A Systematic Survey on Instructional Text: From Representation Formats to Downstream NLP Tasks

ABDULFATTAH SAFA, KUIS AI Lab, Koç University, Türkiye
TAMTA KAPANADZE, Koç University, Türkiye
ARDA UZUNOĞLU, Johns Hopkins University, USA
GÖZDE GÜL ŞAHIN, KUIS AI Lab, Koç University, Türkiye

Recent advances in large language models have demonstrated promising capabilities in following simple instructions through instruction tuning. However, real-world tasks often involve complex, multi-step instructions that remain challenging for current NLP systems. Despite growing interest in this area, there lacks a comprehensive survey that systematically analyzes the landscape of complex instruction understanding and processing. Through a systematic review of the literature, we analyze available resources, representation schemes, and downstream tasks related to instructional text. Our study examines 177 papers, identifying trends, challenges, and opportunities in this emerging field. We provide AI/NLP researchers with essential background knowledge and a unified view of various approaches to complex instruction understanding, bridging gaps between different research directions and highlighting future research opportunities.

CCS Concepts: • **General and reference** → **Surveys and overviews**; • **Computing methodologies** → **Natural language processing**.

Additional Key Words and Phrases: Instructional Text, Procedural Text

ACM Reference Format:
Abdulfattah Safa, Tamta Kapanadze, Arda Uzunoğlu, and Gözde Gül Şahin. 2024. A Systematic Survey on Instructional Text: From Representation Formats to Downstream NLP Tasks. *ACM Comput. Surv.* 37, 4, Article 111 (August 2024), 41 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

The ability to program machines/computers with natural language, if it can be made successful, would fundamentally change the relationship between humans and computers. As of today, only a small percentage of humans (less than 1%) have the necessary skill set to program their computers or phones to perform new tasks. Machines and computers, however, are mostly viewed as preprogrammed devices with a fixed-set of skills. To move towards that goal, *programmable machines of the future* should be equipped with (at least) excellent instruction understanding capabilities.

With the recent advances in the Natural Language Processing (NLP) field, Large Language Model (LLM)s have demonstrated capacity on understanding instructions [121]. This is mostly achieved via "instruction tuning" that

Authors' addresses: Abdulfattah Safa, asafa22@ku.edu.tr, KUIS AI Lab, Koç University, Rumelifeneri, Sarıyer Rumeli Feneri Yolu, Sarıyer, İstanbul, Türkiye, 34450; Tamta Kapanadze, Tkapanadze23@ku.edu.tr, Koç University, Rumelifeneri, Sarıyer Rumeli Feneri Yolu, Sarıyer, İstanbul, Türkiye, 34450; Arda Uzunoğlu, auzunog1@jhu.edu, Johns Hopkins University, Baltimore, MD 21218, Baltimore, Maryland, USA; Gözde Gül Şahin, https://gglab-ku.github.io/, KUIS AI Lab, Koç University, Rumelifeneri, Sarıyer Rumeli Feneri Yolu, Sarıyer, İstanbul, Türkiye, 34450.

performs supervised finetuning on base language models with a large amount of instruction-response pairs [206]. These pairs are typically single sentences, describing a low-level task that can be performed in a single step. Furthermore, the ability to "follow instructions" of such instruction-tuned models is mostly evaluated on similar scenarios with simple instructions [54] and is an active area of research. However, real life tasks are a lot more complex, usually containing multiple instructions that affect each other in various ways. Hence, the next frontier in AI—in particular LLM—research will be to understand these complex instructions in a real-world-like setting.

Understanding such instructions requires—at minimum—understanding of events, the relations between events, their participants and the environment, and be able to perform multi-hop, common sense reasoning with such event knowledge. Various fields have investigated related subtopics, primarily as computational linguistics and NLP (e.g., event semantics, semantic parsing, script/scenario generation, common-sense reasoning, LLM agents), robotics (e.g., manipulation via instructions), business intelligence (e.g., process models), and computer vision (e.g., recipe understanding). Different fields use different naming conventions for complex instructions (e.g., process, procedure, task), different techniques to represent them (e.g., graph, workflow, business model, programming language etc...) and different venues to publish, which creates a high barrier to entry for new researchers.

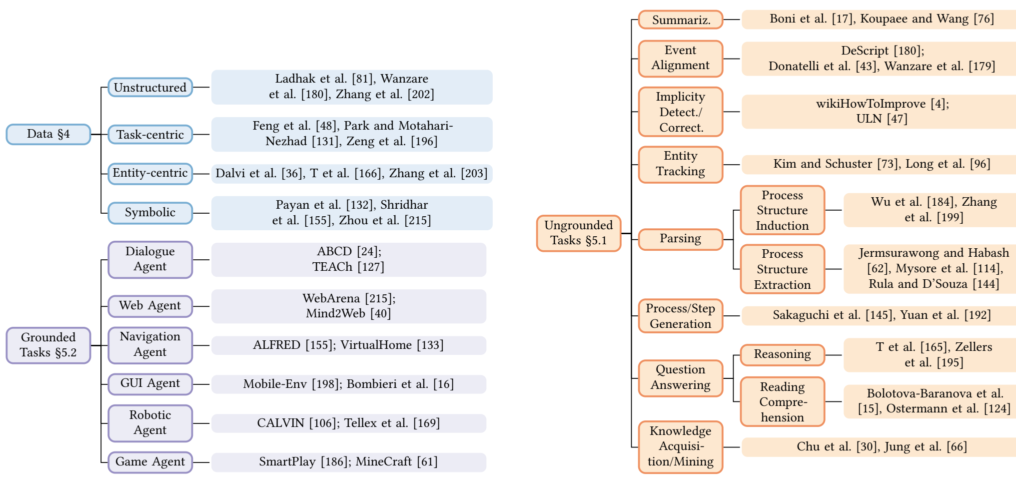


Fig. 1. Taxonomy of data representation types and related tasks in complex instructional text literature.

The main goal of this survey is to equip the AI—in particular NLP—researchers with the necessary background knowledge and provide guidance on the future challenges and oppurtunities for conducting research on complex instructions. In contrast to existing surveys on tangential areas (see §2), we provide a holistic presentation on **available resources** and **the range of tasks** related instructional text, making connections to other fields such as robotics, business intelligence, and computer vision. With this survey, we aim to answer the following research questions:

- **RQ1:** What are the most **common ways to represent** long-form, a.k.a., multistep, **instructional text** across different disciplines? What corpus/corpora (raw or structured) are available for various representation schemes?
- **RQ2:** Which **downstream tasks** are readily available on instructional text? How do they differ by means of domain, methods and evaluation metrics?
- **RQ3:** What are the **bibliographic properties** of the publications, e.g., how are they distributed temporally, geographically, and topic-wise?

- **RQ4:** What are the **common themes and challenges** in the literature concerning this topic?

In Sec. 2, we identify the related areas and existing surveys that complement this work. Next in Sec. 3, we define our PRISMA-based survey methodology [128] and provide insights on **RQ3**. Sections 4 and 5 aim to answer **RQ1** and **RQ2** respectively. We identify the common themes, and remaining challenges separately for each task in Sec. 5 to answer **RQ4**.

We present a taxonomy for the data representation types, and the range of tasks to guide the reader with Fig. 1.

2 RELATED WORK

We identify two closely related fields central to procedural language understanding[1]: i) event understanding, and ii) grounding.

Event Understanding Event-centric NLP field primarily deals with extracting event information from textual documents. Xiang and Wang [187] compile various approaches and challenges for event extraction from textual data, providing an overview of tasks, methods, and performance metrics. The survey categorizes techniques ranging from pattern matching to advanced machine learning models. Chen et al. [27] offer a comprehensive tutorial[2] on event-centric information extraction, prediction, and knowledge acquisition, focusing on event-centric tasks and methodologies. Finally, Li et al. [88] survey the deep learning techniques for event extraction, exploring sophisticated models that detect, categorize, and analyze events across various domains. Unlike these surveys, we focus on the higher-level relations between events, participants and their environments in long procedural text rather than extracting atomic event information.

Grounding Instructions In broad terms, grounding generally refers to a type of task that involves connecting language to some form of external knowledge or real-world context such as images, knowledge bases, robot arms and even operating systems. Ch et al. [23] discuss the evolution of the term "grounding" and draw connections to cognitive science. More recently, several surveys have been published on specific environments and grounding types. For instance, Cohen et al. [32] survey different meaning representations for grounding robotic language for navigation/manipulation tasks, while, Wang et al. [178] discuss various aspects of LLM agents on grounded tasks. In contrast, our objective is to survey the wide range of applications and environments to ground complex instructions rather than focusing on one environment or approach (e.g., LLMs).

To the best of our knowledge, there is no survey focusing on procedural text. The only resource is the tutorial by Zhang [201] that compiles a set of selected resources containing procedures, along with a set of selected applications. In contrast, we provide a **systematic** methodology and taxonomy covering a considerably wider range of representation types, resources and tasks for procedural text. We also extend the scope to other fields such as robotics, business intelligence, and computer vision, aiming to provide a unified perspective across disciplines. Finally, our survey is related to **scripts**—a sequence of events with multiple actors—and **planning**, i.e., generating a feasible (and hopefully minimal) sequence of steps to achieve a specific goal. Although we mention them where related (e.g., script generation, Web agents), we consider them to be outside the scope of this survey, and refer the readers to the classical book by Schank and Abelson [148].

