RecipeScape: An Interactive Tool for Analyzing Cooking Instructions at Scale

Minsuk Chang[1], **Léonore V. Guillain**[2], **Hyeungshik Jung**[1], **Vivian M. Hare**[3], **Juho Kim**[1], **Maneesh Agrawala**[3]

[1]School of Computing, KAIST, {minsuk, hyeungshik.jung, juhokim}@kaist.ac.kr
[2]Department of Communication Systems, EPFL, leonore.guillain@epfl.ch
[3]Computer Science, Stanford University, {vhare, maneesh}@cs.stanford.edu

ABSTRACT

For cooking professionals and culinary students, understanding cooking instructions is an essential yet demanding task. Common tasks include categorizing different approaches to cooking a dish and identifying usage patterns of particular ingredients or cooking methods, all of which require extensive browsing and comparison of multiple recipes. However, no existing system provides support for such in-depth and at-scale analysis. We present RecipeScape, an interactive system for browsing and analyzing the hundreds of recipes of a single dish available online. We also introduce a computational pipeline that extracts cooking processes from recipe text and calculates a procedural similarity between them. To evaluate how RecipeScape supports culinary analysis at scale, we conducted a user study with cooking professionals and culinary students with 500 recipes for two different dishes. Results show that RecipeScape clusters recipes into distinct approaches, and captures notable usage patterns of ingredients and cooking actions.

Author Keywords

Interactive Data Mining; Analysis at Scale; Culinary Analysis; Cooking Recipes; Naturally Crowdsourced Data

ACM Classification Keywords

H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

INTRODUCTION

Cooking recipes provide ingredients and step by step instructions for making a dish, and thousands of recipes are available even for a single dish on the Internet. For example, searching for chocolate chip cookie recipes on Yummly[1] yields 40,000 recipes that span different sets of ingredients, required skills and tools, levels of detail, and even varying arrangements of commonly used cooking actions and ingredients for the dish. These recipes are naturally crowdsourced instructions for a shared goal. Their variety and scale present an opportunity to understand usage patterns of cooking actions and ingredients for different approaches to cooking a dish.

Imagine a chef who wants to be creative with chocolate chip cookies to develop a new dessert menu. The chef has many options to consider, for example, baking unique looking cookies, making a pie using cookies as the crust, or using a specific type of dough that doesn't require baking. Where should the chef start to research the different ways to make or make use of chocolate chip cookies? Imagine a culinary student who is asked to cook a classic tomato pasta and an exotic tomato pasta for an assignment. What is the difference between the set of recipes titled "classic" versus those titled "exotic"? While thousands of recipes for a single dish are available online, it's difficult to browse, compare, and analyze them for coming up with new ideas or interpreting different cooking processes and their results.

For cooking professionals and culinary students, discovering usage patterns of cooking actions and ingredients to understand their implications is just as important as preparing a delicious meal. From our interviews with 10 cooking professionals, we learned that to mine and understand diverse cooking processes, they compare and analyze recipes in three different levels of granularity; *groups of recipes, individual recipes, and individual cooking actions or ingredients.* From a professional chef's menu research activities to training in culinary schools, a wide range of cooking practices emphasize reinterpretation of dishes. Common approaches include applying unusual cooking actions to usual ingredients, applying usual cooking actions to unusual ingredients, or both. These tasks require grouping recipes into similar operational patterns of cooking actions and ingredients, in-depth investigation of individual recipes, and browsing and comparison of individual cooking actions or ingredients.

In this paper, we present RecipeScape (Figure 1), an interactive tool for analyzing hundreds of recipes for a single dish. RecipeScape provides three main visualizations, addressing each of the three data granularity levels in recipe analysis; RecipeMap (Figure 1a) presents a bird's-eye view of *recipes in clusters* generated by the system. Each point on the map is a recipe, and the distance between them indicates their similarity. RecipeDeck (Figure 1b) enables an in-depth inspection and pairwise comparison of *individual recipes*. RecipeStat (Figure 1c) visualizes usage patterns of *individual cooking*

[1]`http://www.yummly.com`

CHI 2018, April 21–26, 2018, Montreal, QC, Canada

ISBN 978-1-4503-5620-6/18/04…$15.00

DOI: `https://doi.org/10.1145/3173574.3174025`

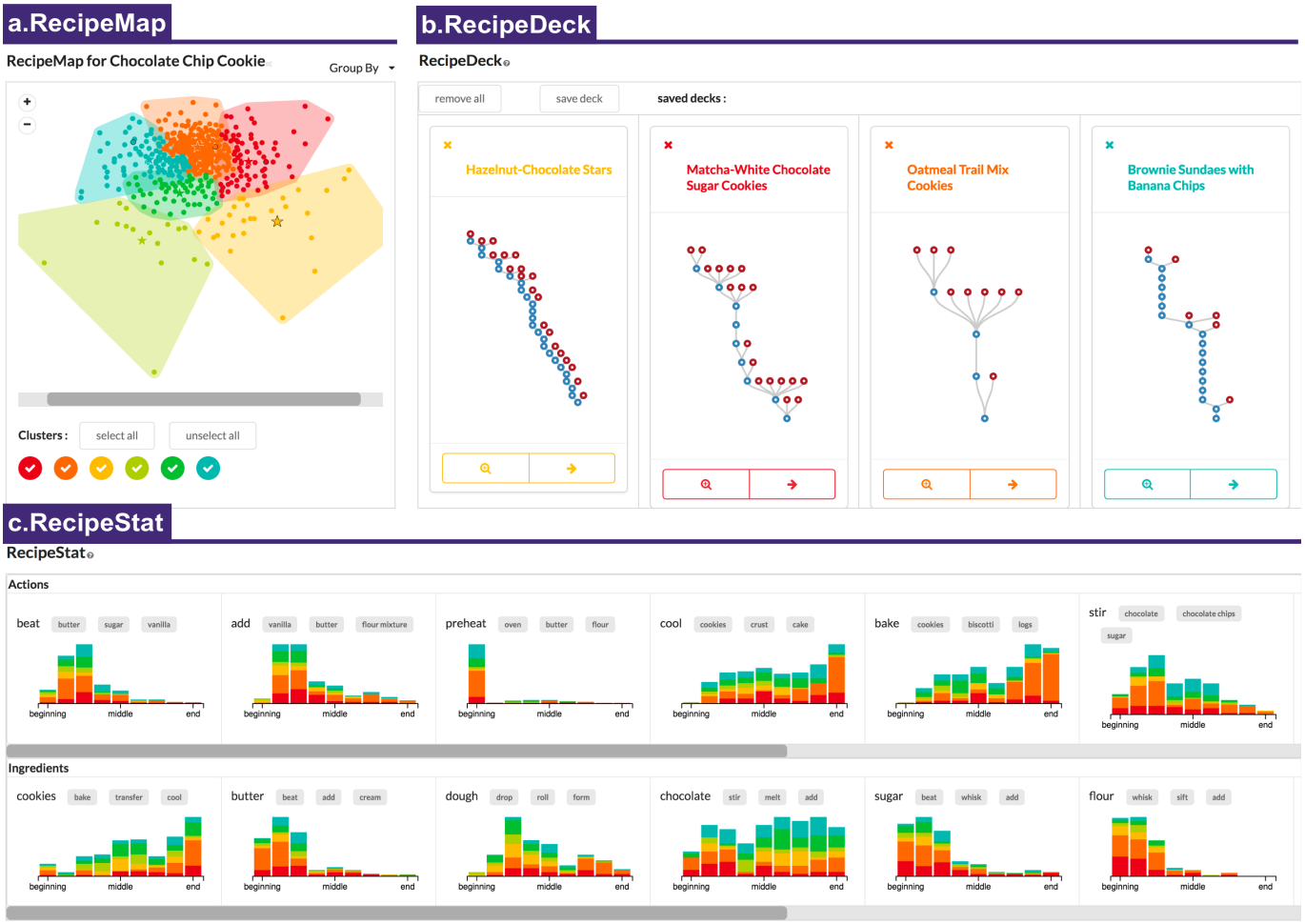


Figure 1: RecipeScape is an interface for analyzing cooking processes at scale with three main visualization components: (a) RecipeMap provides clusters of recipes with respect to their structural similarities, (b) RecipeDeck provides in-depth view and pairwise comparisons of recipes, and (c) RecipeStat provides usage patterns of cooking actions and ingredients.

