selecting the best performing one. In particular, we chose two approaches that are common for related sequence-tagging problems, such as named entity recognition (NER) and part-of-speech (POS) tagging: a common baseline and a recent state-of-the-art model. We also tried a model-enrichment approach with syntactic relational inputs. We briefly describe the models we used below, with full technical descriptions and implementation details, data and code appearing in the supplementary material (Appendix A.1). We note that our goal in this section is to find a *reasonable model* whose output could support creative downstream tasks; many other architectures are possible and could be considered in future work.

| Configuration | P | R | $F_1$ |
|---|---|---|---|
| Enriched BiLSTM | 45.24 | 39.01 | 41.90 |
| Pooled-Flair | 53.30 | 39.80 | 45.50 |
| GCN | 47.85 | 47.93 | **47.89** |
| GCN self-train | 49.00 | 52.00 | **50.50** |

**Figure 3: Left: Precision@K results for the best performing model (GCN + self-training). Right: Raw extraction accuracy evaluation. All approaches use CRF loss. GCN with syntactic edges outperforms baselines. Self-training further improves results. Random-label achieves only** 16.01 $F_1$.

- **BiLSTM-CRF.** A BiLSTM-CRF [53] neural network, a common baseline approach for NER tasks, enriched with semantic and syntactic input embeddings known to often boost performance [101]. We adopt the "multi-channel" strategy as in [101], concatenating input word embeddings (pretrained GloVe vectors [81]) with part-of-speech (POS) and NER embeddings. A conditional random field (CRF) model over the BiLSTM outputs maximizes the tag sequence log likelihood under a pairwise transition model between adjacent tags [5].

- **Pre-trained Language Model (Pooled Flair).** A pre-trained language model [4] based on contextualized string embeddings, recently shown to outperform other powerful models such as BERT [22] in NER and POS tagging tasks and achieve state-of-art results.

- **GCN.** We also explore a model-enrichment approach with syntactic relational inputs. We employ a graph convolutional network (GCN) [58] over dependency-parse edges [101]. GCNs are known to be useful for propagating relational information and utilizing syntactic cues [71, 101]. The linguistic cues are of special relevance and interest to us, as they are known to exist for purpose/mechanism mentions in texts [32].

## 3.3 Evaluation of Extraction Accuracy

In this section we assess *extraction accuracy* (whether we are able to extract purpose and mechanism spans of text). In the next sections, we evaluate the *utility* of the extracted spans for enabling creative innovation tasks.



**Figure 4: Comparing our GCN model predictions (right) to human annotations (left). Interestingly, our model managed to correct some annotator errors ("it's", "heated", "coffee warm", "beverages"). Purposes in pink, mechanisms in green.**

To evaluate raw accuracy of the model's predictions, we use the standard IOB label markup to encode the *purpose* and *mechanism* spans (5 possible labels per token, {Beginning, Inside} x {Purpose, Mechanism} plus an "Outside" label). We conduct experiments using a train/development/test split of 18702/3614/512 .

Due to our challenging setting, we train models on bronze-standard annotations with noisy and partial tagging done by non-experts; for evaluation we use a curated gold-standard test set (Section 3). See Figure 3 (right) for results: GCN reaches an $F_1$ score of ∼ 48%, outperforming the BiLSTM-CRF model (enriched with multi-channel GloVe, POS, NER and dependency relation embeddings) by 6%. GCN also surpasses the strong Pooled-Flair pre-trained language model by nearly 2.5%. A random baseline guessing each token by label frequencies (Section 3) achieves 16.01 $F_1$. We interpret these results as possibly attesting to the utility of graph representations and features capturing syntactic and semantic information when labels are noisy. As a sanity check, we also computed precision@K (Figure 3, left). As expected, precision is higher with low values of $K$, and gradually degrades. Precision for mechanisms is higher than for purposes. Interestingly, a manual inspection revealed many cases where despite the noisy training setting, our models managed to correct mistaken or partial annotations (see Figure 4).

**Self-Training.** According to the results, we chose GCN as our best-performing model. We experimented adding self-training [86] to GCN. Self-training is a common approach in semi-supervised learning where we iteratively re-label "O" tags in training data with model predictions. A large portion of our training sentences are (erroneously) un-annotated by workers, perhaps due to annotation fatigue, introducing bias towards the "O" label.

Self-training with GCN shows an improvement in $F_1$ by an additional 2.6%, substantially increasing recall (more than 12% over Flair), see Figure 3, right. Self-training stopped after 2 iterations, following no gain in $F_1$ on the development set.

In the following two sections we demonstrate that our extraction model's accuracy, while far from perfect, is sufficient for achieving good performance on the *downstream* tasks which are at the focus of this paper. One main reason for this gap is that our downstream tasks involve aggregation of multiple extracted spans: Product

descriptions will typically mention salient mechanisms/purposes several times in the text, such that the effect of local false positives/negatives is mitigated if overall the key aspects are captured somewhere in the text. Further, as we discuss in §5.1, our approach also aggregates purposes and mechanisms across the *entire corpus*, not just single texts, learning from patterns observed sufficiently many times across multiple texts and thus removing noise introduced by extraction errors. As future information extraction technologies advance, our task could benefit from improved extraction accuracy to further reduce the rate of false positives and negatives.

## 4 FINE-GRAINED FUNCTIONAL SEARCH FOR ALTERNATIVE USES

In the previous section we suggested a model for extracting purpose and mechanism spans and assessed extraction accuracy. Our focus in this paper is to study the *utility* of the extracted purposes and mechanisms, in terms of the user interactions they enable. In the following sections we explore two tasks demonstrating the value of our novel representation for supporting creative innovation. We start with a task involving search for *alternative uses*.

**Motivation.** Our task is inspired by one of the most well-known divergent thinking tests [46] for measuring creative ability – the alternative uses test [47], where participants are asked to think of as many uses as possible for some object. Aside from serving as a measure of creativity, the ability to find alternative uses for technologies has important applications in engineering, science and industry. Technologies developed at NASA, the US space agency, have led to over 2,000 spinoffs, finding new uses in computer technology, agriculture, health, transportation, and even consumer products[1]. Procter & Gamble, the multinational consumer goods company, has invested in systematic search for ideas to re-purpose and adapt from other industries, such as using a compound that speeds up wound healing to treat wrinkles - an idea that led to a new line of anti-wrinkle products [27]. And very recently, the COVID-19 pandemic provided a stark example of human innovation, with many companies seeking to pivot and re-purpose existing products to fit the new climate [28].

One teaching story is that of John Osher, creator of the "Spin Pop" — a lollipop with a mechanism for twirling in your mouth. After selling his invention, Osher's team systematically searched for new ideas — "rather than having an idea come to us"[2]. The group eventually landed on the "Spin Brush" – a cheap electric toothbrush adapted from the same twirling mechanisms. This case of repurposing an existing technology involved a systematic search process rather than pure serendipity. Introducing automation could help accelerate the search process by scouring many relevant problems available online, but the task is challenging for existing search systems, requiring a fine-grained, multi-aspect understanding of products.

**Illustrative Example.** Consider a company that manufactures light bulbs. The company is familiar with straightforward usages

of their product (lamps, flashlights), and wants to identify nonstandard uses and expand to new markets. Finding uses for a lightbulb that are not about the standard purpose of illuminating a space would be difficult to do with a standard search query over an idea repository, as the term "lights" or "lighting" will bring back lots of results close to "lamps," "flashlight," and the like. In contrast, with our representation each idea is associated with mechanism and purpose aspects, and one could form a query such as *mechanism="light bulb", purpose= NOT "light"*. Using our system, the searcher adds "light" as a mechanism and also adds "light" as a negative purpose (i.e., results should not include "light" as a purpose). Our prototype returns examples such as billiard laser instructor devices (Table 1), warning signs on food packages to get attention of kids with allergies, and lights attached to furniture to protect your pinky toes at night (Figure 5).

### 4.1 Study Design

We have built a search engine prototype supporting our representation. Figure 5 shows the top two results for the light bulb scenario: warning lights on food for kids with allergies, and lights attached to furniture to protect your pinky toe at night. These are non-standard recombinations [30] (light + allergies, light + furniture guard) that could lead the company to new markets.