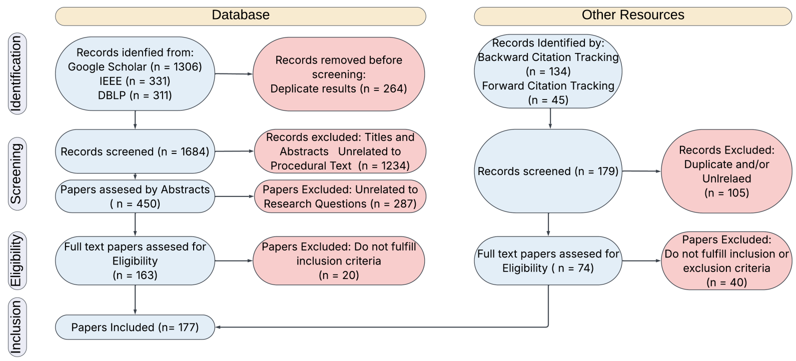


Fig. 2. Study protocol

3 METHODOLOGY

We have followed the guidelines given in the PRISMA statement [128, 129] to conduct this systematic review. The overview of our methodology is given in Fig. 2. It contains four steps: i) identification ii) screening iii) eligibility check and iv) inclusion, which are explained below.

3.1 Identification

We selected various libraries to ensure broad coverage of publications in fields relevant to this survey, such as AI and its subfields—particularly NLP, robotics, machine learning, and industrial engineering. DBLP indexes key computer science works, while IEEE Xplore provides a wider range of scientific content. Scholar primarily indexes peer-reviewed papers without field distinction. Given the rapid publication rate in NLP, we also considered notable non-peer-reviewed ArXiv papers. The review period spans 2010 to the present to capture relevant publications. We used the Google Scholar API from Serp API for our search, manually extracting other digital library entries. Since Google Scholar API lacks abstract retrieval, we sourced missing abstracts from Semantic Scholar. Full list of queried keywords that are based on research questions to capture relevant papers while maintaining a manageable volume are given in App. A.

Fig. 3 shows the distribution of retrieved publications across different keywords and databases after deduplication. An exact phrase match (EPM) searches for the exact phrase in papers, while exact word match (EWM) matches individual words. EPM is supported by Google Scholar and IEEE Xplore, whereas DBLP supports only EWM. To account for variations in terms (e.g.,"instruction" vs. "instructions"), we searched for both forms and combined the results. The AND operator was used to combine search terms for Google Scholar and IEEE Xplore, while DBLP was searched for each word individually with the AND operator. From all digital libraries, we retrieved up to a hundred search results per keyword returned by the database, as papers beyond were highly irrelevant. The execution of search queries in the four digital libraries returned 1,948 papers. After deduplicating the merged results from the databases, 1,684 papers remained. As expected, Google Scholar and IEEE Xplore return more results since they index a broader range of venues and domains, while DBLP only indexes computer science papers. We observe that the most number of papers are retrieved via the keyword "natural language instructions" since it covers various applications and methodologies. The

[1]We use the terms *complex instructions*, *procedural language*, and *script* interchangeably.
[2]Cognitive Computation Group

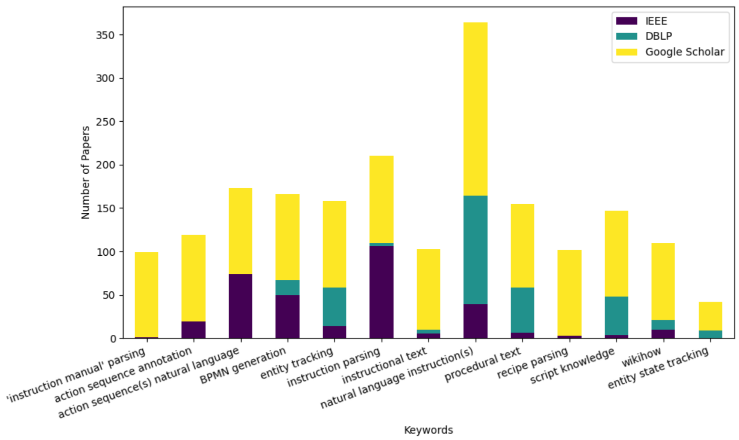


Fig. 3. Publications Per Keyword Per Database

least number of papers are retrieved for more specialized, narrow fields such as “entity state tracking” and “recipe parsing”.

3.2 Screening

Our screening process involved two phases. First, we manually reviewed titles and abstracts to remove irrelevant papers, starting with an initial pool of 1,684 papers. During this inspection, we noticed that papers relevant to our **RQ1** and **RQ2** often included keywords like “dataset,” “evaluation set,” “benchmark,” or “test set.” This makes sense, as the studies of interest typically introduce new datasets on instructional text or novel downstream tasks using existing resources, or sometimes both. These papers were then further filtered based on these keywords, reducing the count to 450.

3.3 Eligibility

Assessing the eligibility entailed manual inspection. Assigned researchers carefully examined the abstracts of the filtered papers to determine their alignment with our research inquiries. This manual review significantly narrowed down the selection to 163 papers.

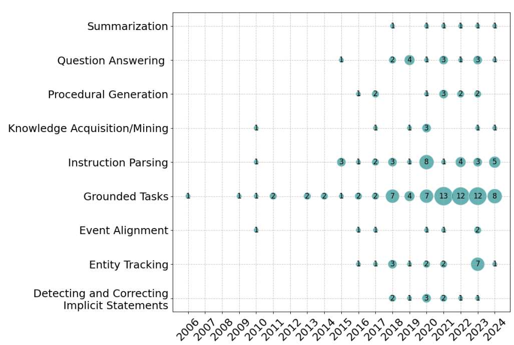
3.4 Inclusion

We curated a shared library of relevant papers for comprehensive full-text review, refining the selection to 143 papers. After retrieving the papers from digital libraries, we performed forward and backward citation tracking, identifying 34 additional papers, resulting in a final count of 177 papers. Once the list was finalized, we tagged the papers based on tasks, data representations, and research locations. This tagging helped us develop a taxonomy of representation types and downstream tasks (related to **RQ1** and **RQ2**) and visualize the information for **RQ3**.

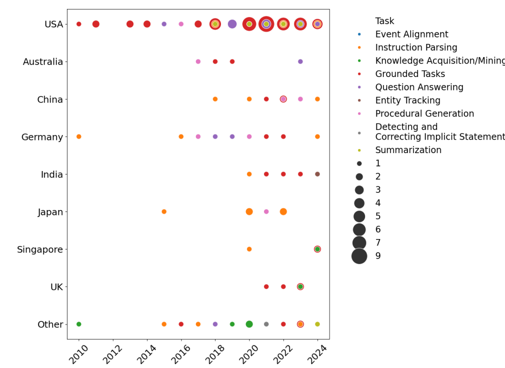
3.5 RQ3: Bibliographic Properties

To gain a comprehensive view of the instructional text processing research landscape, we visualized various data aspects. Fig. 4 shows the geographical distribution of publications, with a large number coming from renowned research hubs like Stanford, Seattle, New York City, and Beijing. This highlights the concentration of research efforts in urban centers known for their academic institutions. Notably, Stanford and Seattle lead in publication numbers, reflecting strong research output in NLP, especially in procedural text fields. This dominance, also seen in other AI subfields[20], underscores the lack of diversity in languages and research groups.

Fig. 4. Geographical distribution of publications



(a) Papers Per Task Distribution over Time



(b) Papers from each country Per Task Distribution over Time

Fig. 5. Comparison of Task Distribution and Country Distribution over Time

Fig. 5a shows the temporal distribution of tasks, illustrating a steady growth in *Summarization* and *Question Answering*, with a notable peak in the latter starting from 2018. Newer fields such as *Grounded Tasks* have seen a significant rise in interest since 2017, reflecting a shift towards more contextual and task-driven AI applications. The increasing trend in *Instruction Parsing* from 2018 onward indicates renewed interest in more structured task interpretation. Meanwhile, *Procedural Generation* has seen a consistent, albeit slower, development. Traditional areas such as *Entity Tracking* and *Event Alignment* have remained relatively steady, while areas like *Knowledge Acquisition/Mining* have gained more prominence in recent years. The growth from 2018 onwards, particularly in these emerging areas, suggests a trend toward more dynamic and interactive AI systems.

Fig. 4 and 5b illustrate the global scope of instructional text research, with certain countries excelling in specific areas. The USA pioneered Step Inference in 2006 and leads in *Grounded Tasks* and Instruction Parsing. Germany has focused on *Instruction Parsing* since 2016. China, active since 2018, has expanded into *Instruction Parsing* and *Entity Tracking*. Japan and India contribute to *Summarization* and *Procedural Generation*. Key research hubs include San Francisco, Berlin, and Beijing. These findings suggest potential collaborations, such as the USA and Germany on Parsing tasks or China and India on domain-specific applications, to advance global instructional text research.

4 DATA

Even though substantial amount of work for procedural text relies on **unstructured** corpora derived from web sources like WikiHow[3], the field is not short of structured, labeled datasets. We categorize the structured representation into three main categories: i) **event-centric** (see §4.2) ii) **entity-centric** (see §4.3) and iii) **symbolic** (see §4.4). Event-centric representations mostly ignore the properties of entities, and how certain events affect those. The entities are mostly referred to as "the argument" of the event—a.k.a., action—and are represented as free text. Instead, they put the events/tasks into focus and define a rich set of relations between them (e.g., subsequent, conditional, concurrent, parent/child etc...). On the other hand, entity-centric representations predefine intrinsic properties of entities (e.g., color, location) and track the state changes caused by extrinsic and intrinsic events. The events are mostly not tied to any static knowledge base or lexicon, and simply represented as free text. Finally, symbolic representations aim to transform the natural language instructions into a machine-readable format, such as first-order-logic, robot actions, and UI actions, so that they can be executed/performed on a specific environment. They are mostly used for grounding tasks (see §5).