actions and ingredients. Providing these visualizations requires processing and analyzing a large number of recipes. To achieve this goal, we present a computational pipeline (Figure 6) that scrapes recipes available online, converts them into a tree representation, and computes pairwise similarities. The pipeline represents each recipe as a tree (Figure 2) to capture the structural information (e.g., a sequence of actions, a set of ingredients involved in an action) embedded in a recipe. For accurate tree construction, we employ a machine-crowd workflow to label cooking actions and ingredients in recipe texts using a custom annotation interface(Figure 7).

In our crawled dataset of 487 chocolate chip cookie recipes and 510 tomato pasta recipes, 27,879 verbs are tagged by the Stanford CoreNLP's Part-of-Speech tagger [16]. Among them, our pipeline identified 14,988 verbs as relevant cooking actions. Also, our pipeline corrected 9,987 cooking actions that were mislabeled by the tagger, which is 40% of the cooking actions in the dataset.

In a qualitative evaluation with cooking professionals and culinary students, we asked participants to carry out a series of browsing and comparison tasks as well as to freely explore for new discoveries. Participants found data-driven evidence for subjective attributes of recipes like "general recipe" or "exotic recipe". They also discovered usage patterns of cooking actions and ingredients that distinguish one recipe cluster from another, by combining insights from the three visualizations of RecipeScape.

This paper makes the following contributions:

- **Design goals** for systems that aim to support analysis for cooking professionals by examining recipes in aggregate. These are identified from interviews with professional chefs, patissiers, cooking journalists, recipe website managers, and food business researchers.

- **RecipeScape**, **an interactive visual analytics interface** that enables browsing, comparison, and visualization of recipes at scale and supports analysis on three different

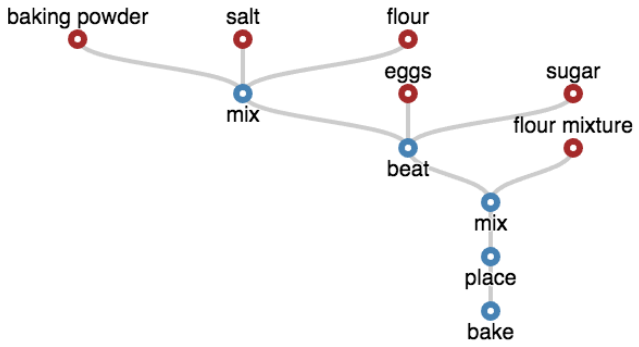


Figure 2: A tree generated by RecipeScape for "Daniela's Brownie" recipe, `https://www.epicurious.com/recipes/food/views/danielas-brownies-104362`.

levels of data granularity: *clusters of recipes, individual recipes, and individual cooking actions and ingredients.*

- **A computational pipeline** that scrapes recipe instructions from online websites, extracts the cooking action and ingredient information using a machine-crowd workflow, translates them into a tree representation, and computes similarities between pairs of recipes.

RELATED WORK

In RecipeScape, we focus on analyzing structures in cooking instructions at scale. It builds on prior work in three research areas; (1) data-driven culinary analytics (2) visual analytics for structured data (3) tree representations and comparison methods for recipes

Data-Driven Culinary Analytics

The research community has investigated a wide range of computational methods for analyzing and mining cooking knowledge. For example, constructed using a large dataset of ingredients, ingredient networks [30] and flavor networks [1, 33] can judge which ingredients go well together and which ones do not. They can also be used to recommend recipes with replaceable ingredients [28]. Deep neural networks can be trained to translate recipes from one style of cuisine to another [12], or to generate flavorful and novel recipes as well as humans [20], or to generate a high-quality text recipe from a food image and vice versa [25]. Also, ingredient similarity can predict recipe ratings [37], and user reviews of recipes can predict their attributes [6]. Features extracted from the recipe text can predict the gender of the recipe uploader [21]. PlateMate [17] analyzes nutrition information from food photographs using crowdsourcing. PlateClick uses a quiz-like visual interface for comparing two food images to elicit user's food preferences [35].

While most existing research focuses on analyzing recipes using ingredient similarities and food image similarities, RecipeScape introduces *structural similarity* to data-driven culinary analytics. We use a tree (Figure 2) to represent a recipe structure. In RecipeScape, the structural similarity of recipes is dependent on both syntactic similarities from the tree shape, and on semantic similarities between the node labels that represent ingredients and actions.

Visual Analytics for Structural Data

Interfaces for analyzing instructions and workflows have been a subject of rich prior work. Sifter [18] is an interface for browsing, comparing, and analyzing a large collection of web-based image manipulation tutorials. Delta [13] is an interface for comparing pairwise similarities of image processing workflows. Visualizing histograms of parameters such as stroke length and brush sizes helped 3D artists to compare digital sculpting workflows [26]. Visualizing worker behavior and worker output in crowdsourcing workflow has been found to be effective for crowdsourcing quality control [23]. Visualizing sequences of intermediate steps students take in problem-solving helped identify different strategies and points of confusion [34]. Visual analytics tools can analyze temporal data, such as tracking and comparing different versions of a slide deck [7], mining statistical insight from event sequences [15], analyzing patterns in health records [27], and finding similar student records [8].

To support analysis of recipe instructions at scale, our approach extends existing research on interfaces for instruction analysis by combining visualization and analysis on clusters of instructions with the lower level analysis features.

Tree Representation and Similarity Comparison

There are two areas where tree representations are widely used to compare structural similarities: comparing phylogenetic trees in bioinformatics and detecting code clones in software engineering. Additionally, we discuss how tree edit distance could be applied to culinary analytics.

A phylogenetic tree is a branch diagram which represents evolutionary dependencies of biological species. Researchers in bioinformatics have been comparing phylogenetic trees by using statistical and structural metrics such as maximum-likelihood of evolutionary parameters [36], neighbor-joining [24], and nodal-distances [3].

Tree similarity comparisons are also popular in code clone detection. For judging structural similarities in code clones, algorithms use features like characteristic vectors [9], syntax patterns [14], and token sequences of syntax trees [2, 11]. For judging semantic similarities, algorithms use graph isomorphism methods to dependency graphs [22] of source code, where nodes represent expressions and statements.

To compute recipe structure similarity, we use a tree edit distance [29] method. Tree edit distance methods find a sequence of operations like addition, deletion, and substitution of nodes, each associated with a cost, which minimizes the total cost to convert one tree to another. Computing the edit distances between unordered, labeled trees is NP-Complete [39] even for binary trees with the label alphabet size of two. However, we build recipe representation as an ordered tree, i.e., the first child of every node is always a cooking action. We then employ polynomial-time algorithms for ordered labeled trees [38, 39] to calculate the edit distances.

FORMATIVE STUDY

We conducted a series of interviews with cooking professionals to understand how they use online recipes in their current analysis practices and how they might benefit from using them more efficiently.