We conduct an experiment simulating scenarios where users wish to find novel/uncommon uses of mechanisms. Table 1 shows the scenarios and examples. To choose these scenarios for the experiment, we find popular/common mechanisms in the dataset and their most typical uses. For example, one frequent mechanism is RFID, which is typically used for purposes such as "locating" and "tracking". We then create queries searching for *different* uses – purposes that do not include concepts related to the typical uses of a given mechanism. To automate scenario selection, we cluster mechanisms (see Section 5.1), select frequent mechanisms from the top 5 largest mechanism clusters, and identify purposes strongly co-occurring with them (e.g., "RFID" co-occurs with "locating", "tracking") to avoid.

**Our Approach.** We represent each product $i$ as a set of purpose vectors $\mathcal{P}_i := \{\mathbf{p}_i^1, \mathbf{p}_i^2, \ldots, \mathbf{p}_i^{P_i}\}$, and a set of mechanism vectors $\mathcal{M}_i := \{\mathbf{p}_i^1, \mathbf{p}_i^2, \ldots, \mathbf{p}_i^{M_i}\}$ extracted with our GCN model. Similarly, we define a set of query vectors $\mathbf{q}_p := \mathbf{q}_1, \mathbf{q}_2, \ldots \mathbf{q}_{Q_p}$ and $\mathbf{q}_m := \mathbf{q}_1, \mathbf{q}_2, \ldots \mathbf{q}_{Q_m}$. Each query chunk can be negated, meaning it should not appear. Finally, we define distance metrics $d_p(\cdot, \cdot)$, $d_m(\cdot, \cdot)$ between *sets* of purposes and mechanisms. For example, to locate a dog using RFID but *not* GPS:

$$\operatorname*{argmin}_i \; d_p(\{\mathbf{q}_{\text{"locate dog"}}\}, \mathcal{P}_{\tilde{i}})$$
$$s.t. \; d_m(\{\mathbf{q}_{\text{"GPS"}}\}, \mathcal{M}_{\tilde{i}}) \geq \text{threshold} \qquad (1)$$
$$d_m(\{\mathbf{q}_{\text{"RFID"}}\}, \mathcal{M}_{\tilde{i}}) \leq \text{threshold}$$

We explore two alternatives for computing distance metrics $d_m, d_p$:

- **FineGrained-AVG.** $d_p(\mathbf{q}_p, \mathcal{P}_i)$ is 1 minus the dot product between average query and purpose vectors (normalized to unit norm). We define $d_m$ similarly.

- **FineGrained-MAXMIN.** We match each element in $\mathbf{q}_p$ with its nearest neighbor in $\mathcal{P}_i$, and then find the minimum over the distances between matches. $d_p$ is defined as 1 minus the

| Query | Example results |
|---|---|
| Mechanism: *light*. Purpose: NOT *light* | Billiard laser instructor (projector) |
| Mechanism: *solar energy*. Purpose: NOT *generating power* | Light bulbs with built-in solar chips. |
| Mechanism: *water*. Purpose: NOT *cleaning*, NOT *drinking* | A lighter that burns hydrogen generated from water and sunlight. |
| Mechanism: *RFID*. Purpose: NOT *locating*, NOT *tracking* | A digital lock for your luggage with RFID access. |
| Mechanism: *light*. Purpose: *cleaning* | A UV box to clean and sanitize barbells at the gym. |

Table 1: Scenarios and example results retrieved by our FineGrained-AVG method. Queries reflect non-trivial uses of mechanisms (e.g., using water not for drinking/cleaning, retrieving a lighter running on hydrogen from water and sunlight).



Figure 5: Applications for light where light is not in the purpose. Left: Interface. Right: Two of the results and their automatic annotations (purposes in pink, mechanisms in green).

minimum. All vectors are normalized. We define $d_m$ similarly. This captures cases where queries match only a small subset of product chunks, erring on the side of caution with a max-min approach.

**Baselines.** We test our model against:

- **AvgGloVe.** A weighted average of GloVe vectors of the entire text (excluding stopwords), similar to standard NLP approaches for retrieval and textual similarity. We average query terms and normalize to unit norm. Distance is computed via the dot product.

- **Aggregate purpose/mechanism.** Representing each document with the model in [52]. This model takes raw text as
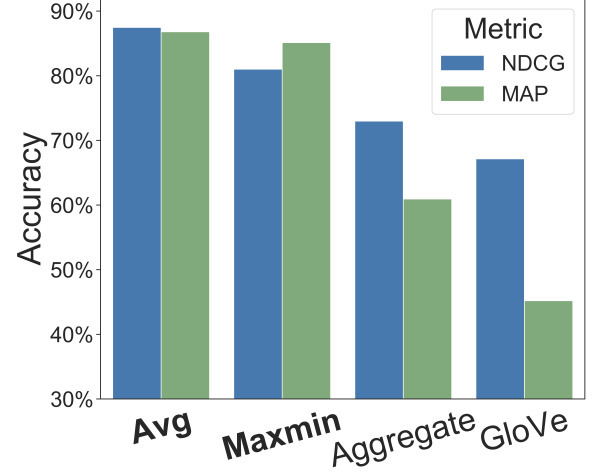


Figure 6: Results for search evaluation test case. Mean average precision (MAP) and Normalized Discounted Cumulative Gain (NDCG) by method, averaged across queries. Methods in bold use our model.

input, applies a BiLSTM neural network and produces two vectors corresponding to *aggregate* purpose and mechanism of the document. We average and normalize query vectors, and use the dot product.

For all four methods, we handle negative (purpose) queries by filtering out all products whose similarity is lower than $\lambda$, where lambda is a threshold selected to be the 90th percentile of similarities (1 minus the distances). This corresponds to the threshold seen in the example in Eq. 1.

### 4.2 Results

We recruited five engineering graduate students (three female, two male) to judge the retrieved product ideas. Each participant provided binary relevance feedback [88] (yes/no) to the top 20 results from each of the four methods, shuffled randomly so that judges are blind to the condition.[3]

See Figure 6 for results. We report Non Cummulative Discounted Gain (NDCG) and Mean Average Precision (MAP), two common metrics in information retrieval [88]. Our FineGrained-AVG wins for both metrics, followed by FineGrained-MAXMIN. The baselines

---

[3]Inter-rater agreement measured across all scenarios was at 50% by both Fleiss kappa and Krippendorff's alpha tests.

perform much worse, with the aggregate-vectors approach in [52] outperforming standard embedding-based retrieval with GloVe. Importantly, our approach achieves high MAP (85% - 87%) in *absolute terms*, in addition to a large relative improvement over the baselines (MAP of 40%-60%).

**Qualitative Analysis.** Table 1 shows example results of FineGrained-AVG. For instance, a query for using light not for lighting results in laser-based billiard instructions. A query for using RFID not for locating or tracking results in an idea for an RFID-based lock, or RFIDs used at supermarket checkouts. To give an intuition for what might be driving our quantitative findings, we examine examples of retrieved results.

For instance, with the query for using light for the non-standard purpose of cleaning, the top ranked result retrieved by FineGrained-AVG is a *UV Light Sterilizer*, with extracted purposes including *Sterilizes bacteria*, *Keep public and people healthy* and *Cleaner fresher air*, and the top result from FineGrained-MAXMIN is similarly a *Standalone bug zapper bulb* that uses *uv light/black light*. Conversely, the top result for both baselines (standard search and aggregate-vectors) is a *Toilet/Bathroom Light*, with "a sensor light that glows around your toilet" and "extra batteries if you lose electricity in the bathroom". It appears that both baselines were not able to accurately capture and disentangle purposes and mechanisms, despite the aggregate-vector being explicitly designed for that.

More generally, it appears that the aggregate-vector approach squashes multiple purposes together by design into one soft, aggregate vector, which in this case includes concepts like *toilet* and *bathroom* that are somewhat topically related to cleaning. The aggregate approach had similar issues in the next product ideas it retrieved (e.g., *Switch that glows in the dark*, a *Dash Light to illuminate ash trays*).

Overall, our results demonstrate that fine-grained purposes and mechanisms lead to better functional search expressivity than approaches based on distributional representations or coarse purpose-mechanism vectors.