4.1 Unstructured

The field has mainly used three techniques to create unstructured corpora to study procedural text: a) scraping/filtering available web resources, b) crowdsourcing, and c) synthetic data generation.

4.1.1 Web Corpora. The most popular way to construct large-scale corpora that can be repurposed (or annotated) for many downstream tasks has been scraping websites that contain how-to information. WikiHow has been exploited tremendously to study procedural, a.k.a., instructional text. Even though it is referred to as "unstructured", wikiHow articles follow strict writing style, possessing a certain file structure. For instance each article contains a goal *(e.g., How to destroy an old computer?)*, several methods to reach the goal *(e.g., destroying a computer entirely, destroying a computer for recycling)* and steps to execute the method *(e.g., 1. Wipe your hard drive, 2. Remove battery etc...)*. Furthermore, each article has a community Q&A and warning/tips sections for people following the article to seek help. WikiHow has a considerably high domain coverage—from make-up tips to writing indie songs. Articles are curated and edited by experts from all around the world in different languages. Nonetheless, similar to other web corpora, it is **predominantly in English**. Furthermore, most articles contain pictures that accompany the step description, that makes WikiHow suitable for multimodal studies.

In summary, WikiHow has been the most popular resource due to i) the strict style followed by the authors that enabled easy scraping, ii) high domain coverage (19 domains, and subdomains), iii) large scale (e.g., more than 235,000 articles), iv) multilinguality, v) multimodality and vi) high quality (i.e., expert curated articles). In addition to WikiHow, Instructables, ifixit and ehow several recipe websites such as Allrecipes, Cookpad, HaoDou, Food and Xiachufang have been used [42, 66, 94, 117, 189, 210, 211].

Web corpora are used in many shapes and forms, i.e., for pretraining or finetuning language models to enhance reasoning or planning capacities. It is also repurposed for a variety of downstream tasks, such as summarization [81],

[3]https://www.wikihow.com/

question answering [190, 195, 202], classification [131, 218], detecting implicit information in instructions [4], learning task hierarchy [217], generation (e.g., next step, all steps) [98, 122] and benchmarking procedural language understanding and planning abilities [172, 173] without any additional annotation layers. Even though recipes and WikiHow are mostly exploited, we find that **many resources such as troubleshooting websites from tech companies** (e.g., Canon **are overlooked** in the literature with some exceptions [51]. The reasons might be i) their small size (which is smaller) ii) the lack of a consistent structure that makes them hard to parse. Another **overlooked resources are local websites** (e.g., Turkish food recipes, which might be due to the lack of enough NLP experts or interest in local communities.

4.1.2 Crowdsourcing. Researchers have used **crowdsourcing to create smaller, but more targeted datasets**. For instance, DeScript [180] is a small corpus of scenarios on everyday tasks such as going for shopping, riding on a bus, getting a haircut and taking a bath. It contains a rather limited number of tasks (only 40), however, each one has been described step-by-step by 100 different crowdtaskers. Steps, a.k.a., event sequence descriptions, written by different taskers are later aligned with each other. Several other studies such as Task2Dial [162] and ABCD [25], employ crowdworkers to generate dialogue datasets grounded on instructional documents such as recipes and call center guidelines by assigning different roles to crowdworkers (e.g., call center employee, information giver on a certain recipe). It is also commonly used to annotate existing small corpora for a specific tasks. Such tasks are mostly related to extracting some form of information from instructional text, e.g., tools from fixing manuals [117], ingredients from recipes [211]. Due to the associated costs with crowdsourcing, this technique has been mostly used for generating scripts, grounded dialogues or to add small annotation layers to existing corpora. We find that, even though, for example, the corpus DeScript [180] and the alignments are publicly available, **such resources are overlooked and not used for related downstream tasks e.g., event paraphrasing or alignment** to the best of our knowledge, with some exceptions [179].

4.1.3 Synthetic data generation. Synthetic datasets and environments are also common for creating toy setups under simplifying assumptions. For example, Weston et al. [181] formulate 20 question answering tasks to evaluate different linguistic and reasoning abilities such as co-reference resolution, temporal and spatial reasoning. The stories are generated in a simulated world which is defined by manually written rules *(e.g., one should find food if hungry)* on a predefined set of entities with predefined attributes such as location and size. Similarly, Kim and Schuster [73] define a set of entities (e.g., book, hat etc..), properties (e.g., location) and a small set of actions (e.g., MOVE) to generate synthetic procedures about moving entities between different boxes via a simple Python script. It is then used to evaluate the entity state tracking abilities of large language models. Another research line develops instruction following game environments such as TextWorld [101] as a testbed for reinforcement learning agents to measure their planning, and exploration abilities. Similar to others, the environment is defined via a fixed set of rooms, entities, actions and entity properties, however, the game is interactive and played step-by-step.

Synthetic generation gives researchers the ability to **control** the complexity of the task and the environment. However, synthetic datasets mostly **lack the linguistic diversity, and long-tail, rare events that cannot be programmed**. In other research areas of NLP, it is common to post process the synthetic data via asking crowdsourcers to **paraphrase the synthetic text [73] to increase linguistic diversity. However we don't observe the same trend for procedural text literature, which could be a promising future direction**. Similar to web corpora, synthetic data is also commonly used for pretraining models. For instance, Shi et al. [153] create a large-scale synthetic dataset by randomly sampling the environment states and associated programs to pretrain a BART-large model, then evaluate on entity state tracking test splits.

4.1.4 Generation with LLMs. One of the latest trends in the field is to **exploit large language models to generate silver, large scale datasets**. Instructional text literature also **follows this trend**. For instance, Yuan et al. [192] create a constrained planning dataset (CoScript) of 55,000 procedures, e.g., procedures to make a cake for **diabetics**, by first generating with InstructGPT and then filtering with constraints.

4.2 Event-centric representation

Despite the differences in naming conventions, we use event, a.k.a., instruction, to refer to a step in a procedure that contains an action phrase to accomplish a subgoal *(e.g., Bake in the oven)*. We define action verb as the predicate (e.g., Bake), and the arguments of the predicate (e.g., it=the thing that is baked, oven=Location) as the entities. In Table 1, we demonstrate the diversity of representations from different angles, on a set of work that is manually selected to give the best overview. We determine the differences to be along the formats used for the **actions** and **entities**, and the **entity roles w.r.t the action**, and **event-event relations**. We note that, **business information literature** that employs Business Process and Model Notation (BPMN) **use a different terminology**. Here an event is represented as an empty circle to mark the start, middle and end of the process; while the event is called an action or an activity. Examples of event-centric representations, such as ‘baking a cake’ and the BPMN, can be found in App. B.

	Action	Entity	Action-Entity	Event-Event	OD
Wiki HG [48]	Text	Text	Ex, Es, Op	Suc, Ex	✓
MS [114]	Text	Tagged	Tagged	Suc	✗
SIMMR [62]	Tagged	Tagged	Input, Output	Suc, Con	✗
Action Graph [70]	Text+Implicit	Text, Food, Location	✗	✗	✗
APSI [199]	Text	✗	✗	Suc	✓
proScript [145]	Text	✗	✗	Suc, Con	✓
InScript [112]	Tagged	Tagged	✗	Suc	✗
Process Graph [131]	Text	Text	Actor, Time, Location	Suc, Opt, Con, PC, If-Else	✓
Process Model [135]	Text	Text	Object	Suc, Opt, Con	✓
BPMN [197]	Text	✗	Object	Suc, Ex, Con	✓

Table 1. Common task-centric representation schemes for procedural text. OD: Open domain, i.e., no domain specific tags. Ex: Exclusive, Es: Essential, Opt: Optional, Suc: Successive, Con: Concurrent, PC: Parent-Child, Pre: Precondition, Eff: Effect

4.2.1 Actions and Entities. We use event, action and task interchangeably following the previous work since the **granularity of the task representations varies greatly**. They refer to **a single step in the instruction** that needs to be performed to accomplish a goal. Furthermore, majority of the work make a simplifying assumption that a step/task contains only one action. For instance, Sakaguchi et al. [145] take the **full step** *(e.g., bake for the right amount of time)* as the task representation, while Zhang et al. [199] represent them as an **action phrase** that contains only one predicate and argument *(e.g., search car, apply loan)*. Another line of work [131, 135, 197] use the BPMN [75]—an industry standard—and similarly **represent a task as a predicate-argument pair**, where the argument is always an object. All the studies mentioned above use a free text representation, (denoted as “Text” in the Action and Entity columns of Table 1), sometimes limiting the number of actions to the most frequent ones. **This assumption might be problematic, especially when the articles contain different author styles** as in WikiHow, or DeScript. To address that, **several studies limit the domain and use domain-specific tags**. For instance Modi et al. [112] define **scenario-specific events** (e.g., apply-soap, undress, turn water on) and tag all the action phrases. Similarly, SIMMR [62] use a **predefined recipe-specific language** named MILK to represent the actions and ingredients. These are noted with “Tagged” in the Action and Entity columns of Table 1.