Interviews

We interviewed 2 professional chefs (10 or more years of experience), 1 patissier (5 years of experience), 2 cooking journalists (20 years of experience, 5 years of experience), 2 recipe website managers (3 years of experience), and 3 food business researchers (15 or more years of experience).

Each session took approximately 90 minutes and included a semi-structured interview followed by a feedback session on the general idea of exploring recipes at scale.

We asked our participants (1) what kinds of recipe analysis are involved in their daily job and what their current practices are, (2) what additional analysis would make their job more convenient, and (3) what analysis could increase and expand their capabilities.

After open-ended discussions, we presented component sketches designed to support recipe exploration at scale to encourage further discussion and ground their feedback. We used these low-fidelity sketches since exploring recipes at scale is an unaccustomed concept even for professionals. The participants were encouraged to think aloud while they were browsing and comparing recipes using the components presented in the sketches[2].

Results

Current Practice: Casual but Interrogative Browsing

When developing new menus, professional chefs and patissiers said they compare recipes and search for unusual ingredients or creative uses of usual ingredients. A professional chef noted, *"There's no such thing as a completely novel recipe"*, and emphasized the importance of casually browsing a variety of recipes to maximize exposure to diverse recipes even for a single ingredient or a single dish.

When planning an article about a specific dish, both cooking journalists explained that they browse more than ten different recipes to understand common cooking actions, common ingredients, and common tips and hacks. They visit restaurants and ask the chef about the recipe, or use cookbooks and publicized recipes by famous chefs to grasp these characteristics. When they write articles with specific themes, they sometimes ask chefs to either devise or introduce recipes that are unfamiliar to the general public.

Recipe website managers examine recipes and label them with tags for feeding the recipes into their search engine. This is manually done, because existing algorithms available to end users cannot accurately provide rich and appropriate tag labels.

In summary, cooking professionals with analytical needs commonly search for unusual ingredients and cooking process, casually browsing a variety of recipes for a single dish.

Desired Information: Statistics of Recipes

While all participants agreed they want to be able to easily find recipes with uncommon cooking actions and ingredients, their reasons are different. We found professional chefs and patissiers rely on reverse engineering a dish to study unusual usages of a specific cooking skill or an ingredient. But reverse engineering requires a lot of trial and error and exact replication is very hard to achieve. Cooking journalists are interested in creative variants or unusual reinterpretations of a dish to be able to introduce them to the public. However,"creative" and "unusual" are very subjective measures, and they normally spend weeks trying to find something unimaginable for their audience. Recipe website managers are interested in standard recipes, and evidence to claim "standardness" of the recipes.

Food business researchers want to measure to what extent a de-facto standard version of a dish has been established. They explained if recipes are more standardized for a specific food, it is likely to be a saturated market, whereas if the recipes vary, the corresponding food business is still in its growth stage. It is used as one of the metrics to evaluate the market cycle and to predict an upcoming trend in the food business.

All participants want a categorization feature. Suggested ideas of categorization criteria include specific ingredient constraints like "gluten-free" or "sugar-free", cooking tool constraints like "no oven" or "microwave only", and types of cuisine. They also want more subjective criteria like uniqueness, difficulty of the recipe, and different ways of cooking the dish.

In summary, recurring needs expressed by the professionals are methods for discovering common and uncommon cooking actions and ingredients, and clues from which they could answer questions like how a set of recipes differ from another set of recipes, and what factors contribute to the difference. Also, participants want to easily discover "average" or "standard" versions of a dish, and evidence of subjective attributes like uniqueness or difficulty of recipes.

System Design Goals

Two researchers iteratively analyzed the interview data more than four times in total with an interval of at least two days between sessions to enhance validity. We identified 52 topics during this process, and subsequently clustered these topics into 13 different themes.[3] From the 13 topics, we focused on the needs that would benefit most from exploring recipes at scale. We excluded several classes of topics beyond the scope of this work: (1) those requiring information that is unavailable in online recipes, such as trends and cultural information; (2) those relevant to real-time cooking support like hacks and mistakes; and (3) those already supported by existing systems, like categorization criteria and ingredient-based information.

[2]The component sketches are available in the supplemental materials.

[3]The 52 topics and 13 themes are available in the supplemental materials.

Four researchers independently brainstormed hypotheses and frames of explanation for the remaining topics. Through rounds of discussion, we agreed that the data granularity framework is most explanatory. Then we classified the remaining topics into their data granularity and derived the three design goals.

The interview results emphasize the need to provide users with an interactive analytics system that enables browsing and comparing recipes. Based on the analysis of the interviews and the participants' suggestions, we identified three design goals for **tools to support recipe analysis at scale**. The individual design goals address three different levels of data granularity for recipe analysis: ingredients and cooking actions (D1), recipes (D2), and clusters of recipes (D3).

D1. **Provide statistical information about ingredients and cooking actions** to support analysis at the individual ingredient/cooking action level, such as answering questions like "what are some unusual ingredients?" and "what are some unusual cooking actions?".

D2. **Provide in-depth examination and comparison of individual recipes** to support instruction level analysis, such as answering questions like "what are the detailed step by step instructions of this recipe?" and "what are shared instructions and ingredients between two recipes?".

D3. **Provide analysis for recipes in aggregate** to support cluster level analysis and between-cluster similarities and across-cluster differences, such as answering questions like "what makes a set of recipes standard of the dish?", "what are some creative variants of this recipe?", and "what are different ways of cooking the dish?"

RECIPESCAPE

To address these design goals, we present RecipeScape (Figure 1), an interactive tool that enables browsing, comparison, and analysis of recipes at scale. RecipeScape provides three main components: RecipeMap, RecipeDeck, and RecipeStat. RecipeMap (Figure 3) provides an overview of recipes with cluster information. RecipeDeck (Figure 4) provides information about individual recipes with the original description, the corresponding tree representation, and pairwise comparison of recipes. RecipeStat (Figure 5) provides the usage patterns of cooking actions and ingredients in the selected clusters of recipes from RecipeMap.

RecipeMap

RecipeMap (Figure 3) supports queries related to D3 (Provide analysis for recipes in aggregate), e.g., the prototypical recipes, outlier recipes, and a bird's-eye view of different clusters of recipes for a specific dish. Each point on the map is a recipe. The distance between recipes reflects the structural similarity of the respective recipe instructions. The prototypical recipe, i.e., most structurally representative one in each cluster, is marked with a star (Figure 3c). Users can select clusters by clicking on the color key at the bottom (Figure 3a). Users can also select any recipe on the map (Figure 3b) for an in-depth view. When one or more clusters are selected,

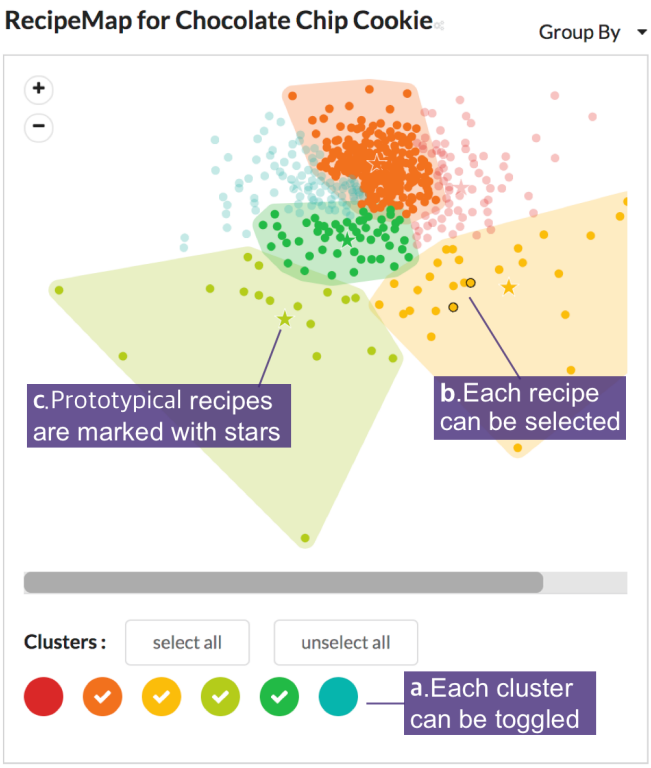


Figure 3: RecipeMap provides clusters of recipes with respect to their structural similarities.