## 5 EXPLORING THE DESIGN SPACE WITH A FUNCTIONAL CONCEPT GRAPH

In this section we test the value of our novel representation for supporting users in exploring the design space for solving a given problem. We use our span-based representation to construct a corpus-wide graph of purpose/mechanism concepts. We demonstrate the utility of this approach in an ideation task, helping users identify useful inspirations in the form of problems that are related to their own.

Our goal is to help users "break out" of fixation on a certain domain, a well-known hindrance to innovation [15, 60]. Doing so is challenging because it requires some level of *abstraction*: being able to go beyond the details of a concrete problem to connect to a part of the design space that may look dissimilar on the surface, but has abstract similarity. Numerous studies in engineering and cognitive psychology have shown the benefits of problem abstractions for ideation [32, 34, 45, 60, 64, 98, 99]. However, these studies either involve non-scalable methods (relying on highly-structured annotations, or on crowd-sourcing) or simple, syntactical pattern-matching heuristics incapable of capturing deeper abstract
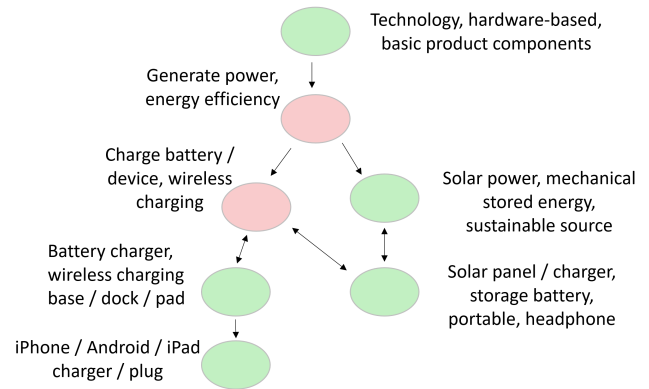


**Figure 7: An example of our learned functional concept graph extracted from texts. Mechanism in green, purpose in pink. Titles are tags nearest to cluster centroids (redacted to fit).**

relations. In [52] (aggregate-vectors baseline from the previous section), crowdworkers were given a product description from the Quirky database, and asked to come up with ideas for products that solve the same problem in a different way. Aggregate vectors representing purpose and mechanism were used to find near-purpose, far-mechanism analogies. Thus, finding analogies relied on having a given mechanism to control for structural distance.

Unlike [52], in our setup we assume a more realistic scenario where we are given only a short problem description – e.g., *generating power for a phone, reminding someone to take medicine, folding laundry* – and aim to find inspirational stimuli [45] in the "sweet spot" for creative ideation – structurally related to the given problem, not too near yet also not too far [33].

**Functional Concept Graph.** To address this challenge, we build a tool inspired by functional modeling [51], which we call a *Functional Concept Graph*. A functional model is, roughly put, a hierarchical ontology of functions and ways to achieve them, and is a key concept in engineering design. Such models are especially useful for innovation, allowing problem-solvers to break out of a fixed overly-concrete purpose or mechanism and move up and down the hierarchy. Despite their great potential, today's functional models are constructed manually, and thus do not scale. While automatically inducing full abstraction hierarchies/ontologies of functional properties of real-world products remains a daunting challenge, in our approach we construct a rough approximation — simple enough to extract automatically from noisy product texts, while still being useful for exploring the design space and suggesting inspirations to users. Specifically, in our approach Functional Concept Graphs consist of nodes corresponding to purposes or mechanisms, and edges reflect semantic relatedness that is not guaranteed to directly encode abstraction. We build this graph by observing fine-grained co-occurrences of concepts appearing together in products, using rule-mining to infer which concept is likely to be more general to (roughly) capture different levels of abstraction.

For example, Figure 7 shows an actual subgraph from our automatically constructed functional concept graph related to electricity, power and charging. Products that mention certain purposes (e.g., "charge your phone") will often mention other, structurally related problems that could be more general/abstract (e.g., "generate power") or more specific ("wireless phone charging"), resulting in edges in our graph (only high-confidence edges are shown). A designer could go from the problem of charging batteries to the more general problem of generating power, and from there to another branch (e.g., solar power and mechanical stored energy), to get inspired by structurally related ideas.

## 5.1 Building a Functional Concept Graph

We develop a method to infer this representation from co-occurrence patterns of the fine-grained spans of text. Naively looking for co-occurrences of problems may yield inspirations too near to the original $p_i$, as many frequently co-occurring purposes tend to be very similar, while we are interested discovering the more abstract relations. In addition, raw chunks of text extracted from our tagging model have countless variants that are not sufficiently abstract and are thus sparsely co-occurring. We thus design our approach to encourage abstract inspirations. As an overview of our approach before presenting the technical details, we take the following two steps:

**I. Concept discretization.** Intuitively, nodes in our graph should correspond to groups of related spans ("charging", "charging the battery", "charging a laptop"). To achieve this, we take all purpose and mechanism spans $\hat{\mathcal{P}}$, $\hat{\mathcal{M}}$ in the corpus, extracted using our GCN model, and cluster them (separately), using pre-trained vector representations. We refer to the clusters $C_p, C_m$ as *concepts*.

**II. Relations.** We employ rule-mining [80] to discover a set of relations $\mathcal{R}$ between concepts (see §5.2 for implementation details). Relations are *Antecedent* $\implies$ *Consequent*, with weights corresponding to rule confidence. To illustrate our intuition, suppose that when "prevent head injury" appears in a product description, the conditional probability of "safety" appearing too is large (but not the other way around). In this case, we can (weakly) infer that preventing head injuries is a sub-purpose of "safety".

Indeed, manually observing the purpose-purpose edges, the one-directional relations captured are often *sub-purpose,* and the bi-directional ones often encode *abstract similarity*. Similarly, for mechanism concepts the one-directional relations are often *part of* ("cell phone" and "battery"), and bi-directional are mechanisms that *co-occur* often. For pairs of purpose and mechanism concepts, the relation is often *functionality* ("charger", "charge"). Exploring more relations is left for future work.

## 5.2 Study Design & Implementation

Next, we set out to test the utility of the functional concept graph in an ideation task. In our setup participants are given problems (e.g., reminding people to take their medication in the morning) and are asked to think of creative solutions. Participants were also given a list of *potential inspirations*, grouped into boxes, and were instructed to mark whether each was novel and helpful. They were also encouraged to explain the solution it inspired.



Figure 8: A snippet from our ideation interface for "morning medicine reminder". Users see inspirations grouped into boxes. Each box is supposed to represent a concept – a cluster of related spans as found by our method or by the baselines (see §5.1). Users indicate which inspirations were useful, and what ideas they inspired. For example, seeing "real time health checker" inspired one user to suggest a monitoring device for finding the best time for reminding to take the medicine.

Figure 8 shows our interface. In this example, seeing inspirations about health monitors caused one user to suggest monitoring the person to find the best time to remind them to take medicine; seeing inspirations about coffee caused them to suggest integrating medicine reminders into coffee machines.

To create a set of **seed problems**, a graduate student mapped between problems from WikiHow.com (a website of how-to guides) to purposes in our data. Using this source allowed us to collect real-world problems that are broadly familiar, with succinct and self-explanatory titles that do not require further reading to understand. The student was tasked with confirming that our Quirky dataset contains idea descriptions that mention these problems. For a given problem in WikiHow (*how to remember to take medication*), they performed keyword search over $17K$ purpose spans gleaned by our model from Quirky, and found matching spans (*morning medicine reminder*). We use those matching spans as our seed problem description given to users (purple text in Figure 8). We collect 25 problems this way. Table 2 shows more examples, such as *Tracking distance walked*, *folding laundry* or *sensing dryness level.*

**Inspirations** are other purpose spans from our dataset (see Table 2), selected automatically using our approach or baseline approaches.

**Our Method.** For our approach, we construct a functional concept graph as in Section 5.1. To cluster related spans into concept nodes, we explore two common and powerful vector representations of spans to capture semantic similarity:

- **GloVe** [81] pre-trained word embeddings, averaged across tokens.
- **BERT-based** [84] contextualized vectors that have been fine-tuned for semantic similarity tasks[4].

---

[4]We use RoBERTa-large-STS-SNLI, available at github.com/UKPLab/sentence-transformers.

We cluster the spans using K-Means++ [5][8]. We then apply the Apriori algorithm[6] to automatically mine association rules between clusters, [80] and use the confidence metric to select the top rules[7]. To use the mined rules between purpose nodes (clusters) for selecting inspirations shown to users, we start from the purpose node corresponding to the given problem and take its *consequents*; as explained earlier, this captures a weak signal of abstract similarity.