The studies that use free text representation mostly annotate the indices of the objects of interest (e.g., action verb, action phrase, argument etc...) [48, 114], or simply extract them with existing tools like semantic role labelers [131, 135, 199]. **Majority presume that all actions are explicitly stated in the text**. One exception is Kiddon et al. [70] that also include **implicit actions** as nodes in the graph. To the best of our knowledge, the same does not apply to entities, i.e., **event-centric representations mostly ignore the implicit entities**. For instance, imagine the phrase *“Mix a, b, c and then bake”*. The thing that is baked is the implicit, unmentioned mixture. Although there is a growing body of work on implicit instructions (see §5.1.3).

Entities are marked (e.g., for the position) in the text [48, 131, 135], tagged with predefined entity types [62, 112, 114], or simply ignored [135, 145]. Kiddon et al. [70] extract entities, however, only label a few important entity types, namely as food and location. Jermsurawong and Habash [62] represent the ingredients/entities and actions of the recipe as terminal and internal nodes respectively.

4.2.2 Action-Entity Relation. The role the entity plays in the action is crucial for task representation. Several studies [112, 114] define highly fine-grained entity types that also include information about the relation (e.g., Material_of). **Studies on scripts [112, 145, 197] mostly ignore such relations, while others do not have any consensus**. For instance Feng et al. [48] defines the arguments as exclusive, essential and optional. Imagine the phrase “Take a pen or a pencil”. Here, pen and pencil are defined as exclusive arguments of action “take”. Essential and optional arguments are also common in semantic schemas (e.g., Arg-0, Arg-TMP in PropBank). However the “actor” role rarely exists in procedural text, while the text is mostly imperative. Furthermore, **exclusive arguments are not part of the standard semantic schemes (e.g., PropBank, FrameNet, VerbNet), however, is important for procedural text**.

Despite the representation power and ample of choices in BPMN, the **surveyed studies mostly focus on the “object”, ignoring other essential information such as duration and location** with some exceptions [131]. Another approach is to represent such relations with unlabeled dependency links [62]. Here, the entities are given as input to the action. Unlike others, this enables linking entities with multiple actions, representing **implicit** entities such as mixtures with a link towards the root and representing entities and actions on the same level.

4.2.3 Event-Event Relation. One important feature that **differentiates the field of procedural language understanding from event extraction is the rich set of higher-level relations between tasks/events**. **Defining procedures with only successive events has the minimal representation power, but has been the default way**. This simplistic approach is problematic and is mostly addressed by business information management, and software engineering literature via defining and standardizing richer relations such as “exclusive”, “successive”, “concurrent” and “optional” (see Table 2 for examples). Some exceptions for NLP field are Feng et al. [48] and [145] including exclusive, and concurrent relations respectively. Another exception is Jermsurawong and Habash [62] that can also handle concurrent events implicitly in a dependency tree structure.

Parent-child relation for procedures is also understudied. The reasons are that procedures are mostly studied in isolation due to their challenging structure, and links to subtasks are usually nonexistent. Similarly, **conditional tasks are vastly ignored**. To study those, one needs alternative methods and the conditions to switch. Even though WikiHow contains different methods to accomplish a goal, which one to choose is not explicitly stated, and inferred by the readers. **Although there are exceptions, hierarchy and conditional processes are severely understudied according to our survey results**.

Relation	Event A	Event B	Explanation
Exclusive	Pay by cash	Pay by credit	Either/Or relation
Successive	Receive feedback	Process feedback	Event B happens after Event A
Concurrent	Book a flight	Book a hotel	Event A and Event B occurs simultaneously
Optional	Book a flight	Add travel insurance	Event B is optional
Parent Child	Write a paper	Do literature search	Event B is the child process of Event A
Conditional	Application is received	Application is confirmed	There is condition to move from Event A to Event B, e.g., GRE>80

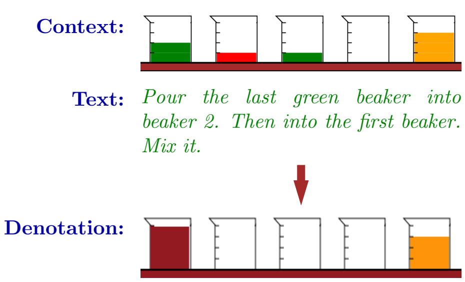
Table 2. Examples of event relations from the surveyed literature

4.3 Entity-centric representation

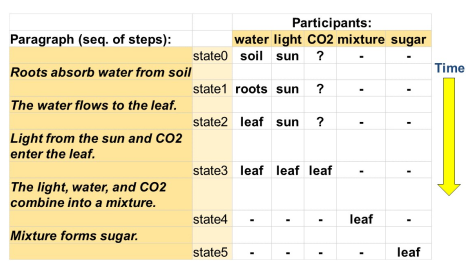
	Entity Types	#States	#Actions	#Entities	Avg #Steps	#Examples	OD
ProPara v1.0 [36]	Multiple	2	3	<10	>4	488 paragraphs	✗
OpenPI [166]	Unlimited	Unlimited	Unlimited	Unlimited	n.a	810 articles, 30K state changes	✓
NPN [18]	Multiple	6	384	Varying	n.a	65K recipes	✗
Boxes [73]	Multiple	2	3	Varying	12	2200 scenarios	✗
SCONE [96]	1	<3	<6	<6	5	4K scenarios	✗

Table 3. Common entity tracking datasets. OD: Open domain

In Table 3, we compile the common entity tracking studies and analyze them in several dimensions, such as domain coverage and the complexity (e.g., number of steps, number of actions). The datasets fall into two main categories: **simulated** and **crowdsourced**. Boxes [73], TextWorld [85] and SCONE [96] use simulation to easily keep track of object states and can be easily extended. For the simulated ones, the vocabulary is generally small—approximately 1,100 to 1,200 words, the text is artificially generated, hence usually uses a simple generative grammar and might contain spurious correlations. Kim and Schuster [73] attempt to increase the language diversity by including paraphrases. The second category of datasets include ProPara v1.0 [36], OpenPI [166] and NPN [18]; and use natural language text written by experts where the entities and states are annotated by crowdworkers. Opposite to simulated text, natural procedures are linguistically rich. Thus the same entity types or states can be referred to using different names. Recently released OpenPI-v2.0 [203] attempts to address this by clustering similar objects (e.g., spice and seasoning) together for a more uniform dataset. Fig. 6 shows two examples from simulated (left) and crowdsourced (right) datasets.



(a) An example from the ALCHEMY dataset. The content of each beaker at each step is visualized showing the results of the corresponding action.



(b) An example from ProPara dataset. Each filled row shows the existence and location of participants between each step.

Fig. 6. Examples of Entity Tracking Scenarios from the Alchemy [96] (simulated) and ProPara [36] (crowdsourced) datasets.

One important difference between the two approaches is the initial environment/world state. **Simulated** datasets explicitly include **statements about the initial state** of all entities (e.g., Box 1 is empty) **while real world procedures/processes do not** mention the initial states but required to be inferred. For instance, for the photosynthesis process, we assume (but not explicitly stated) that, the plants are buried in the ground, they have leaves, and there is sun above etc... There are several studies [31, 212] that aim to *inject* such common sense knowledge through external knowledge bases. For some of the domains such as cooking (e.g., Onion: NotCooked, Pan: Empty), default initial states can be easier to identify, while not so easy for other expert domains e.g., technical manuals [51]. On the other hand, **the goal is known beforehand in crowdsourced work** (e.g., how to fix Wifi adapter, produce energy by photosynthesis), although, the **end states are again not explicitly stated but can be inferred** (e.g., Adapter: fixed). Meanwhile, **simulated work are mostly random simulations without any clear goal**, i.e., things moving from box to box for no reason. But the **final states are again well-defined**. Another important difference is the "validity" of the scenarios. **Crowdsourced** procedures written by humans are **valid** by nature; however the simulated environments need to have certain measured for the validity of the generated scenario. For instance Li et al. [85] define the game as valid, if it is possible to reach the end goal from the initial state.

Since these studies focus on tracking the state changes of the entities, the number of entity types and possible states are domain-specific and limited to a small number. One exception is OpenPI [166] and its derivatives [185, 203] that consider all entity and state configurations in hundreds of WikiHow articles from various domains. The earliest work by Long et al. [96] define only a single entity type such as a beaker, bottle or a person. Later studies [18, 36] increase the number of entity types, i.e., Bosselut et al. [18] keep a short list of ingredients, where Dalvi et al. [36] focus on a small set of entities that are part of physical processes (e.g., photosynthesis) and Li et al. [85] incorporate text adventure game objects (e.g., player, chest).

For the sake of tractability, number of possible actions that can be applied to entities are limited to a small number (from 2 to 384). Most actions are generic—not domain dependent—such as create, remove, move; while NPN [18] also includes several domain-specific actions (e.g., bake, boil, cut). Entity properties, a.k.a., states, are again defined as a small set of fixed states (between 1 and 6). Although they are mostly domain independent (e.g., Location, Existence), some are domain-specific (e.g., Cookedness). States can be binary (edible, isOpen, exists etc...), categorical (shape, color) or free text (location). Note that location for simulated datasets are dominantly categorical (Box1, Beaker1 etc...).