RecipeStat updates to only reflect the information in the chosen clusters. Similarity metrics and clustering algorithms are discussed in the Computational Pipeline Section.

RecipeDeck

RecipeDeck (Figure 4) supports queries related to D2 (Provide in-depth examination and comparison of individual recipes). User-selected recipes in RecipeMap are added here (Figure 4b), with a default view of the tree representation. The tree in this view does not have any labels, allowing users to focus on structural comparisons of multiple recipes on RecipeDeck. Users can click on the magnifier icon (Figure 4c) for a detailed popup with the recipe text and the labeled tree representation. Users can also click on the right arrow icon to view textual instructions without invoking the popup. Furthermore, users can perform a pairwise comparison of two recipes by selecting two recipes on RecipeDeck, and then clicking "compare". Upon clicking, a popup (Figure 4a) opens with a side-by-side comparison of cooking action sequences and an ingredient comparison of the two recipes in a Venn diagram.

RecipeStat

RecipeStat (Figure 5) supports queries related to D1 (Provide statistical information about ingredients and cooking actions). For the 10 most used cooking actions in the selected clusters of recipes, users can examine how each cooking action is used at different stages of the cooking process in recipes of

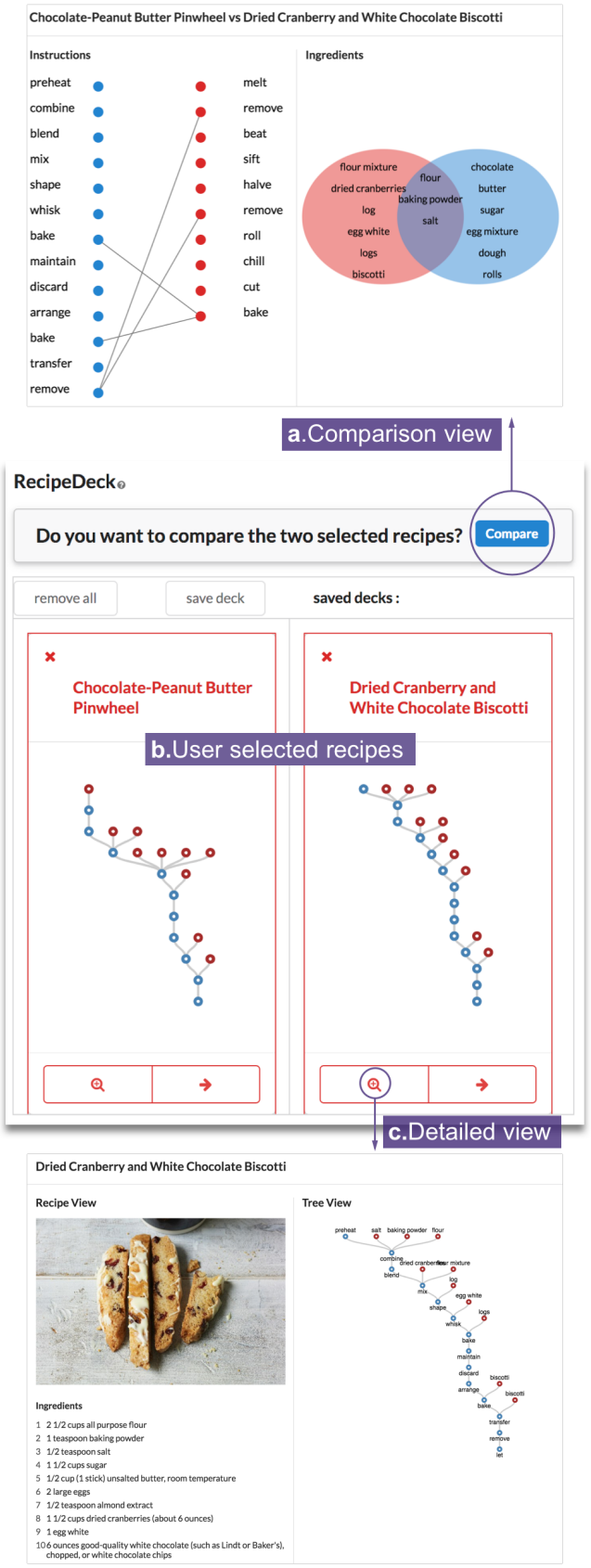


Figure 4: RecipeDeck: RecipeDeck displays (b) user selected recipes and provides (c) a detailed view and (a) pairwise comparisons of recipes.

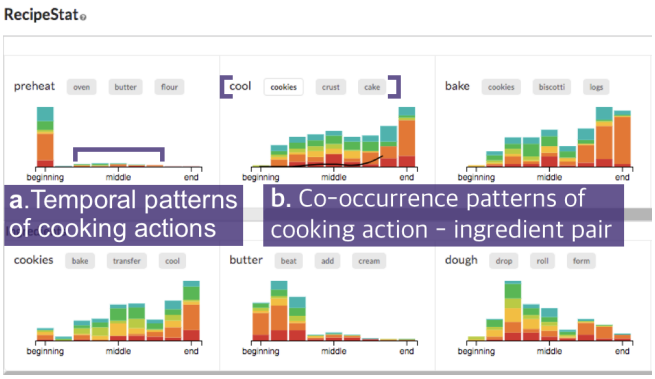


Figure 5: RecipeStat provides temporal usage trends of cooking actions and ingredients, and co-occurrences patterns of cooking action-ingredient pairs.

the selected cluster. This information supports the discovery of different approaches to cooking a dish. For example, users can click on the recipes with "Preheat" occurring in different stages of the cooking process (Figure 5a). Users are able to hover the bars on the histogram to see which recipes correspond to the specific selection. The corresponding recipes are highlighted in RecipeMap, and are added to RecipeDeck with a click. Users can also click on the top three most used ingredients next to each cooking action label to view the usage pattern of cooking action-ingredient pairs like cool-cookies (Figure 5b). When the cluster selection changes, RecipeStat recalculates the statistics and redraws the histogram. The same visualization and functionality is also provided for the 10 most used cooking ingredients.

COMPUTATIONAL PIPELINE

In this section, we describe the underlying pipeline (Figure 6) of RecipeScape for constructing graphical representations of recipes and obtaining similarity metrics by highlighting the data gathering, parsing, and similarity comparison steps.

Data Gathering

In the data gathering step, we crawl all search results for a queried dish, like chocolate chip cookie and tomato pasta, from recipe websites that use the schema.org's Recipe scheme[4]. Schema.org's schemes are agreed templates for storing structured data, with specific document elements like ingredients and instructions. Most major recipe websites use the Recipe scheme as their internal representation, which makes using it for data gathering step more generalizable.