Some of these nodes contain tens of spans in them. Thus, we also explore two approaches to "summarize" each concept cluster with representative spans displayed to users – one that attempts to summarize the cluster independently of the seed problem, and one that takes the seed problem into account:

- **TextRank [74].** We construct a graph where nodes are the spans in a cluster and edges represent textual similarity. We run PageRank [78] on this graph, selecting the top $K$ spans to present.
- **Nearest spans.** Following the findings in [33], select the top $K$ spans in $C_p$ that are **nearest** to the query $p_i$. (For both approaches, we use $K = 5$).

**Baselines.**[8]

- **Purpose span similarity.** Given a problem $p_i$, we find the $K = 5$ nearest purpose spans of text in our corpus (out of $17K$ purposes). We experiment with the same two vector representations used by our approach: GloVe and BERT. This method is similar to applying the methodology in [52] to our setting, where in our setting we are given only a problem $p_i$ and no mechanism $m_i$ is available to control for structural distance. While this approach relies on our model for extracting purpose spans, we consider it a baseline to study the added value of our hierarchy.
- **Linguistic abstraction.** We use the WordNet [75] lexical database to extract hypernyms (for each token in $p_i$), in order to capture potential abstractions. WordNet is often used in similar fashion for design-by-analogy studies [42, 45, 64].
- **Random concepts.** Random inspirations are often considered as a baseline in ideation studies since diversity of examples is a known booster for creative ability [52]. For each task, we select a random cluster from $C_p$ and display its TextRank summary.

**Study Participants.** We recruit 10 raters to evaluate the inspirations, via university mailing lists. 8 raters were engineering graduate students, and the remaining two raters included a senior engineering professor and an architect. This cohort is intended to reflect a target user base of people interested in innovation and involved in creative inventive thinking as part of their work.

**Rating Collection.** In our study, each method generated $K = 5$ spans (concept summaries), which are grouped and displayed together in a box (see Figure 8). For each problem a rater views 8 boxes in randomized order, to avoid bias. Raters were instructed to

---

[5] $K = 250$ selected automatically with elbow-based criteria on silhouette scores.
[6] http://www.borgelt.net/pyfim.html.
[7] We use the top 3 rules in our experiment.
[8] We note that all methods and baselines include both single and multi-word spans of text as inspirations, ensuring users are blind to the condition.
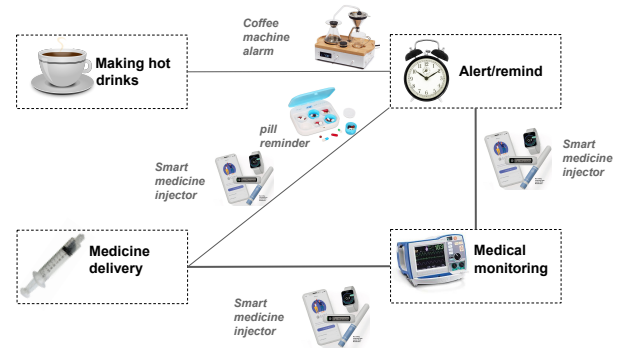


**Figure 9: Excerpt from our Functional Concept Graph. Nodes represent concepts (clusters of purposes). To give intuition for how edges were created, they are annotated with example products containing spans from both concepts. All nodes and edges in this figure were automatically constructed and used to create the user-facing inspirations shown in Figure 8. This figure provides a graphic illustration (not shown to users) explaining how the boxes in Figure 8 are generated, with node names provided here by us for readability. The problem of "medicine morning reminder" is mapped (via embedding) to the *Alert/remind* concept cluster (as named by us), which is linked to the concepts of medical monitoring and making hot drinks through products such as "smart medicine injector" and "coffee machine alarm" (among others, not displayed in the figure). These links serve as the source for inspirations in our study, as seen in Figure 8.**

mark inspirations they consider useful and relevant for solving a given problem, while being *not about the same problem*. Raters were also encouraged to write comments, especially for non-trivial cases which they found of interest (see Table 2). In total, raters viewed 2584 boxes, or 12920 purpose descriptors.

## 5.3 Results

**Analyzing inspirations.** Table 2 and Figure 8 show examples of problems, inspirations and user explanations from our study. For instance, users facing the "morning medicine reminder" problem were presented with nearby concepts in the Functional Concept Graph that included health monitoring and coffee machines. To explore why these concepts are connected in our graph and why they are potentially useful as inspirations, we make use of the direct interpretability of our approach. We examine the purpose co-occurrences from which the Functional Concept Graph was constructed.

Figure 9 shows the subgraph with concept nodes of *Making hot drinks, alerting/reminding, health monitoring, medicine delivery*, and edges representing products in which two adjacent purposes were co-mentioned (e.g., a "coffee machine alarm" product that mentioned the purposes of *making hot drinks* and *alerting/reminding*, or a "smart medicine injector" that mentioned both *alerting/reminding* and *medicine delivery*). This explains why the concepts are nearby

| Problem | Inspirations | Rater explanation |
|---|---|---|
| Track distance walked | Protect children | Get ideas from devices that keep track of children |
| Folding laundry | Store toilet paper | Roll laundry around a tube instead of folding |
| Dispense medicine | Pet bowl that keeps ants away | Based on pet bowls that can dispense food during the day |
| Sense dryness level | Voltage reading | Use electric current to measure water level (safely) |
| | Waterproof | Ideas from sensors in waterproof devices |
| | Temperature reading | |
| Morning medicine reminder | Schedule coffee, coffee alarm | Alarm clock with coffee and medicine reminders |
| | Send vital data, real-time health checker | Health trackers to tell if medicine not taken, alert accordingly |
| | Heart rate monitoring, continuously monitor glucose | Find the best time to take medicine |

Table 2: Example inspirations and explanations given by human evaluators.

in the graph, as there are multiple products in our dataset that share purposes from both concepts.

For example, a "pill reminder" product refers to the problem of forgetting to take medicine at prescribed times (*Sends notification if you forgot to take your AM or PM meds*), while a "smart injector" device administers medicine *on set time intervals*. At the same time, both of these products mention purposes of medicine delivery. When our graph construction algorithm observes enough similar co-occurrence patterns between the concepts of alerting and medicine delivery, across multiple products, an edge is added between the two in the graph. Similarly, an "Alarm coffee maker" product mentions the purposes of *time management* and *making coffee at a set time* as well as *alerting when the coffee is ready*, explaining how it emerges as a potential inspiration in our graph.

This type of linkage or overlap between an original problem space and inspiration problems helps get at a sweet-spot of innovation [16] by finding ideas that are not too near and not too far from the original problem, helping users break out of fixation as discussed earlier in this section. Users in our study used these inspirations to come up with a tracker that alerts the user at the best time to take a medicine and a coffee machine reminding the user to take their medication with their morning coffee. Those creative directions demonstrate the utility of the Functional Concept Graph for exploring the design space.

**Quantitative results.** Figure 10 shows the results of the user study. On the left, we show the proportion of inspirations (individual spans) selected by at least two raters, for each method. Our approach significantly outperforms all the baselines. The effect is particularly pronounced for the BERT-based approach, with 51% of inspirations found useful, while the best baseline reaches less than 30%. Interestingly, for both BERT and GloVe representations, the Nearest-span summarization approach fares better, potentially due to striking a balance between being too far/near the initial problem $p_i$.

Figure 10 (right) shows the proportion of inspiration *boxes* that got at least 2 individual inspirations marked (by at least 2 raters). This metric measures the effect of a box as one unit, as each box is meant to represent a coherent cluster. Our method is able to reach 62%, while the best baseline (GloVe search on purpose spans) yields only 39%. Again, the nearest-span summarization is prefered

to TextRank. Importantly, for both individual inspiration spans and inspirations boxes, 51%-62% are rated as useful – high figures considering the challenging nature of the task.

## 6 DISCUSSION AND CONCLUSION

In this paper we introduced a novel span-based representation of ideas in terms of their fine-grained purposes and mechanisms and used it to develop new tools for creative ideation. We trained a model to extract spans from a noisy, real-world corpus of products. We used this representation to support human creativity in two applications: expressive search for alternative, uncommon uses of products, and generating a graph to help problem-solvers explore the design space around their problem. In both ideation studies, we were able to achieve high accuracies, significantly outperform baselines and *help boost user creativity*.