Another factor that affects the complexity of the task is the number of steps that cause at least one state change in the set of entities. Both for the simulated and crowdsourced datasets, the researchers use a threshold for the minimum and maximum number of steps. For instance Dalvi et al. [36] sets the minimum to 4, while the simulated ones [73, 96] explicitly set the number of steps to 12 and 5 accordingly. In theory, the simulated datasets can be easily extended, however, currently they are restricted to a few thousand scenarios (2K-4K). On the other hand, crowdsourced datasets are generally smaller, mostly containing less than 1,000 articles. Finally, several datasets provide additional annotations for supporting tasks. For instance Dalvi et al. [36] provides an additional dependency explanation graph to include a cause-effect relation between the sentences; and OpenPI-v2.0 [203] introduces an additional "saliency score", i.e., importance in accomplishing the task for entities in the range of 1-5.

4.4 Symbolic representation

Entity- and event-centric representations allow for training and testing on various downstream tasks. However, there exists a discrepancy between downstream and real-world performance. There are multiple reasons: i) low representation power of downstream tasks, ii) data contamination and iii) offline evaluation. Another area of research addressed several problems via using symbolic languages to ground instructions to environments. This final category of work generally define—or use a predefined—formalism/language that can be automatically interpreted by an external symbolic interpreter and executed/performed on a simulation environment interactively. The type of the symbolic language depends on the environment it is interpreted. To name a few, Planning Domain Definition Language (PDDL) [108], Linear Temporal Logic (LTL) [16], First-Order-Language (FOL) [96] are commonly used in robotic environments for navigation and manipulation tasks. UI actions related to mobile and operating systems are used for environments such as AndroidHowTo [90], PixelHelp [90], WinHelp [19]. Custom-made, simplistic languages are designed mostly for 3D simulation and game environments like ALFRED [155], VirtualHome [133], JerichoWorld [2], FAVOR [87], TextWorld [101], SmartPlay [186]. In addition to environments focused on UI actions and operating systems, the procedural text is also used in the realm of code generation Payan et al. [132]. We discuss the grounded tasks and surveyed environments in more details in §5.2.

5 TASKS

Plenty of downstream tasks have been proposed to evaluate procedural language understanding in general. These tasks are naturally constrained by the data representation. We broadly categorize the procedural tasks into two categories: (1) grounded and (2) ungrounded. *Grounded* tasks refer to the tasks where the instruction is grounded on an external database, document or environment, whereas *ungrounded* tasks are solely defined on textual data. Please refer to Fig. 1 for the taxonomy and to App. C for the full list of tasks and papers.

5.1 Ungrounded Tasks

In this section, we discuss various ungrounded tasks related to instructional text.

5.1.1 Summarization. The trend focuses on extracting key instructions from either **single** [76, 81, 83] or **multiple** [17] procedural documents. This can be **extractive**, where key sentences are directly compiled from the source text, or **abstractive**, where new phrases are generated to summarize the content. WikiHow [76] and HowSumm [17], discussed in 4.1, are the main resources. For instance, Koupaee and Wang [76] generate human-crafted summaries for entire WikiHow articles, while Boni et al. [17] produce extractive summaries for external documents. Le and Luu [83] generate extractive summaries from WikiHow without explicit labels, selecting key sentences that approximate their abstractive summaries. DeChant and Bauer [39] utilize the ALFRED [155] dataset to generate abstractive summaries of robotic actions, leveraging video frames and detailed action descriptions to produce both brief, one-sentence summaries and detailed, step-by-step instructions for each task.

Methods and Evaluation. Early studies [76] employ sequence-to-sequence models to generate summaries, whereas later work [17, 39, 81, 83] **mostly fine-tune transformer-based architectures**, like T5 [39], mBART [81] and BART and Pegasus models [83]. The majority is in English following the general trend in summarization, except Ladhak et al. [81] which align WikiHow article steps across 18 different languages using the images that illustrate each step. There is **no consensus** among studies on **evaluation metrics**. Although ROUGE [39, 76, 81, 83] remains the predominant metric, BLEU [39], BERTScore [84] and METEOR [76] are also commonly used.

Challenges. The sequential nature of procedures requires maintaining the correct order of steps to avoid confusion. Models must also capture specific details, like measurements, as omissions can lead to inaccurate summaries [39]. Preserving key steps is crucial to ensure the summaries remain effective [83]. Generating consistent summaries is further complicated by varied writing styles and languages, particularly on platforms like WikiHow [76, 81]. Finally, models must track entity states throughout the process to maintain coherence in the summarized content. Unfortunately, it is difficult to gauge how well these challenges are addressed, as current evaluation metrics are not designed to detect whether key steps are included, measurements and step orders are correct, or the states of entities are given in the correct order. Future research should aim to **develop metrics specifically tailored** for these aspects.

5.1.2 Event Alignment. Event alignment typically refers to the process of identifying and matching corresponding actions or steps across different sets of unstructured instructions. An example on recipes [92] is given in Fig. 7. Event alignment is either one-to-one [115, 139, 179], a.k.a. *paraphrasing*, or one-to-many [43, 92]. The task is rich

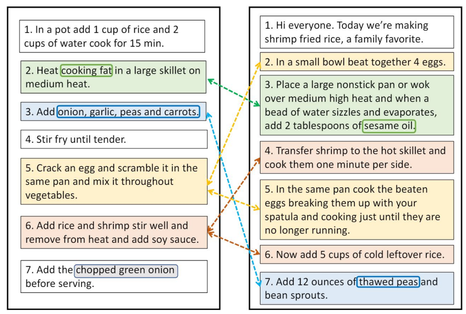


Fig. 7. Recipe Alignment Example from Lin et al. [92]

in domains thanks to the availability of raw procedural text. For example, several work [43, 92, 115] focus on the alignment of instructions within different cooking **recipes**—N et al. [115] use a dataset containing 1 million recipes from Recipe1M+ [102], RecipeNLG [12], and various how-to blog websites, while Lin et al. [92] introduce a dataset comprising 150K instruction pairwise alignments across 4,262 dishes. More recently, Donatelli et al. [43] provide the Aligned Recipe Actions (ARA) corpus comprising recipes for 10 different dishes, originally sourced from the Lin et al. [92]. Apart from recipes, a considerable amount of work [139, 179, 180] focus on **everyday tasks** and compile novel resources for Event Sequence Description (ESD) such as DeSCRIPT, SMILE and OMICS[4].

Methods and Evaluation. All the aforementioned datasets except from the DeSCRIPT [180] and the ARA [43], **do not contain any hand-annotated alignments**, hence, the studies mostly use unsupervised learning algorithms. The models process procedural descriptions such as ESDs [139, 179, 180] or natural language instructions [43, 92, 115]. Then they align actions/events across different descriptions [43, 92, 115] or generate script structures that capture temporal and semantic relationships between events [139, 179, 180]. Other works align events across different modalities, such as text and videos [78, 104] which are outside the scope of this survey. Wanzare et al. [179] use an Affinity Propagation (AP) semi-supervised clustering algorithm driven by semantic and positional similarities to group event descriptions into paraphrase sets that represent event types. N et al. [115] use BERT and Donatelli et al. [43] use LSTM networks to encode action descriptions and align them across different recipes. Finally, Lin et al. [92] use the unsupervised Hidden Markov Model (HMM) to generate pairwise alignments between recipes. To evaluate their unsupervised models, researchers adopted two distinct approaches. The first approach involves annotating part of the dataset for evaluation, where portions of the data are manually labeled to serve as golden data [139, 179]. The second approach uses a separate

[4]http://openmind.hri-us.com/

dataset for evaluation, independently annotated to assess model performance [43, 92]. Accuracy, precision, recall, and F1-score are commonly used to measure how well predicted alignments match the golden data [43, 92, 115, 139, 179].

Challenges. The evaluation results indicate substantial room for improvement, as accuracy scores on most datasets fall below an F1 score of 73. The biggest bottleneck for progress is the lack of large datasets with golden alignments. However, manually aligning actions is labor-intensive. A potential solution is to generate synthetic alignment datasets by leveraging large language models (LLMs). Furthermore, it remains unknown how effective LLMs will be for this task. Hence future work could explore integrating them for event alignment.

5.1.3 Detecting and Correcting Implicit Instructions. An intriguing but understudied challenge in procedural text is the presence of implicit or unclear statements. Anthonio et al. [4] address this by constructing a corpus based on revisions to WikiHow articles, focusing on whether sentences need clarification (e.g., adding missing pronouns or quantifiers). They also introduce a shared task [143] to promote research on implicit language, and initiated a workshop series [5]. Zeng et al. [197] tackle the challenge of *repairing procedural text* by using external activity templates to fill in missing information. Zhang et al. [210] develop a dataset of Chinese recipes from Haodou recipe website, where missing entities are supplemented using both text and images. Bisk et al. [13] introduce the KIDSCOOK dataset, pairing cooking instructions with hierarchical sub-steps, and proposed a task to infer hierarchical relations between instructions and sub-steps. Similarly, Feng et al. [47] create the R2R-ULN dataset, which involves guiding an agent through environments with underspecified navigation instructions. The works in this task fall into two main categories: (1) **detecting and correcting implicit or missing information**, and (2) **clarifying procedural instructions**. The first category focuses on identifying and completing missing or implicit details in procedural texts, using inputs that often include underspecified texts, sometimes with images or process models. The output is a refined text, a simulated representation, or a predicted completion, where missing actions, entities, or arguments are corrected, inferred, or predicted [29, 31, 197, 210]. The second category aims to improve clarity and structure in instructional texts by refining vague or ambiguous instructions. The output is a clearer, explicit, and more actionable set of instructions or a prediction of where clarification is needed to achieve this [4, 13, 47, 143, 200].