Parsing

In the parsing step, we use a Stanford CoreNLP [16]'s Part-of-Speech (POS) tagger to label verbs and nouns in recipe instructions crawled in the data gathering step. Most state-of-the-art POS taggers are statistically trained using mostly declarative sentences. As a result, their performance is rather limited with imperative sentences in recipe instructions like

[4]http://schema.org/Recipe/

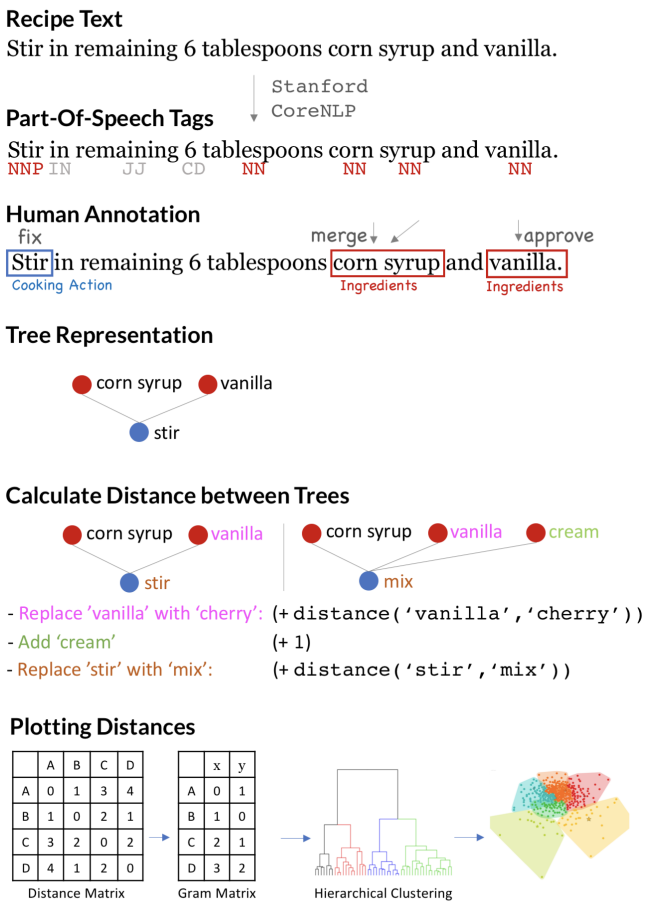


Figure 6: Our computational pipeline combines Part-of-Speech tagging and human annotation to convert recipe text into a tree representation, and calculates pairwise distance between recipes.

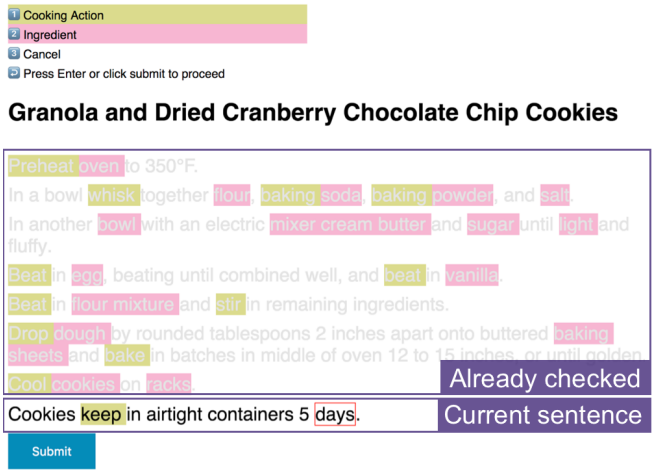


Figure 7: A web-based annotation interface for correctly labeling recipe actions and ingredients.

"Whisk in chocolate hazelnut spread until combined and remove from heat.". The parsers recognize "Whisk" as a noun and "spread" as a verb, but they are a verb and a noun in this sentence, respectively. To overcome this drawback and to more accurately identify tokens for cooking actions and cooking ingredients, we recruited 12 annotators to use a custom web-based interface (Figure 7) for annotating recipe instructions. After an iteration with the POS tagger, the crowd annotator fixes the tags that are labelled incorrectly. The parser then generates an ordered tree representation for each recipe, where the first child of every node is a cooking action and the siblings are the ingredients involved in that action.

There are several reasons we decided to use a tree structure over a sequence. Our preliminary study [4] using a sequence of cooking actions and ingredients, and string edit distances did not yield meaningful clusters to users, meaning the sequence representation did not capture meaningful structural differences in the cooking context. The clustering result was highly sensitive to the length of the sequences (Figure 8), which motivated us to consider a branching structure rather than a linear structure. Also, there exists information that cooking actions or ingredient alone cannot capture. For example, generic action verbs like "place", "remove", and "heat" yields different interpretations depending on whether they are associated with ingredients or cooking tools. Thus it is advantageous to incorporate a hierarchical structure.

Similarity Comparison

In order to obtain similarities between recipes, we use a tree edit distance [38], a commonly used technique for comparing tree structures. It finds a sequence of operations like adding, removing, and relabeling nodes, each associated with a cost, which minimizes the total cost to convert one tree to another. To incorporate the semantic difference between individual cooking actions and ingredients in capturing the structural difference, we dynamically adjust the weights associated with the relabel operations. These weights are calculated from a pre-trained word embedding model with one million recipe instructions [25]. We use the cosine similarity between two words in the embedding space as the weight associated with relabel operations; the associated cost then is $(1 - weight)$, because weight is the similarity, and the cost is the difference. This is motivated by intuition that a resulting structure from replacing "mix" with "add" should be considered more similar than that of replacing "mix" with "heat". For add and remove operations, we assign a unit length cost of 1. This discourages adding and removing of nodes and promotes relabeling of nodes. It is another attempt at making difference measures less sensitive to lengths of the structure, a limitation we encountered when using a sequential representation.

This similarity information is stored in a pairwise distance matrix, where each element is the tree edit distance between the corresponding recipes. The distance matrix is then converted into x,y coordinates using the Gram matrix [32, 5] to preserve distance information. We use the calculated coordinates to plot the recipes on RecipeMap for a bird's-eye view of structural similarities between all recipes. To highlight the

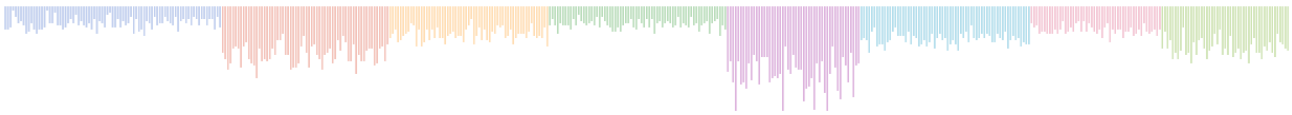
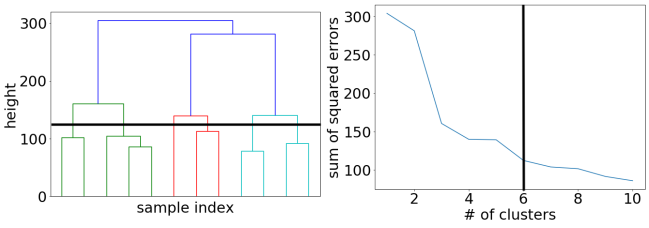


Figure 8: Lengths of recipes and their cluster membership shown with colors: this clustering result is based on sequence representation, and are dominated by the sequence lengths.