### 6.1 Limitations

While our results showed the promise of a functional aspect-based representation, and demonstrated potentially feasible technical approaches for extracting this representation from unstructured text, the approach has several limitations.

**Challenging Annotation Task for Crowds.** First, the annotation task proved to be somewhat difficult for crowdworkers, and the outputs were noisy. One direction for future work would be to explore weak supervision approaches to augment annotation. One issue that might exacerbate the problem is that sometimes the boundary between purpose and mechanism is fuzzy, and it is genuinely difficult to tell how to annotate the span.

**Limited Functional Schema.** In a similar vein, it might be interesting to explore more expressive schemas, containing elements other than just purpose and mechanism (similar to [12]). One particularly useful element to add might be context/constraints (e.g., nanoscale), to restrict the search space to feasible solutions.

**Surface Form Abstraction.** Another limitation of our approach is that our functional aspects (and resulting embeddings) remain quite closely anchored to the original texts. This limits their ability to be used to match across domains, to make connections such as inspiring new optimization approaches by analogy to "heating/cooling" schedules in metallurgy. Achieving abstraction to match across
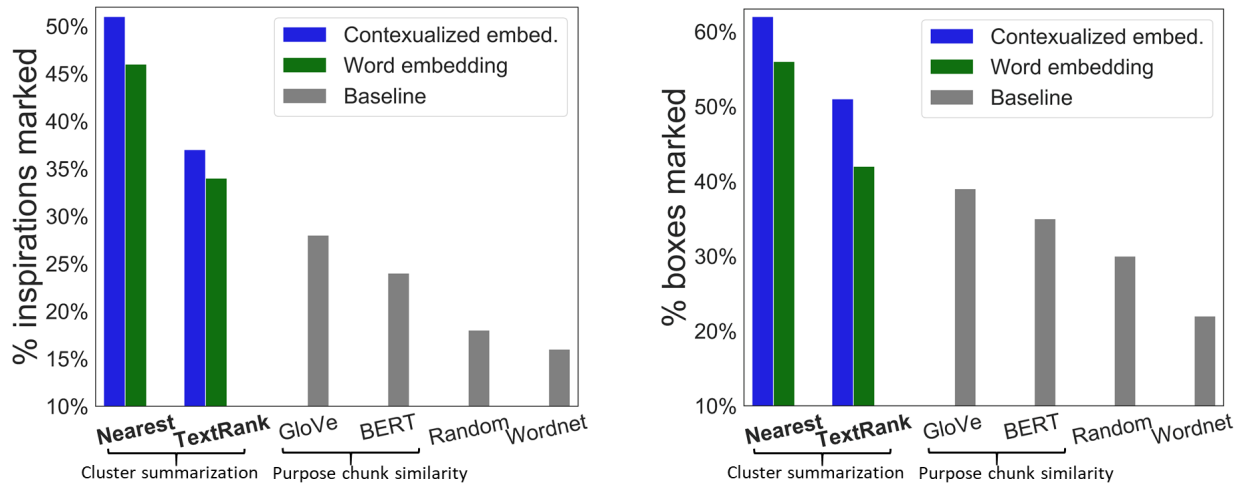
**Figure 10: Inspiration user study results. Left: Proportion of inspirations selected by at least 2 raters, per condition. Right: Proportion of boxes (clusters) with at least 2 spans marked by ≥ 2 raters.**

distant domains without burdening the user with a combinatorial explosion of noisy matches remains an open problem. We wonder if abstracting key objects or entities in a purpose functional aspect — such as a more automated approach to replacing objects with their "commonsense" properties — might be more feasible than attempting to abstract from an entire product description or abstract, given that the chunk is already a rich signal of the product's functional meaning.

## 6.2 Future Work and Broader Implications

Moving to future directions, we are excited about the potential of functional aspects to lead to advances in the interpretability of content-based recommender systems in these complex domains. Keywords are inherently interpretable, but are limited in their capacity to support crossing knowledge boundaries; and until now, embedding-based approaches (e.g., [52]) have not always led to interpretable justifications for matches. Functional aspects could provide the basis of not just more powerful search operators, but also more interpretable results and feedback loops.

**Deeper Functional Graph Exploration.** A key component for the above might be expanding on our use of functional graphs, built from the extracted functional aspects. In our experiments, we used our functional concept graph to retrieve inspirations from "around" the problem. But what would it take to be able to explore this graph? Could we identify and optimize for latent coordinates in the functional space, moving "up" and "down" abstraction levels, or "across" sibling nodes in a functional graph? Taking inspiration from network perspectives on ideation [11, 44, 94], could we retrieve interesting "lineages" of ideas, or compute the potential inspiration value of functional aspects based on network connectivity metrics? Could we combine these content-based functional aspects with measures of use (e.g., citations), to enrich approaches that combine content- and social-based signals, such as literature-based discovery [93, 95]?

**Identifying Overlap and Gaps Across Fields.** These approaches rely on identifying interesting overlaps in concepts that simultaneously coincide with disjunctions in citations, as signals of potentially impactful "undiscovered public knowledge": a persistent challenge is how to define "concepts" - keyword or unstructured text approaches can lead to combinatorial explosions of noise to sift through, and controlled vocabulary (e.g., MEDLine) can help increase the signal to noise ratio, but are only available in specialized circumstances [89]. Functional aspects might be a useful bridge between unstructured text and controlled vocabularies for identifying points of overlap and disjunction between different fields, accelerating the discovery of gems hidden in plain sight.

**Functional aspects for Collaborative Ideation.** Future work could also explore new interactions in collaborative and crowd innovation that might be enabled by the ability to quickly extract functional aspects in idea corpora. The source of our primary dataset here, Quirky, was actually a crowd innovation platform. HCI research on these platforms have begun to emphasize moving away from mere "selection" of best ideas from large samples of ideas, towards supporting generative collaboration over ideas . Open problems include synthesizing major themes in large-scale corpora of user-generated ideas and identifying gaps in the idea space [13, 68, 90], as well as supporting intelligent matching and structuring ways for crowd innovators to collaborate and build on each others' ideas [14, 69] between crowd innovators. We are excited about the potential for functional aspects to assist with these functions, as a complement to other approaches like crowd-powered synthesis [7, 18, 43, 91]. Here, too, the potential for functional aspects to be highly interpretable could power novel explorations of mixed-initiative systems for augmenting collaborative ideation at scale [67, 91].

**Mapping of Design Spaces.** Beyond supporting richer search for creative inspiration, a data-driven approach to extracting functional aspects and learning relationships between the aspects could power much more expansive approaches to mapping out design spaces

for entire domains or problem areas, identifying key subproblems and constraints and novel paths through the design space. Mapping approaches like this, such as technological roadmapping [10], have already shown significant promise for reinvigorating research and development in real-world applications such as neural recording [70]. However, these mapping exercises are still highly manual and labor-intensive processes; computational support for such tasks could have transformative impacts on innovation.