Methods and Evaluation. Early models often rely on symbolic approaches like Semantic Role Labeling (SRL) and VerbNet to infer missing information in procedural texts [31]. Recent studies predominantly employ pretrained language models [143], or vision-and-language transformers [47, 200] depending on the nature of the task. For evaluation, Precision, Recall and F1-score are commonly used across both correcting and clarification tasks [29, 31, 68, 212], while accuracy [29, 197, 210] is more frequently used for the correcting one.

Challenges. Current models, which are often domain-specific (e.g., cooking, navigation), struggle with dynamic, multi-context reasoning and implicit temporal and causal relationships [47, 200]. Tasks like troubleshooting, and requiring corrective actions for unforeseen events, are seldom addressed. Future research could explore moving from simple to more ambiguous or evolving instructions, improving model robustness in real-world applications.

5.1.4 Entity State Tracking. In §4.3, we present a detailed analysis of how entities, procedures, and their relationships are represented in the literature. Entity (state) tracking task is defined on these representations and focuses on tracking entities (e.g., ingredients, tools) and their properties (e.g., color, location) and interactions through each step. Several

[5] https://unimplicit.github.io/
https://unimplicit2022.github.io/

datasets have been created, each focusing on different domains, e.g., hardware and software setup [51], WikiHow [166, 204] and cooking recipes [141]. Entity tracking is mostly the primary task [18, 31, 36, 37, 73, 85, 96, 204, 212], or serves as an intermediate step for other event related tasks [29, 68, 115, 116, 141, 205]. For the first category, Dalvi et al. [36] and Clark et al. [31] leverage scientific procedural text, i.e., Process Paragraph (ProPara), while Zhang et al. [212] and Bosselut et al. [18] use Recipes to model entity movements and state changes across multiple steps. These datasets use predefined set of entities, properties and actions, offering a more controlled setup. More recently, open-domain tracking datasets [37, 166] are introduced where entities and properties need to be **extracted or inferred**, which provides a more challenging setup. On the other hand, several others [73, 96] build simple synthetic datasets for a predefined set of actions and states. For the second category, entity tracking plays a supporting role for other tasks like causal reasoning, coreference resolution, and multi-task learning. Tracking how entities change state helps identify relationships between actions and their effects [37] or predict hypothetical event outcomes [205] for causal reasoning. Tracking also helps resolving coreferences [141] and identifying implicit arguments [29]. Furthermore, tracking supports action alignment, step prediction, and understanding event-driven state changes [68, 115, 116].

Methods and Evaluation. The approaches for tracking are more diverse compared to other tasks. For instance, using a tailored neural network, e.g., Neural Process Networks [18] is common. As in other tasks, pretrained models [115, 212] still dominate the field due to their large context length with various approaches. For instance, N et al. [116] and Shi et al. [153] explore different pretraining strategies and show that pretraining on similar datasets or with different order improves the entity tracking performance. However, recent models also integrate symbolic reasoning using resources such as ConceptNet and VerbNet into neural models [68], which highlights the difficulty of the task. Accuracy is the most commonly used metric for assessing the correctness of model predictions [29, 36, 96, 116, 212]. Precision and recall are also frequently used in various models [51, 68], often alongside F1 [18, 51, 68, 73, 166, 204, 205, 212]. Additionally, in open-vocabulary entity tracking tasks, BLEU, ROUGE, and BERTScore are used since these tasks involve generating flexible, context-dependent descriptions of state changes, making generation metrics more suitable for evaluation [166, 204].

Challenges. Open-domain entity tracking has been proven to be challenging even for state-of-the-art models, especially with complex tasks [204]. LLMs struggle in dynamic scenarios with multiple state changes across steps, particularly with varied entities and complex transitions [51, 73]. While simple properties like “isOpened” are often tracked, nuanced ones like “isSettingsChanged” remain difficult [51]. Neurosymbolic models, incorporating event semantics is the latest trend, showing improvement by capturing event dynamics [68]. Nonetheless, performance still degrades with task complexity, especially with sparse property data [36, 212]. These challenges highlight the need for advanced approaches, like **neurosymbolic models** and **refined metrics**, to capture long-range dependencies and event-driven changes [18, 51, 204].

5.1.5 Instruction Parsing. Parsing natural language instructions into structured data representations such as trees, graphs, and BPMN is one of the most crucial steps to be able to run/execute them. There are two main approaches: 1) Supervised parsing, where the model uses labeled data to train on **Process Structure Extraction (PSE)**, and 2) Semi-supervised or unsupervised parsing, referred to as **Process Structure Induction (PSI)**, explained in details below.

Process Structure Extraction. In PSE, unstructured procedural text is parsed into structured formats like trees, graphs, or BPMNs. PSE plays a key role in domains like **business process management** by extracting actions, actors, and their relationships within task sequences. Fig. 8 shows an illustration of the PSE problem from Qian et al. [135].

This task is rich with diverse domains like **cooking recipes** [11, 42, 48, 70, 100, 108, 130, 135, 175, 211], **household and everyday tasks** [48, 77, 108, 144, 207, 208, 217], **technical and scientific procedures** [108, 114, 144], and **general instructions** [59]. Diwan et al. [42], Feng et al. [48] and Wang et al. [175] extract the structure (ingredients, cooking techniques, and utensils) of recipes from the RecipeDB dataset [8], cookingtutorials.com, and the Recipe1M dataset [146], respectively. On the other hand, several other works generate new datasets. Mysore et al. [114] generate 230 synthesis procedures (discrete process steps taken to synthesize the target material) with labeled graphs that express the semantics of the synthesis sentences (actions, materials involved, conditions, etc.). Zhang et al. [208] collect the MSComplexTasks dataset that contains complex tasks from Wunderlist[6] with their sub-task graph, while Zhang et al. [211] create the MIAIS dataset (see §4). Graphs and trees are the most commonly used due to their relations with workflow representations [100, 114, 130, 135, 144, 175, 208, 211, 217]. Other works produce well-structured key-value pairs [42], PDDL [108, 207], or lists of structured sequential actions [48].

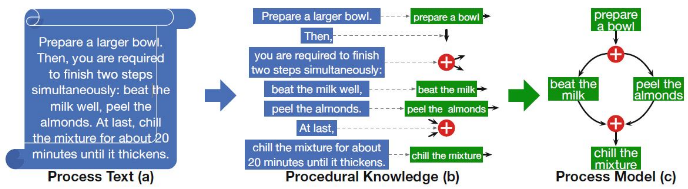


Fig. 8. Illustration of The PME Problem [135]

Methods and Evaluation. Early PSE approaches rely on basic NLP techniques (NER, POS Tagging, Clustering) for structuring recipes [42], followed by neural architectures like Bi-LSTM and CNN for process representation [131, 135]. Deep Reinforcement Learning (DRL) expands capabilities in action sequence extraction and game level generation [48, 79]. Semantic and syntactic parsing advance the field through applications in Trigger Action Program (TAP)s and flow graphs [11, 59, 114]. The emergence of transformer models, particularly BERT, MSBART, SBERT, and DEBERTA, enable more sophisticated approaches to flow graph extraction [11], multi-modal workflows [130], and task decomposition [208, 217]. Recent work leverages LLMs like GPT-4 for PDDL generation, game level creation, and business process modeling [77, 79, 144, 207]. Not surprisingly, precision, recall and F1 (based on the extracted action names and arguments) are used to evaluate these models [11, 42, 48, 100, 108, 130, 135, 140, 217]. Some approaches use more fine-grained metrics, e.g., accuracy of the extracted process compared to the gold ones (how accurate the model is in predicting the connections and nodes in the graphs) and success rate [77, 135, 207, 208], BERTScore [208] and ROUGE [144, 208] for evaluating the similarity between the extracted process represented as graph or RDF and the gold ones represented as text—after transforming the graph into a linear sequence of subtasks.

Challenges. The scores for majority of the existing benchmarks (i.e., recipes and technical manuals) are over 92.9% F1 score [108] and effectively generating BPMN model quality guarantees and soundness constraints [77]. However, substantial challenges persist. For instance, LLMs still struggle with correctly identifying sequences of steps and their dependencies [207]. Most importantly, the field lacks large, standardized datasets for training and evaluation [108], as creating annotated process datasets requires significant manual effort.

[6] https://en.wikipedia.org/wiki/Wunderlist

Process Structure Induction. Induction uses unsupervised or semi-supervised techniques to *induce the latent* structure of event sequences. Here, the goal is to automatically identify and align similar event descriptions across multiple instances of a scenario/procedure, clustering them into paraphrase sets that represent the event types (nodes) within the procedure. Most common scripts are from everyday tasks, such as baking a cake, grocery shopping, or booking a flight, described as ESDs [139, 179, 180], or with text narratives describing the process [94, 144, 184]. Additionally, some approaches incorporate structured data, such as graphs, to generalize subprocesses into more generic ones [199]. In addition to collecting processes from resources like WikiHow [184, 199] and public manuals [144, 184], other works [139, 179] use existing datasets, such as the OMICS corpus and the MiniWoB dataset [154]. Similar to the process extraction task, graph-based structures dominate the field [139, 179, 180, 184, 199]. Several other works generate data in other formats, such as sequences of low-level actions (e.g., atomic or specific user actions, such as clicking buttons) [94] or ontologies [144].