(a) A Dendrogram

(b) The Elbow Method

Figure 9: (a) Dendrogram for chocolate chip cookie recipes. (b) Elbow Method: a plot of unexplained variance vs number of clusters: adding another cluster at six clusters does not improve the validity. From both (a) and (b), six clusters seem reasonable.

structurally different clusters of recipes, we use hierarchical clustering [10]. Hierarchical clustering methods do not require the predetermined number of clusters before the analysis, which is common in other popular clustering methods, e.g., K-means clustering. With hierarchical clustering, the researcher can choose the most appropriate number of clusters that suits their analytical purpose after calculating the similarities in the data. We found this post-analysis control over the number of clusters advantageous in this study, because we do not know how many distinct approaches to cooking a dish exists in advance. To decide the number of clusters to display to users, we used both dendrogram [19] and the elbow method [31]. A dendrogram is an arrangement of the clusters produced by hierarchical clustering based on a distance metric. The elbow method provides a graph of the amount of variance explained by the number of clusters. There is no definitive answers to how many clusters should be selected, because the interpretation of the resulting hierarchical structure is context-dependent and often several solutions are equally good. For our chocolate chip cookie example, the dendrogram (Figure 9(a)) suggests six clusters are a reasonable choice. Consulting the elbow method plot (Figure 9(b)), adding another cluster at six causes a minimal change on the slope, so we pick six clusters.

Pipeline Results

Dataset: We crawled 487 recipes for "chocolate chip cookie" and 510 recipes for "tomato pasta" from `epicurious.com`. We chose chocolate chip cookie and tomato pasta because they are popular and accessible dishes.

Parsing Accuracy: The ground truth label for all 214,109 tokens in the crawled cooking recipes are unavailable to assess the precision/recall accuracy. However, we made a significant improvement over the baseline parser. Out of 27,879 verbs tagged by the Stanford CoreNLP's POS tagger [16], 14,988 were cooking actions confirmed by human annotators. This means only 54% of the machine-tagged labels are relevant to culinary analysis. Furthermore, human annotators corrected 9,987 cooking actions the NLP tagger mislabeled. This counts up to 40% of the final 24,975 cooking action verbs used in the study, which represent improvements realized by human annotation.

RECIPESCAPE PROJECT WEBSITE

We provide links to our dataset, source code repository, and dashboard interface at `https://recipescape.kixlab.org/`.

EVALUATION

RecipeScape is a tool for open-ended discovery by exploring cooking recipes of a single dish at scale and is not designed as an assistant to improve cooking skills. Hence, we investigated the effectiveness of novel exploration techniques for the professional analytic needs with open-ended qualitative studies rather than a task-based evaluation that measures an improvement over a baseline. Goals of evaluation were (1) to assess the feasibility of representing cooking process as a tree, and (2) to gain feedback on the effectiveness of RecipeScape in answering the following analytical questions that follow from the three design goals:

Q1. What are patterns of common and unusual ingredient and cooking action usage?

Q2. What are different ways of cooking a dish?

Q3. What are representative recipes of cooking a dish?

Q4. What are outlier recipes of cooking a dish?

Q5. What are the simplest and most complex recipes of cooking a dish?

Q6. What is the evidence for answers in Q1-Q5?

User Study with Cooking Professionals

We reached out to the same experts we interviewed in the earlier stage of the research for understanding analysis tasks of cooking professionals. Among them, two recipe website managers, one professional chef, and one cooking journalist participated in the interface evaluation study followed by a semi-structured interview, which lasted two hours. Experts were given a 5-minute tutorial of the interface and asked to freely explore and evaluate RecipeScape. They were asked to choose one or more clusters on RecipeMap and find characteristics that define the cluster. Experts were encouraged to think aloud as they browsed and compared recipes, and how they interpreted the findings.

Lab Study with Culinary Students

We invited 7 culinary students in a 90-minute session each. Participants were first asked to fill out a questionnaire on their current practices of searching and browsing recipes, i.e., how they search for recipes, when and how often they search for recipes. They were given a brief tutorial of the RecipeScape interface. Then they were asked to explore and use the interface. After participants indicated that they felt confident using the interface, we gave them 30 minutes to evaluate the interface by carrying out tasks to answer the questions Q1 to Q6 outlined above. A session ended with an interview to understand deeper the observed interface usage patterns, and solicit qualitative feedback about the interface.

Results

We summarize the results and present main findings with respect to the three design goals, patterns of tool usage, and usability and usefulness of RecipeScape.

D1. Ingredient and action level analysis

The professional chef made an observation that recipes with must-have ingredients are plotted at the center of RecipeMap, and the outer ring of the recipes have additional ingredients that go well with the dish but are not necessary. The chef found the recipes on the edges to have ingredients that reflect more personal preferences, such as use of goat cheese and artichoke for pasta. The chef was also surprised to see recipes that use salt in the later stages of cooking pasta in RecipeStat, as opposed to the convention of using it in earlier stages, e.g. cooking pasta noodles in salted water. He mentioned, *"This is a professional tip that good restaurants use to make the first spoon of pasta taste extra sweet. If you put salt on tomato, it really brings out the sweetness. I'm surprised this hack is captured."* For one cluster of chocolate chip cookie recipes, the cooking journalist wanted to find cookies with decorations, and examined RecipeStat for recipes where "cover" was mostly used in later stages of the recipes. The corresponding highlighted recipes in RecipeMap agreed with her hypothesis. One student participant used the identical approach to find recipes with sugar frosting.

D2. Individual recipe level analysis

The professional chef spent significant time examining individual recipes near the edge of RecipeMap. He noted *"We (professional chefs) sometimes start from a specific main ingredient and seek creative interpretations. I find these recipes near the edges are more exotic."*

The cooking journalist frequently used the pairwise comparison of two adjacent recipes on RecipeMap to examine replaceable ingredients and actions.

Four out of seven student participants found the tree diagram in the in-depth view especially helpful for grasping the overall process of individual recipes and they felt confident about cooking the dish only by looking at the tree. One student specifically noted, *"I find recipes in the usual text format hard to visualize the process, because the ingredient sections and the instruction sections are separate. But this tree diagram summarizes the process very well, I can easily picture the cooking process."*

D3. Cluster level analysis

Two experts and five out of seven student participants mentioned the clusters reflect different ways of cooking very well. After selecting a cluster from RecipeMap, experts examined RecipeStat and formed hypotheses of what some characterizing attributes of the cluster might be. Then they used other components to verify their hypotheses. For example, the cooking journalist looked at one cluster and noticed there are baking soda and baking powder in the most used ingredients list in RecipeStat. She immediately mentioned, *"Recipes in this cluster probably do not use any eggs and probably involve baking in the later stages."*, which was confirmed by examining the recipes in the cluster in detail. The journalist also found a cluster where water was in the most used ingredients list in RecipeStat. She then checked whether there is "chill" or "cool" in RecipeStat for cooking actions. When she found "chill", she mentioned *"These are the recipes for more crispy cookies. You use water so the ingredients don't stick as much, resulting in crispy cookies. This kind of dough tastes better when you cool them."* After reviewing a few recipes in the cluster, the hypothesis was verified. The professional chef discovered "salted water" in RecipeStat for one cluster. He then mentioned *"I would trust the recipes in this cluster more than the other ones. The fact that people described salted water, not just water, implies the instructions are more friendly and detailed."* Upon examining a few recipes in the cluster, the recipes were indeed more detailed than the others.

Patterns of Tool Usage

Every participant started from RecipeMap by choosing cluster(s) of their interest. Then they would select recipes in the center of the clusters and examine them in detail. Some would repeat these steps back and forth, but RecipeStat was always used in the last stage. When asked, participants explained RecipeMap is a good place to start analysis due to its similarity presentation. We learned the design of RecipeStat assumed knowledge of histograms, which some participants did not have. Participants explained they needed to examine a few individual recipes in depth to understand the information displayed in RecipeStat, which led to the usage pattern.