## ACKNOWLEDGMENTS

## REFERENCES

[1] 2013. The car mechanic who uncorked a childbirth revolution. *BBC News* (2013).
[2] Gustavo Aguilar, Suraj Maharjan, Adrian Pastor López Monroy, and Thamar Solorio. 2017. A multi-task approach for named entity recognition in social media data. In *3rd Workshop on Noisy User-generated Text*.
[3] A. Akbik, T. Bergmann, D. Blythe, K. Rasul, S. Schweter, and R. Vollgraf. 2019. FLAIR: An Easy-to-Use Framework for State-of-the-Art NLP. In *NAACL-HLT*.
[4] A. Akbik, T. Bergmann, and R. Vollgraf. 2019. Pooled contextualized embeddings for named entity recognition. In *NAACL*.
[5] A. Akbik, D. Blythe, and R. Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *International Conference on Computational Linguistics*.
[6] Genrich Altshuller. 2002. *40 principles: TRIZ keys to innovation*.
[7] Paul André, Aniket Kittur, and Steven P. Dow. 2014. Crowd Synthesis: Extracting Categories and Clusters from Complex Data. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW '14)*. ACM, New York, NY, USA, 989–998. https://doi.org/10.1145/2531602.2531653
[8] David Arthur and Sergei Vassilvitskii. 2007. k-means++: The advantages of careful seeding. In *ACM-SIAM symposium on Discrete algorithms*.
[9] Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. 2017. Semeval 2017 task 10: Scienceie-extracting keyphrases and relations from scientific publications. *arXiv preprint arXiv:1704.02853* (2017).
[10] Edward S. Boyden and Adam H. Marblestone. 2019. Architecting Discovery: A Model for How Engineers Can Help Invent Tools for Neuroscience. *Neuron* 102, 3 (May 2019), 523–525. https://doi.org/10.1016/j.neuron.2019.03.023 Publisher: Elsevier.
[11] H. Cai, E. Y. Do, and C. M. Zimring. 2010. Extended linkography and distance graph in design evaluation: An empirical study of the dual effects of inspiration sources in creative design. *Design Studies* 31, 2 (2010), 146–168.
[12] J. Chan, J. Chang, T. Hope, D. Shahaf, and A. Kittur. 2018. SOLVENT: A Mixed Initiative System for Finding Analogies between Research Papers. *CSCW* (2018).
[13] Joel Chan, Steven Dang, and Steven P. Dow. 2016. Comparing Different Sensemaking Approaches for Large-Scale Ideation. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 2717–2728. http://joelchan.me/files/2016-chi-sensemaking-ideation.pdf
[14] Joel Chan, Steven Dang, and Steven P. Dow. 2016. Improving Crowd Innovation with Expert Facilitation. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing (CSCW '16)*. ACM, New York, NY, USA, 1223–1235. https://doi.org/10.1145/2818048.2820023
[15] Joel Chan, Steven P. Dow, and Christian D. Schunn. 2015. Do The Best Design Ideas (Really) Come From Conceptually Distant Sources Of Inspiration? *Design Studies* (2015).
[16] Joel Chan, Katherine Fu, Christian Schunn, Jonathan Cagan, Kristin Wood, and Kenneth Kotovsky. 2011. On the benefits and pitfalls of analogies for innovative design: Ideation performance based on analogical distance, commonness, and modality of examples. *Journal of mechanical design* (2011).
[17] J. Chan, T. Hope, D. Shahaf, and A. Kittur. [n.d.]. Scaling up Analogy with Crowdsourcing and Machine Learning. In *ICCBR-16*.
[18] Lydia B. Chilton, Greg Little, Darren Edge, Daniel S. Weld, and James A. Landay. 2013. Cascade: Crowdsourcing taxonomy creation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1999–2008. https://doi.org/10.1145/2470654.2466265
[19] Darren W Dahl and Page Moreau. 2002. The influence and value of analogical thinking during new product ideation. *Journal of Marketing Research* (2002).
[20] Edward De Bono. [n.d.]. *Lateral thinking*.
[21] Edward De Bono. 2010. *Lateral thinking: a textbook of creativity*. Penguin UK. 01064.
[22] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*.
[23] Kees Dorst. 2011. The core of "design thinking" and its application. *Design Studies* (2011).
[24] Hugh Dubberly and Shelley Evenson. 2008. On Modeling: The Analysis-synthesis Bridge Model. *interactions* (2008).
[25] Joost R. Duflou and Paul-Armand Verhaegen. 2011. Systematic innovation through patent based product aspect analysis. *CIRP Annals - Manufacturing Technology* (2011).
[26] C. Eckert and M. Stacey. 2003. Adaptation of Sources of Inspiration in Knitwear Design. *Creativity Research Journal* 15, 4 (2003), 355–384. 00000.
[27] Kristi Essick. 2006. Technology scouts: hoping to find the next big thing. *Science Business* (Feb. 2006). https://sciencebusiness.net/news/73953/Technology-scouts:-hoping-to-find-the-next-big-thing
[28] Kristi Essick. 2020. Innovation and creativity in a time of crisis. *Science Business* (2020). https://globalambition.ie/innovation-and-creativity-to-combat-covid-19/
[29] Michael Färber, Frederic Bartscherer, Carsten Menne, and Achim Rettinger. 2018. Linked data quality of dbpedia, freebase, opencyc, wikidata, and yago. *Semantic Web* (2018).
[30] Lee Fleming. 2001. Recombinant uncertainty in technological search. *Management science* 47, 1 (2001), 117–132.
[31] Jonas Frich, Lindsay MacDonald Vermeulen, Christian Remy, Michael Mose Biskjaer, and Peter Dalsgaard. 2019. Mapping the landscape of creativity support tools in HCI. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–18.
[32] K. Fu, J. Cagan, K. Kotovsky, and K. L. Wood. 2013. Discovering Structure In Design Databases Through Functional And Surface Based Mapping. *JMD* (2013).
[33] Katherine Fu, Joel Chan, Jonathan Cagan, Kenneth Kotovsky, Christian Schunn, and Kristin Wood. 2013. The Meaning of Near and Far: The Impact of Structuring Design Databases and the Effect of Distance of Analogy on Design Output. *JMD* (2013).
[34] Katherine Fu, Joel Chan, Christian Schunn, Jonathan Cagan, and Kenneth Kotovsky. 2013. Expert representation of design repository space: A comparison to and validation of algorithmic output. *Design Studies* (2013).
[35] Mitsuo Gen and Lin Lin. 2007. Genetic Algorithms. *Wiley Encyclopedia of Computer Science and Engineering* (2007).
[36] Dedre Gentner, Sarah Brem, Ron Ferguson, Philip Wolff, Arthur B Markman, and KD Forbus. 1997. Analogy and creativity in the works of Johannes Kepler. *Creative thought: An investigation of conceptual structures and processes* (1997).
[37] Dedre Gentner and Kenneth J. Kurtz. 2005. Relational Categories. In *Categorization inside and outside the laboratory: Essays in honor of Douglas L. Medin*. American Psychological Association, Washington, DC, US.
[38] Dedre Gentner and Arthur B Markman. 1997. Structure mapping in analogy and similarity. *American psychologist* (1997).
[39] Kilian Gericke and Boris Eisenbart. 2017. The integrated function modeling framework and its relation to function structures. *AI EDAM* (2017).
[40] Mary L Gick and Keith J Holyoak. 1980. Analogical problem solving. *Cognitive psychology* 12, 3 (1980), 306–355.
[41] M. L. Gick and K. J. Holyoak. 1983. Schema induction and analogical transfer. *Cognitive Psychology* 15, 1 (1983).
[42] Karni Gilon, Joel Chan, Felicia Y. Ng, Hila Lifshitz-Assaf, Aniket Kittur, and Dafna Shahaf. 2018. Analogy Mining for Specific Design Needs *(CHI '18)*. ACM, Article 121, 11 pages. https://doi.org/10.1145/3173574.3173695
[43] Victor Girotto, Erin Walker, and Winslow Burleson. 2017. The Effect of Peripheral Micro-tasks on Crowd Ideation. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 1843–1854. https://doi.org/10.1145/3025453.3025464
[44] Milene Gonçalves and Philip Cash. 2021. The life cycle of creative ideas: Towards a dual-process theory of ideation. *Design Studies* 72 (Jan. 2021), 100988. https://doi.org/10.1016/j.destud.2020.100988 00000.
[45] Kosa Goucher-Lambert and Jonathan Cagan. 2019. Crowdsourcing inspiration: Using crowd generated inspirational stimuli to support designer ideation. *Design Studies* (2019).
[46] Joy Paul Guilford. 1959. Three faces of intellect. *American psychologist* (1959).
[47] Joy Paul Guilford. 1967. The nature of human intelligence. (1967).
[48] Andrew Hargadon and Robert I. Sutton. 1997. Technology Brokering and Innovation in a Product Development Firm. *Administrative Science Quarterly* 42, 4 (1997), 716–749. https://doi.org/10.2307/2393655 03534 Publisher: [Sage Publications, Inc., Johnson Graduate School of Management, Cornell University].
[49] Scarlett R. Herring, Chia-Chen Chang, Jesse Krantzler, and Brian P. Bailey. 2009. Getting inspired!: understanding how and why examples are used in creative design practice. In *Proceedings of the 27th international conference on Human factors in computing systems*. ACM, 87–96. https://doi.org/10.1145/1518701.