Methods and Evaluation. Traditional unsupervised and semi-supervised methods, such as Multiple Sequence Alignment (MSA) and clustering algorithms are used to group and align event descriptions based on semantic and positional similarities to construct temporal script graphs [139, 179, 180]. Heuristic approaches, including coreference resolution and keyword detection, are also used to generate weak labels while fine-tuning models such as RoBERTa, to link actions with their conditions [184]. Sequence-to-sequence models and frameworks like APSI identify analogies across processes, helping models to generate sub-event sequences for new tasks [199]. Recently, GPT-4 has enhanced procedural text mining, enabling the model to structure procedural elements in zero-shot settings, representing them in both textual and ontological formats [144]. Since unsupervised or weakly supervised models often lack fully annotated process structures, evaluations adapt based on available resources. When golden labels are available [199], predicted sequences are compared to expected processes using metrics like E-ROUGE to assess structural and temporal accuracy. In the absence of full annotations, a subset is annotated via crowdsourcing, and traditional metrics e.g., precision, recall [179], and ROUGE variants [144] are used for evaluation. ROUGE is particularly useful when processes are expressed as text, evaluating how well the extracted process structure—hierarchically ordered steps and sub-steps—matches the gold-standard sequences.

Challenges. Process structure induction remains an unresolved task, with models consistently underperforming compared to human benchmarks by over 20 points in key metrics [184, 199]. Challenges include complex sub-event prediction, conceptualization, and contextual understanding of instructional texts [184]. Models also struggle with predicting dependencies like preconditions and postconditions, as well as handling ambiguous or incomplete instructions [144]. Future research could focus on enhancing causal and temporal reasoning, improving data augmentation for imbalanced datasets, and developing more comprehensive evaluation frameworks tailored to procedural text processing.

5.1.6 Procedural Generation. Procedural generation, is also referred to as script construction [98], Similar to other tasks, cooking recipes [71, 95, 97, 123], and everyday tasks [98, 112, 122, 123, 145, 163, 192] like "baking a cake" or "fueling a car" from WikiHow and other sources, are the most common domains. While most datasets are derived from existing resources [71, 98, 122, 123, 142, 163], several are created specifically for this task, such as proScript [145], CoScript [192], WIKIPLAN and RECIPEPLAN [97], InScript [112] and XIACHUFANG [95] datasets. The output of the models vary, ranging from sequences of natural language steps [71, 95, 98, 142, 145, 163, 192], to multimodal procedural plans pairing text and images [97, 123] and to graph in DOT language—graph description language detailing event sequences [145]. Input can also be diverse; such as descriptions of scenarios with ordered [122] or partially ordered [145] events, or just the procedure goal [97, 98], with additional details like a list of ingredients [71], constraints [192], or user preferences [95, 163]. Other input formats can include image sequences or pairs of "before" and "after" images [142].

Methods and Evaluation. Sequence-to-Sequence models, employing advanced RNN variants such as LSTM and GRU, are exemplified Nguyen et al. [122], Nishimura et al. [123], Rojowiec et al. [142], where Sun et al. [163] further enhance a Sequence-to-Sequence framework by integrating task-specific concepts through a concept prompting framework. More recent work leverage pretrained language models like T5 and GPT-3 [97, 98, 145, 192]. For evaluation, typical generation metrics such as BLEU [71, 95, 122, 142, 192], ROUGE [123, 142, 163, 192], BERTScore [95, 98, 163, 192] and METEOR [71, 97, 142] are used.

Challenges. Procedure generation faces intrinsic challenges that affect the performance of the state-of-the-art (SOTA) models, which still lag behind human baseline [95, 145]. Models struggle with constructing coherent, logically ordered scripts, especially in diverse, open-ended scenarios [95, 192], and the inclusion of visual data exacerbates this challenge, as its quality and relevance are crucial [123]. Integrating concept knowledge from structured sources, like Task Concept Dictionaries, can sometimes introduce contextual noise, which may lead to less relevant or useful outputs [163]. A notable challenge arises when adapting procedural knowledge to counterfactual conditions, such as changing primary ingredients in recipes while maintaining coherence [95]. Automatic metrics fall short in capturing the complexity of generated scripts, making human evaluation essential, despite its limitations. This evaluation bottleneck highlights the need for more robust frameworks to better assess model performance [95, 98, 123, 163, 192].

5.1.7 Question Answering. QA for procedural text typically requires an understanding of action sequences, causal relationships between steps, and implicit knowledge of the process described. Numerous datasets have been developed to support this task. For instance, datasets sourced from **WikiHow** include WikiHowQA [15], the WikiHow dataset [218], PARADISE [172], and other WikiHow variants [173, 190, 202]. **Everyday task** datasets include MCScript [124] and its successor, MCScript2.0 [125], along with HellaSwag [195]. Additional datasets address other domains, such as RecipeQA [189] for **recipes** and **social interactions** [9]. QA in procedural text either focuses on reasoning or reading comprehension. **Reading comprehension** benchmarks [15, 189, 218] mostly rely on retrieving correct information directly from the text without requiring deeper inferences. In contrast, **reasoning-based** benchmarks [9, 124, 125, 165, 176, 181, 190, 195] require models to go beyond merely extracting information, such as understanding causal relationships, making inferences (e.g., determining the effect of a missing step in a cooking recipe), predicting outcomes (e.g., what would happen if a particular process in a machine were altered), or applying commonsense knowledge to answer questions. Further categorization can be made based on whether the approach requires answering from a single document or multiple documents. **Single-document** retrieval involves tasks where models must derive answers from a single text or data source [124, 125, 165, 176, 189, 195, 209, 218], while **multiple-document** retrieval [9, 15, 190, 202] requires synthesizing information from various documents to answer questions.

Methods and Evaluation. The **reading comprehension** approaches usually use LSTM-based architectures [189, 218] or fine-tune transformer-based models [15] like BERT, BART and RoBERTa. In contrast, **reasoning** papers fall into three categories: *procedural and causal*, *commonsense*, and *strategic reasoning and planning*. Procedural/causal and commonsense reasoning QAs also employ pretrained models [165], however, there is some variation. For instance, triplet networks [190] are used for multi-step process management and sequential goal-oriented tasks; while commonsense reasoning models employ additional knowledge bases such as ConceptNet and ATOMIC [9]. Strategic reasoning leverages reinforcement learning, notably PPO [209], for multi-turn decision-making and entity deduction. There are mostly no specialized metrics (with exceptions), hence typical QA evaluation metrics are used: e.g., ROUGE, BLEU, and BERTScore [15] for abstractive, accuracy for multiple-choice QA [124, 125, 165, 189, 195]. Additional metrics like success rate, recall at k, and exact match are used in specific contexts; for example, success rate in Zhang et al. [209] evaluates entity identification over turns, and recall at k in Yang et al. [190] assesses whether the correct step appears within top-k predictions in sequential tasks.

Challenges. Procedural QA remains unsolved, with models trailing human performance by over 10 percent [124, 165, 190, 202]. Some of the existing datasets [189, 195] focus on correct task execution, overlooking error-prone scenarios which are more realistic. To improve robustness, future work could explore creating datasets with such scenarios, and develop models that would identify and correct errors. Multi-step reasoning is still challenging even for state-of-the-art models, as systems often handle isolated steps without considering the overall action sequence.

5.1.8 Knowledge Acquisition/Mining. This task focuses on constructing comprehensive semantic knowledge structures by identifying and formalizing entities, actions, relationships, and processes from unstructured sources (e.g., natural language instructions). The goal is to enable AI systems to better reason, plan, and execute procedural tasks. Unlike process extraction and parsing, which focus on identifying sequential steps or syntactic structures, knowledge acquisition emphasizes building rich contextual understanding and relationships between concepts. Again WikiHow serves as the primary source [28, 30, 67, 108, 150, 160], in addition to scenarios and instruction manuals [6, 108]. Knowledge acquisition encompasses two main areas: 1) **Task Mining** extracts procedural knowledge (tasks, sub-tasks, methods) from instructional texts to support task-oriented conversational agents, recommendation systems, and knowledge bases [28, 30, 67, 150], and 2) **Domain Acquisition** formalizes knowledge from natural language task descriptions into structured representations (e.g., PDDL models) for automated planning systems. These can be either static models [108, 160] or dynamically adaptable ones that acquire new knowledge during execution [6].

Methods and Evaluation. Earlier methods use rule-based systems, POS tagging, embedding-based clustering and syntactic parsing to cluster task steps and create a hierarchical knowledge base [30], to adapt the domain model dynamically [6], and to generate PDDL models from instructions [160]. Later, pretrained models like variants of BERT and LongFormer are used to encode task descriptions and step information to identify relevant attributes (slots) for creating a task knowledge base [150]; combined with SRL [28] to capture context and produce a semantically typed task graph; and DRL [108] to extract action sequences and enhance the model's ability to generalize across various instructional data types. Although the approaches in both Task Mining and Domain Acquisition are largely unsupervised or weakly supervised, the evaluation of these models often relies on various forms of ground truth data and indirect performance metrics. In **Task Mining**, models are evaluated by comparing the extracted task structures (steps, actions, objects) against human-annotated datasets [67] or using crowdsourced evaluations [30], with accuracy and recall as primary metrics. In **Domain Acquisition**, models are evaluated based on their ability to generate valid PDDL models that can be executed by planners (though PDDL itself is not executable and must be processed by a planner to produce action sequences). For example, Steinert and Meneguzzi [160] evaluate success by testing if the generated PDDL plans achieve their objectives in simulated environments, measuring the accuracy and success rates of these plans. In more dynamic systems like Babli and Onaindía [6], evaluation focuses on the agent's ability to adapt to new objects or actions by updating its planning ontology in real time, with success measured through correct plan adjustments.