Usability and Usefulness of RecipeScape

The professional chef and cooking journalist noted RecipeScape would be useful for learning about recipes. The professional chef mentioned he would use this tool to understand a dish that he does not have much experience in. The cooking journalist wanted to use RecipeScape for Bibimbap, a traditional Korean dish. She explained there are many recipes of Bibimbap in English, because it is internationally popularized. She said RecipeScape would reveal diverse approaches that capture how this traditional dish is interpreted outside Korea.

Six student participants noted they want to use this tool in their studies if it supported dishes of their interest. They found similar recipes being located closely together in RecipeMap to be useful in comparison to existing services they use, because browsing in RecipeScape does not involve going back and forth between the list of search results and the specific recipe page. Three student participants also men-

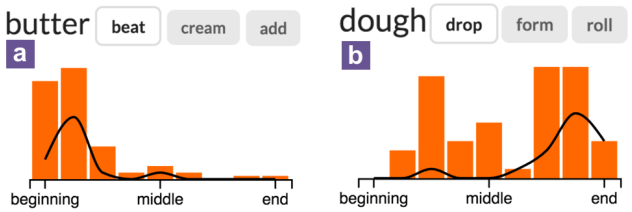


Figure 10: Usage patterns of two contrasting action-ingredient pairs (beat-butter and drop-dough) in chocolate chip cookie recipes.

tioned RecipeScape would be useful when they're preparing for cooking contests, where they have to reinterpret an existing dish. Using RecipeMap to explore various recipes helped them not only brainstorm ideas but also simulate how their interpretations can be translated into recipes. For example, one participant said *"I had thought about using liquor for making creative cookies (as an assignment), but was not sure how I could do it. I was able to spot a recipe that uses liquor (Frozen Grand Marnier Torte with Dark Chocolate Crust and Spiced Cranberries) just by browsing the recipes outside (near the edges), and it helped me understand better how liquor could be used in cookies."*

DISCUSSION

We discuss findings, generalizability and possible limitations of this work.

Expert Knowledge in Naturally Crowdsourced Data

We observed an interesting example of an expert knowledge recovered in the analysis of crowd generated recipes supported by RecipeScape. Butter cream, made by beating the butter, is added in the last step of making the dough, because it inhibits gluten formation. According to RecipeStat, the beat-butter pair (Figure 10a) occurs mostly in the beginning of the cooking process, and drop-dough (Figure 10b), an intermediate product of making the dough, occurs after the creamed butter is made. Extending from this observation, it would be meaningful to further identify and characterize expert knowledge that is transferred and not transferred for informing further interaction designs around instructions.

Structural Representation and Parser Accuracy

In this research, the low accuracy of the NLP parser in generating the structural representation could make the analysis challenging. Ill-structured text and non-standard phrasing can lead to mislabels. To minimize confusion in such cases, we show the original recipe text next to the tree diagram in the detailed view for the parser errors to have less impact on user tasks. With a better performing Part-of-Speech tagger that successfully detects action verbs in imperative sentences, RecipeScape can immediately benefit from the algorithm by replacing the parser module.

Explainability of Clustering Algorithm

RecipeScape only provides clusters of recipes and does not provide explanations of the clustering results. In the end, users will have to make sense of the clusters generated by the algorithm. While this is inevitable for all clustering algorithms, we do have control over choosing the number of clusters after the similarity calculations. A number of ways can improve the explainability. Supporting manual labelling, or accompanying carefully designed topic modelling to generate themes for each clusters, or both could benefit the users.

Generalizability of the Pipeline

While RecipeScape is focused on culinary analytics, our pipeline is generalizable and could potentially apply to analyzing other instructions at scale. Researchers have explored how to present other kinds of procedural instructions like image manipulation tutorials, sculpting workflows. It varies in degree, but even in assembly or in photo manipulation tutorials where it seems like there is only one correct sequence of operations, there are often multiple feasible solutions. For example, when assembling a chair, one can start from the legs, the back, the armrests. A systematic analysis of instructions across domains is open to future study, but we believe our approach is still applicable to other domains.

Scalability of the Pipeline: More Data Dimensions

Our structural similarity comparison of tree representations allows adding more dimensions like time or tools. There remains a design decision of whether to treat these dimensions as separate nodes or as parameters of cooking action. Regardless, dynamically retrieving edit distance weights from a vector embedding space handles the semantic similarities of different dimensions. However, visualizing multiple dimensions and presenting them with meaningful interaction is an open challenge, which we hope to address in future work.

CONCLUSION AND FUTURE WORK

This paper presents RecipeScape, an interactive system for analyzing hundreds of recipes for a single dish by visualizing summaries of their structural patterns. Our user study with cooking professionals and culinary students demonstrates that RecipeScape provides data-driven evidence to usual and unusual ingredients and cooking actions, common and exotic recipes, and different approaches to cooking a dish.

There are a number of directions for possible future studies. As an immediate next step, we plan to extend this work to video-based recipes and how-to videos. Examining rich context embedded in videos in aggregate could uncover trends and patterns, and analyzing them at scale will be able to provide answers to questions text recipes at scale cannot.

ACKNOWLEDGMENTS

We thank the Brown Institute for Media Innovation at Stanford University and members of KIXLAB at KAIST for their support and feedback. This work was also supported in part by HRHRP at KAIST and by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korean government (MSIT) (No.2017-0-01217, Korean Language CALL Platform Using Automatic Speech-writing Evaluation and Chatbot).