1518717

[50] Mary B Hesse. 1966. Models and analogies in science. (1966).

[51] J. Hirtz, R. Stone, D. A McAdams, S. Szykman, and K. Wood. 2002. A functional basis for engineering design: reconciling and evolving previous efforts. *Research in engineering Design* (2002).

[52] Tom Hope, J. Chan, A. Kittur, and D. Shahaf. 2017. Accelerating innovation through analogy mining. In *KDD*.

[53] Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991* (2015).

[54] David G. Jansson and Steven M. Smith. 1991. Design fixation. *Design Studies* 12, 1 (1991), 3–11.

[55] Di Jin and Peter Szolovits. 2018. Hierarchical Neural Networks for Sequential Sentence Classification in Medical Scientific Abstracts. *arXiv preprint arXiv:1808.06161* (2018).

[56] Andruid Kerne, Nic Lupfer, Rhema Linder, Yin Qu, Alyssa Valdez, Ajit Jain, Kade Keith, Matthew Carrasco, Jorge Vanegas, and Andrew Billingsley. 2017. Strategies of Free-Form Web Curation: Processes of Creative Engagement with Prior Work. In *Proceedings of the 2017 ACM SIGCHI Conference on Creativity and Cognition (C&C '17)*. ACM, New York, NY, USA, 380–392. https://doi.org/10.1145/3059454.3059471

[57] Andruid Kerne, Andrew M Webb, Steven M Smith, Rhema Linder, Nic Lupfer, Yin Qu, Jon Moeller, and Sashikanth Damaraju. 2014. Using metrics of curation to evaluate information-based ideation. *ACM Transactions on Computer-Human Interaction (ToCHI)* 21, 3 (2014), 1–48.

[58] Thomas N. Kipf and Max Welling. 2016. Semi-Supervised Classification with Graph Convolutional Networks. (sep 2016).

[59] Scott Kirkpatrick, C Daniel Gelatt, and Mario P Vecchi. 1983. Optimization by simulated annealing. *science* (1983).

[60] 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. *PNAS* (2019).

[61] G. Knoblich, S. Ohlsson, H. Haider, and D. Rhenius. 1999. Constraint relaxation and chunk decomposition in insight problem solving. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 25, 6 (1999), 1534–1555. 00691.

[62] Tatsuki Kuribayashi, Hiroki Ouchi, Naoya Inoue, Paul Reisert, Toshinori Miyoshi, Jun Suzuki, and Kentaro Inui. 2019. An Empirical Study of Span Representations in Argumentation Structure Parsing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 4691–4698. https://doi.org/10.18653/v1/P19-1464

[63] Douglas B. Lenat. 1995. CYC: a large-scale investment in knowledge infrastructure. In *Communications of the ACM*.

[64] JS Linsey, AB Markman, and KL Wood. 2012. Design by analogy: a study of the WordTree method for problem re-representation. *JMD* (2012).

[65] J. Linsey, A. B. Markman, and K. L. Wood. 2008. WordTrees: A method for design-by-analogy. In *Proceedings of the 2008 ASEE Annual Conference*.

[66] Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-Task Identification of Entities, Relations, and Coreferencefor Scientific Knowledge Graph Construction. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*.

[67] Maximilian Mackeprang, Claudia Müller-Birn, and Maximilian Timo Stauss. 2019. Discovering the Sweet Spot of Human-Computer Configurations: A Case Study in Information Extraction. *Proc. ACM Hum.-Comput. Interact.* 3, CSCW (Nov. 2019), 195:1–195:30. https://doi.org/10.1145/3359297

[68] Narges Mahyar, Diana V. Nguyen, Maggie Chan, Jiayi Zheng, and Steven P. Dow. 2019. The Civic Data Deluge: Understanding the Challenges of Analyzing Large-Scale Community Input. In *Proceedings of the 2019 on Designing Interactive Systems Conference (DIS '19)*. Association for Computing Machinery, New York, NY, USA, 1171–1181. https://doi.org/10.1145/3322276.3322354 00012.

[69] Thomas W. Malone, Jeffrey V. Nickerson, Robert J. Laubacher, Laur Hesse Fisher, Patrick de Boer, Yue Han, and W. Ben Towne. 2017. Putting the Pieces Back Together Again: Contest Webs for Large-Scale Problem Solving. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing (CSCW '17)*. ACM, New York, NY, USA, 1661–1674. https://doi.org/10.1145/2998181.2998343

[70] Adam H. Marblestone, Bradley M. Zamft, Yael G. Maguire, Mikhail G. Shapiro, Thaddeus R. Cybulski, Joshua I. Glaser, Dario Amodei, P. Benjamin Stranges, Reza Kalhor, David A. Dalrymple, Dongjin Seo, Elad Alon, Michel M. Maharbiz, Jose M. Carmena, Jan M. Rabaey, Edward S. Boyden, George M. Church, and Konrad P. Kording. 2013. Physical Principles for Scalable Neural Recording. *Frontiers in Computational Neuroscience* 7 (2013). https://doi.org/10.3389/fncom.2013.00137 arXiv: 1306.5709.

[71] Diego Marcheggiani and Ivan Titov. 2017. Encoding Sentences with Graph Convolutional Networks for Semantic Role Labeling. 1 (2017). arXiv:1703.04826

[72] Arthur B Markman and Jeffrey Loewenstein. 2010. Structural comparison and consumer choice. *Journal of Consumer Psychology* (2010).

[73] T. McCaffrey. 2012. Innovation Relies on the Obscure: A Key to Overcoming the Classic Problem of Functional Fixedness. *Psychological Science* 23, 3 (2012), 215–218. 00117.

[74] Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *EMNLP*.

[75] George A Miller. 1995. WordNet: a lexical database for English. *Commun. ACM* (1995).

[76] Tom Mitchell, William Cohen, Estevam Hruschka, Partha Talukdar, Bo Yang, Justin Betteridge, Andrew Carlson, B Dalvi, Matt Gardner, Bryan Kisiel, et al. 2018. Never-ending learning. *Commun. ACM* (2018).

[77] Andrew Murray. 2016. Salvador Luria and Max Delbrück on random mutation and fluctuation tests. *Genetics* (2016).

[78] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The PageRank citation ranking: Bringing order to the web*. Technical Report. Stanford InfoLab.

[79] Colin Pask. 2003. Mathematics and the science of analogies. *American Journal of Physics* (2003).

[80] Nicolas Pasquier, Yves Bastide, Rafik Taouil, and Lotfi Lakhal. 1999. Discovering frequent closed itemsets for association rules. In *Database Theory—ICDT'99*.

[81] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global Vectors for Word Representation. In *EMNLP*.

[82] Matthew E. Peters, Sebastian Ruder, and Noah A. Smith. 2019. To Tune or Not to Tune? Adapting Pretrained Representations to Diverse Tasks. In *RepL4NLP@ACL*.

[83] Ning Qian. 1999. On the momentum term in gradient descent learning algorithms. *Neural networks* (1999).

[84] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *EMNLP*.

[85] Mark A. Runco and Selcuk Acar. 2019. Divergent thinking. In *The Cambridge handbook of creativity, 2nd ed.* Cambridge University Press, New York, NY, US, 224–254. https://doi.org/10.1017/9781316979839.013 00878.

[86] Mrinmaya Sachan and Eric Xing. 2018. Self-training for jointly learning to ask and answer questions. In *NAACL-HLT*.

[87] R. Keith Sawyer. 2012. *Explaining creativity: the science of human innovation* (2nd ed.). Oxford University Press, New York.

[88] Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. 2008. *Introduction to information retrieval*. Vol. 39. Cambridge University Press Cambridge.

[89] Yakub Sebastian, Eu-Gene Siew, and Sylvester O. Orimaye. 2017. Emerging approaches in literature-based discovery: techniques and performance review. *The Knowledge Engineering Review* 32 (Jan. 2017). https://doi.org/10.1017/S0269888917000042

[90] Pao Siangliulue, Kenneth C. Arnold, Krzysztof Z. Gajos, and Steven P. Dow. 2015. Toward Collaborative Ideation at Scale: Leveraging Ideas from Others to Generate More Creative and Diverse Ideas. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW '15)*. ACM, New York, NY, USA, 937–945. https://doi.org/10.1145/2675133.2675239

[91] Pao Siangliulue, Joel Chan, Steven P. Dow, and Krzysztof Z. Gajos. 2016. IdeaHound: Improving Large-scale Collaborative Ideation with Crowd-powered Real-time Semantic Modeling. In *UIST'16*.

[92] Ut Na Sio, Kenneth Kotovsky, and Jonathan Cagan. 2015. Fixation or inspiration? A meta-analytic review of the role of examples on design processes. *Design Studies* 39 (July 2015), 70–99. https://doi.org/10.1016/j.destud.2015.04.004

[93] Neil R. Smalheiser. 2017. Rediscovering Don Swanson:The Past, Present and Future of Literature-based Discovery. *Journal of Data and Information Science* 2, 4 (Dec. 2017), 43–64. https://doi.org/10.1515/jdis-2017-0019

[94] Ricardo Sosa. 2019. Accretion theory of ideation: evaluation regimes for ideation stages. *Design Science* 5 (2019), e23. https://doi.org/10.1017/dsj.2019.22

[95] Don R Swanson and Neil R Smalheiser. 1996. Undiscovered Public Knowledge: a Ten-Year Update. In *Proceedings of KDD '96*. 4.

[96] Swaroop Vattam, Bryan Wiltgen, Michael Helms, Ashok K. Goel, and Jeannette Yen. 2011. DANE: Fostering Creativity in and through Biologically Inspired Design. In *Design Creativity 2010*.

[97] T. B. Ward. 1995. What's old about new ideas. In *The creative cognition approach*. 157–178.

[98] Lixiu Yu, Aniket Kittur, and Robert E Kraut. 2014. Searching for analogical ideas with crowds. In *CHI*.

[99] L Yu, B Kraut, and A Kittur. 2014. Distributed analogical idea generation: innovating with crowds. In *CHI'14*.

[100] Lixiu Yu, Robert E Kraut, and Aniket Kittur. 2016. Distributed Analogical Idea Generation with Multiple Constraints. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*. ACM.

[101] Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. Graph Convolution over Pruned Dependency Trees Improves Relation Extraction. In *EMNLP*.

# A  TECHNICAL APPENDIX

## A.1  Model Details

- **BiLSTM-CRF.** A BiLSTM-CRF [53] neural network, a common baseline approach for NER tasks, enriched with semantic and syntactic input embeddings known to often boost performance [101]. We first pass the input sentence $\mathbf{x} = (x_1, x_2, \ldots, x_T)$ through an embedding module resulting in $\mathbf{v}_{1:T}$, $\mathbf{v}_i \in \mathbb{R}^{d_e}$, where $d_e$ is the embedded space dimension. We adopt the "multi-channel" strategy as in [101], concatenating input word embeddings (pretrained GloVe vectors [81]) with part-of-speech (POS) and NER embeddings. We additionally add an embedding corresponding to the incoming dependency relation. The sequence of token embeddings is then processed with a BiLSTM layer to obtain contextualized word representations $\mathbf{h}_{1:T}^{(0)}$, $\mathbf{h}_i \in \mathbb{R}^{d_h}$, where $d_h$ is the hidden state dimension. The outputs are fed into a linear layer $f$ to obtain per-word tag scores $f\left(\mathbf{h}_1^{(L)}\right), f\left(\mathbf{h}_2^{(L)}\right), \ldots, f\left(\mathbf{h}_T^{(L)}\right)$. These are used as inputs to a conditional random field (CRF) model which maximizes the tag sequence log likelihood under a pairwise transition model between adjacent tags [5].

- **Pooled Flair.** A pre-trained language model [4] based on contextualized string embeddings, recently shown to outperform powerful approaches such as BERT [22] in NER and POS tagging tasks and achieve state-of-art results. Flair [9] uses a character-based language model pre-trained over large corpora, combined with a memory mechanism that dynamically aggregates embeddings of each unique string encountered during training and a pooling operation to distill a global word representation. We follow [4] and concatenate pre-trained GloVe vectors to token embeddings, add a CRF decoder, and freeze the language-model weights rather than fine-tune them [22, 82].

- **GCN.** We also explore a model-enrichment approach with syntactic relational inputs. We employ a graph convolutional network (GCN) [58] over dependency-parse edges [101]. GCNs are known to be useful for propagating relational information and utilizing syntactic cues [71, 101]. The linguistic cues are of special relevance and interest to us, as they are known to exist for purpose/mechanism mentions in texts [32].

  We used a GCN with same token embeddings as in the BiLSTM-CRF baseline, with a BiLSTM layer for sequential context and a CRF decoder. For the graph fed into the GCN, we use a pre-computed syntactic edges with dependency parsing: For sentence $\mathbf{x}_{1:T}$, we convert its dependency tree to $\mathbf{A}^{syn}$ where $\mathbf{A}^{syn}_{ij} = 1$ for any two tokens $x_i, x_j$ connected by a dependency edge. We also add self-loops $\mathbf{A}^{self} = I$ (to propagate from $\mathbf{h}_i^{(l-1)}$ to $\mathbf{h}_i^{(l)}$ [101]). Following [101], we normalize activations to reduce bias toward high-degree nodes. For an $L$-layer GCN, denoting $\mathbf{h}_i^{(l)} \in \mathbb{R}^{d_h}$ to be the $l$-th layer
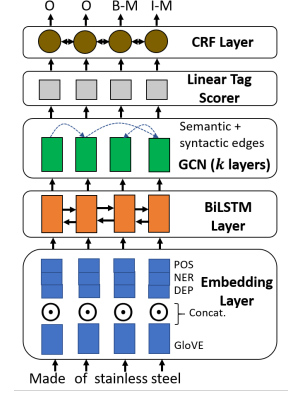


**Figure 11: Schema of our GCN model.**

output node, the GCN operation can be written as

$$h_i^{(l)} = \sigma\left(\sum_{r \in \mathcal{R}}\left[\sum_{j=1}^{n} \mathbf{A}_{ij}^r \mathbf{W}_r^{(l)} h_j^{(l-1)} / d_i^r + \mathbf{b}_r^{(l)}\right]\right)$$

where $\mathcal{R} = \{\text{syn}, \text{self}\}$, $\sigma$ is the ReLU activation function, $\mathbf{W}_r^{(l)}$ is a linear transformation, $\mathbf{b}_r^{(l)}$ is a bias term and $d_i^r = \sum_{j=1}^{T} \mathbf{A}_{ij}^r$ is the degree of token $i$ w.r.t $r$. In the GCN architecture, $L$ layers correspond to propagating information across $L$-order neighborhoods. We set the contextualized word vectors $\mathbf{h}_{1:T}^{(0)}$ to be the input to the GCN, and use $\mathbf{h}_{1:T}^{(L)}$ as the output word representations. Figure 11 illustrates the GCN model architecture. Similarly to [71], we do not model edge directions or dependency types in the GCN layers, to avoid over-parameterization in our data-scarce setting. We also attempted **edge-wise gating** [71] to mitigate noise propagation but did not see improvements, similarly to [101].

In our experiments, we followed standard GCN training procedures. Specifically, we base our model on the experimental setup detailed in [101] (see also the authors' code which we adapt for our architecture, at https://github.com/qipeng/gcn-over-pruned-trees). We pre-process the data using the spaCy (https://spacy.io) package for tokenization, dependency parsing, and POS/NER-tagging. We use pretrained GloVE embeddings of dimension 300, and NER, POS and dependency relation embeddings of size 30 each, giving a total embedding dimension $d_e = 390$. The bi-directional LSTM and GCN layers' hidden dimension is $d_h = 200$, with 1 hidden layer for the LSTM. We find that the setting of 2 hidden layers works best for the GCNs. We also tried training with edge label information based on syntactic relations, but found this hurts performance. The training itself was carried out using SGD with gradient clipping (cutoff 5) for 100 epochs, selecting the best model on the development set.

For the Pooled-Flair approach [4], we use the FLAIR framework [3], with the settings obtaining SOTA results for CONLL-2003 as in [4] (see https://github.com/flairNLP/flair/blob/master/resources/docs/EXPERIMENTS.md). We also experiment with non-pooled embeddings and obtain similar results. We experiment with initial learning rate and batch size settings described in [4], finding 0.1 and 32 to work best, respectively.

---

[9]https://github.com/flairNLP/flair