REFERENCES

1. Yong-Yeol Ahn, Sebastian E Ahnert, James P Bagrow, and Albert-László Barabási. 2011. Flavor network and the principles of food pairing. *Scientific reports* 1 (2011). DOI:http://dx.doi.org/10.1038/srep00196
2. Ira D Baxter, Aaron Quigley, Lorraine Bier, Marcelo Sant'Anna, Leonardo Moura, and Andrew Yahin. 1999. CloneDR: clone detection and removal. In *Proceedings of the 1st International Workshop on Soft Computing Applied to Software Engineering*. 111–117.
3. John Bluis and Dong-Guk Shin. 2003. Nodal distance algorithm: Calculating a phylogenetic tree comparison metric. In *Bioinformatics and Bioengineering, 2003. Proceedings. Third IEEE Symposium on*. IEEE, 87–94. DOI: http://dx.doi.org/10.1109/bibe.2003.1188933
4. Minsuk Chang, Vivian M Hare, Juho Kim, and Maneesh Agrawala. 2017. Recipescape: Mining and analyzing diverse processes in cooking recipes. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, 1524–1531. DOI:http://dx.doi.org/10.1145/3027063.3053118
5. GM Crippen. 1978. Note rapid calculation of coordinates from distance matrices. *J. Comput. Phys.* 26, 3 (1978), 449–452. DOI: http://dx.doi.org/10.1016/0021-9991(78)90081-5
6. Gregory Druck. 2013. Recipe attribute prediction using review text as supervision. In *Cooking with Computers 2013, IJCAI workshop*.
7. Steven M Drucker, Georg Petschnigg, and Maneesh Agrawala. 2006. Comparing and managing multiple versions of slide presentations. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*. ACM, 47–56. DOI: http://dx.doi.org/10.1145/1166253.1166263
8. Fan Du, Catherine Plaisant, Neil Spring, and Ben Shneiderman. 2017. Finding similar people to guide life choices: Challenge, design, and evaluation. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 5498–5544. DOI:http://dx.doi.org/10.1145/3025453.3025777
9. Lingxiao Jiang, Ghassan Misherghi, Zhendong Su, and Stephane Glondu. 2007. Deckard: Scalable and accurate tree-based detection of code clones. In *Proceedings of the 29th international conference on Software Engineering*. IEEE Computer Society, 96–105. DOI: http://dx.doi.org/10.1109/icse.2007.30
10. Stephen C Johnson. 1967. Hierarchical clustering schemes. *Psychometrika* 32, 3 (1967), 241–254. DOI: http://dx.doi.org/10.1007/bf02289588
11. Toshihiro Kamiya, Shinji Kusumoto, and Katsuro Inoue. 2002. CCFinder: a multilinguistic token-based code clone detection system for large scale source code. *IEEE Transactions on Software Engineering* 28, 7 (2002), 654–670. DOI: http://dx.doi.org/10.1109/tse.2002.1019480
12. Masahiro Kazama, Minami Sugimoto, Chizuru Hosokawa, Keisuke Matsushima, Lav R Varshney, and Yoshiki Ishikawa. 2017. Sukiyaki in French style: A novel system for transformation of dietary patterns. *arXiv preprint arXiv:1705.03487* (2017).
13. Nicholas Kong, Tovi Grossman, Björn Hartmann, Maneesh Agrawala, and George Fitzmaurice. 2012. Delta: a tool for representing and comparing workflows. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1027–1036. DOI:http://dx.doi.org/10.1145/2207676.2208549
14. Nicholas A Kraft, Brandon W Bonds, and Randy K Smith. 2008. Cross-language Clone Detection.. In *SEKE*. 54–59.
15. Sana Malik, Fan Du, Megan Monroe, Eberechukwu Onukwugha, Catherine Plaisant, and Ben Shneiderman. 2015. Cohort comparison of event sequences with balanced integration of visual analytics and statistics. In *Proceedings of the 20th International Conference on Intelligent User Interfaces*. ACM, 38–49. https://doi.org/10.1145/2678025.2701407
16. Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit.. In *ACL (System Demonstrations)*. 55–60. DOI: http://dx.doi.org/10.3115/v1/p14-5010
17. Jon Noronha, Eric Hysen, Haoqi Zhang, and Krzysztof Z Gajos. 2011. Platemate: crowdsourcing nutritional analysis from food photographs. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. ACM, 1–12. DOI:http://dx.doi.org/10.1145/2047196.2047198
18. Amy Pavel, Floraine Berthouzoz, Björn Hartmann, and Maneesh Agrawala. 2013. Browsing and analyzing the command-level structure of large collections of image manipulation tutorials. *Citeseer, Tech. Rep.* (2013).
19. JB Phipps. 1971. Dendrogram topology. *Systematic zoology* 20, 3 (1971), 306–308. DOI: http://dx.doi.org/10.2307/2412343
20. Florian Pinel and Lav R Varshney. 2014. Computational creativity for culinary recipes. In *CHI'14 Extended Abstracts on Human Factors in Computing Systems*. ACM, 439–442. DOI: http://dx.doi.org/10.1145/2559206.2574794
21. Markus Rokicki, Eelco Herder, Tomasz Kuśmierczyk, and Christoph Trattner. 2016. Plate and prejudice: Gender differences in online cooking. In *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization*. ACM, 207–215. DOI: http://dx.doi.org/10.1145/2930238.2930248

22. Chanchal K Roy, James R Cordy, and Rainer Koschke. 2009. Comparison and evaluation of code clone detection techniques and tools: A qualitative approach. *Science of computer programming* 74, 7 (2009), 470–495. DOI: http://dx.doi.org/10.1016/j.scico.2009.02.007

23. Jeffrey Rzeszotarski and Aniket Kittur. 2012. CrowdScape: interactively visualizing user behavior and output. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*. ACM, 55–62. DOI: http://dx.doi.org/10.1145/2380116.2380125

24. Naruya Saitou and Masatoshi Nei. 1987. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular biology and evolution* 4, 4 (1987), 406–425. DOI:http://dx.doi.org/10.1093/oxfordjournals.molbev.a040454

25. Amaia Salvador, Nicholas Hynes, Yusuf Aytar, Javier Marin, Ferda Ofli, Ingmar Weber, and Antonio Torralba. 2017. Learning Cross-modal Embeddings for Cooking Recipes and Food Images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. DOI: http://dx.doi.org/10.1109/cvpr.2017.327

26. Christian Santoni, Claudio Calabrese, Francesco Di Renzo, and Fabio Pellacini. 2016. SculptStat: Statistical Analysis of Digital Sculpting Workflows. *arXiv preprint arXiv:1601.07765* (2016).

27. Ben Shneiderman, Catherine Plaisant, and Bradford W Hesse. 2013. Improving healthcare with interactive visualization. *Computer* 46, 5 (2013), 58–66. DOI: http://dx.doi.org/10.1109/mc.2013.38

28. Tiago Simas, Michal Ficek, Albert Diaz-Guilera, Pere Obrador, and Pablo R Rodriguez. 2017. Food-bridging: a new network construction to unveil the principles of cooking. *arXiv preprint arXiv:1704.03330* (2017).

29. Kuo-Chung Tai. 1979. The tree-to-tree correction problem. *Journal of the ACM (JACM)* 26, 3 (1979), 422–433. DOI: http://dx.doi.org/10.1145/322139.322143

30. Chun-Yuen Teng, Yu-Ru Lin, and Lada A Adamic. 2012. Recipe recommendation using ingredient networks. In *Proceedings of the 4th Annual ACM Web Science Conference*. ACM, 298–307. DOI: http://dx.doi.org/10.1145/2380718.2380757

31. Robert L Thorndike. 1953. Who belongs in the family? *Psychometrika* 18, 4 (1953), 267–276. DOI: http://dx.doi.org/10.1007/bf02289263

32. Warren S Torgerson. 1952. Multidimensional scaling: I. Theory and method. *Psychometrika* 17, 4 (1952), 401–419. DOI: http://dx.doi.org/10.1007/bf02288916

33. Kush R Varshney, Lav R Varshney, Jun Wang, and Daniel Myers. 2013. Flavor pairing in Medieval European cuisine: A study in cooking with dirty data. *arXiv preprint arXiv:1307.7982* (2013).

34. Yiting Wang, Walker M White, and Erik Andersen. 2017. PathViewer: Visualizing Pathways through Student Data. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 960–964. DOI: http://dx.doi.org/10.1145/3025453.3025819

35. Longqi Yang, Yin Cui, Fan Zhang, John P Pollak, Serge Belongie, and Deborah Estrin. 2015. Plateclick: Bootstrapping food preferences through an adaptive visual interface. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, 183–192. DOI: http://dx.doi.org/10.1145/2806416.2806544

36. Ziheng Yang. 1997. PAML: a program package for phylogenetic analysis by maximum likelihood. *Bioinformatics* 13, 5 (1997), 555–556. DOI:http://dx.doi.org/10.1093/bioinformatics/13.5.555

37. Ning Yu, Desislava Zhekova, Can Liu, and Sandra Kübler. 2013. Do good recipes need butter? Predicting user ratings of online recipes. In *Proceedings of the IJCAI Workshop on Cooking with Computers, Beijing, China*.

38. Kaizhong Zhang and Dennis Shasha. 1989. Simple fast algorithms for the editing distance between trees and related problems. *SIAM journal on computing* 18, 6 (1989), 1245–1262. DOI: http://dx.doi.org/10.1137/0218082

39. Kaizhong Zhang, Rick Statman, and Dennis Shasha. 1992. On the editing distance between unordered labeled trees. *Information processing letters* 42, 3 (1992), 133–139. DOI: http://dx.doi.org/10.1016/0020-0190(92)90136-j