Challenges. The fields of task mining and domain acquisition have advanced yet remain unsolved. Steinert and Meneguzzi [160] report a plan execution success rate of 0.5, and Chen et al. [28] achieve 0.80 Recall in action-object prediction using weak supervision. Sen et al. [150] find that limited annotated data hinders automated task extraction, impacting reliability. The scarcity of *annotated data* constrains scalability, and current reliance on *simulators* and human input often falls short of real-world complexity, reducing generalizability. Improved automatic metrics and better-annotated datasets could provide more scalable, reliable evaluations and reduce costly human oversight.

5.2 Grounded Tasks

The final category of tasks we consider is **grounded tasks**, which aim to perform instructions that can be represented as a sequence of actions in a symbolic language on **simulated environments** (e.g., mobile phone, 3D simulation), external **knowledge resources** (e.g., querying a database, searching web) or in **structured or unstructured documents** through multi-turn task-oriented dialogues. These tasks typically require understanding and utilizing contextual information, and domain-specific knowledge to effectively perform the tasks within specific domains, such as web, mobile, game, robotics, and navigation. We build a taxonomy of environments that are commonly used in literature, namely as Web (§5.2.1), Navigation/Household (§5.2.2), Robotic (§5.2.3), Game (§5.2.4), and GUI (§5.2.5); and provide the list of papers for each environment in Appendix D.

Meanwhile, **grounded task-oriented dialogue systems** build upon the principles of grounded tasks by enabling interactive, language-based guidance within various environments. These systems ground their responses in external sources such as documents [45, 46, 162] or real-time environmental data [120, 127]. For instance, they can guide users to navigate in dynamic virtual environments [120, 127], or in troubleshooting and household management through static resources [45, 46, 162]. Recent advancements in NLP, including LLMs and Retrieval-Augmented Generation (RAG) frameworks, have expanded the capabilities of these systems, making them essential in fields such as customer support and healthcare [45, 136]. We discuss them in details in §5.2.7.

5.2.1 Web Environments. These environments mimic real-world web interactions, tasking agents with activities from information seeking [41, 215] to e-commerce [94, 154, 191]. It is generally defined as partially observable Markov decision processes(POMDP) [219], where the state is the webpage content or past actions[40, 215]; and actions are clicking, typing, and sending queries [40, 94, 215].

Methods and Evaluation. A **Web agent** performs step-by-step actions in these Web environments, interacting with elements such as DOM Trees, screenshots, or forms [154, 188, 215]. These agents are increasingly used to benchmark language models as web agents [215] or reinforcement learning environments [94, 154]. Language models such as WebGPT [119], GPT-4 [40], and others built on BERT [188] are common, but reinforcement learning techniques are also employed [94, 154, 191]. Reward functions, like those in WOB [154] and miniWOB++[94], use binary success/failure scores, while WebShop[191] evaluates actions differently. User tasks and annotation granularity also vary. WebNav [105] and DialQueries [105] focus on command/navigation instructions and real queries, while Mind2Web [40] annotates tasks with actions and HTML snippets. MiniWoB [94, 154] handles synthetic and real-world tasks, with structured annotations for user interactions. E-commerce environments like WebShop [191] and WebArena [215] annotate actions like purchasing. These environments often involve symbolic annotations mapping natural language to web actions [40, 52, 105, 188]. Evaluation metrics include exact and fuzzy match, and success rates [40, 154, 215].

Challenges. None of the web environments are fully solved; for example, the state-of-the-art Web Agent for WebArena [215] has a success rate of 57% [7] as of October 2024. Moreover, the complexity of Web environments vary

[7] X-WebArena-Leaderboard

drastically. Some environments allow more realistic grounding, such as WebShop [191] and environments in Gur et al. [52], Mazumder and Riva [105], Nakano et al. [119] handling **multiple pages**, whereas others like WOB [154] and miniWOB++[94] focus on single-page interactions. The WebArena environment[215] offers **handling different tabs** and more complex challenges such as unachievable tasks and noisy environments [94, 154, 215]. Agents navigate these tasks with high-level instructions, requiring them to **independently plan** and **execute** actions. Future work should explore cross-environment generalization across multiple settings [40, 191, 215] that would explore different capabilities of Web agents.

5.2.2 2D/3D Navigation/Household Environment. This environment involves simulated settings where autonomous agents interpret natural language instructions to execute navigation tasks [152]. VirtualHome [133] is one such environment, characterized by spatial layouts agents must navigate, often using maps or virtual simulations[152]. Common tasks are household assistance [133, 183] and virtual navigation [137], with some environments focused solely on navigation [3, 5, 89, 99, 134, 194] and others on complex, multi-step tasks in household settings [38, 47, 87, 110, 133, 155, 216]. Environments vary in structure and focus: REVE-CE [89] and REVERIE [134] emphasize visual navigation, while others use linguistic-driven navigation, guiding agents with natural language instructions [5, 174, 194]. Meanwhile, VirtualHome [133, 183] and FAVOR [87] involve object manipulation and rearrangement through detailed instructions and AR-driven systems. Some, like ALFRED [155], integrate both visual and linguistic inputs to enhance navigation and interaction capabilities [111, 216]. The studies differ in terms of **observability**, with some, like VirtualHome [133], offering **complete map awareness**, while others, like ALFRED [155], present **partial observability** where agents must explore and adapt [3, 38, 47, 89, 99, 134, 174]. This dynamic adaptation makes tasks more challenging as agents gather new information and encounter obstacles [89, 134, 155]. The **action space is often limited** and repeated across environments, involving tasks like camera adjustments, moving through spaces, and manipulating objects [3, 87, 89, 126, 133, 134, 174]. Task complexity in environments like REVERIE [134] and REVE-CE [89] is increased by adding object identification or manipulation [87, 133, 155].

Methods and Evaluation. Earlier agents, such as in Matterport3D Simulator[3] and REVERIE [134], often rely on sequence-to-sequence models with attention mechanisms and reinforcement learning to complete tasks. Recent research explores LLMs for simulated navigation tasks (e.g., LLM-Planner [137, 158]), combining LLMs with hierarchical planning models. A more detailed summary of the intersection of LLMs and navigation is provided in Lin et al. [93]. Metrics like success rate and path length are common in evaluating agents, as seen in environments like Matterport3D [3] and ALFRED [155]. Success rates in REVERIE [134] combine navigation and object identification, and VirtualHome uses Longest Common Subsequence (LCS) to compare actions with ground truth. ALFRED [155] uses path-weighted success rates for task completion efficiency, while FAVOR evaluates the quality of object rearrangement using perceptual metrics [87].

Challenges. Navigation and task execution environments are complex, with ALFRED's [155] leaderboard [8] showing a significant gap to human performance, as the best model achieves only a 67% success rate. Challenges remain, particularly in the simulation-to-reality gap and the difficulty in interpreting ambiguous instructions. Metrics like Task Success and Goal-Condition Success in ALFRED and navigation accuracy in EmbodiedQA provide valuable insights but often fail to generalize across new environments. Future research could explore improving generalization by

[8] X-WebArena-Leaderboard

integrating LLMs into vision-and-language tasks, which have been largely missing in navigation environments, with some exceptions like Feng et al. [47], Rajvanshi et al. [137].

5.2.3 Robotic Environments. We differentiate robotic environments from 2D/3D environments, as the former focuses on enabling real robots to understand and carry out tasks based on natural language instructions in dynamic settings [56, 103], whereas the latter refers to simulated environments. All suryeved papers in this category involve robots or robotic systems executable on robots [118]. Key features include integrating sensory inputs like visual, tactile[106], and proprioceptive data [118], allowing robots (e.g., Franka Emika Panda robot arm [106, 118, 157], PR2 [21, 109], Baxter [147], robotic lift [103, 168] etc...) to interact effectively with their surroundings. Robots mostly operate in **partially observable** environments, interpreting complex language commands for tasks like **navigation, manipulation, and trajectory reshaping** [21, 157, 168]. State spaces typically include the robot's position, object locations, and environmental features. For example, [109, 168, 169] define locations and paths using semantic maps, while Mees et al. [106] and others focus on object positions and robot joints [14, 21]. CALVIN [106] uses multiple cameras to capture sensory inputs, focusing on RGB data, while other papers incorporate depth images for multi-step tasks [157].

Methods and Evaluation. Many studies focus on models that combine probabilistic and neuro-symbolic approaches [157, 169], using tools like Generalized Grounding Graphs (G3) [169] to map language to actions. Other works utilize **imitation learning** and **offline reinforcement learning**, as in MCIL and LOReL, to train agents for object manipulation [106, 118]. Existing surveys [69, 74, 156, 167] on RL for robotics delve deeper into available approaches. More recently, LLMs have been explored in robotic tasks, like SayCan [1], which combines LLMs with RL for task execution, where LLMs handle high-level understanding, and RL governs actions. A common theme is the pairing of language commands with robot actions and environmental contexts, such as route instructions [169], spatial relations, or mobile-manipulation environments [168]. Furthermore, several datasets focus on synthetic data for robot manipulation, including single- and multi-step commands [157]. Others, like Scalise et al. [147], leverage crowdsourced data for real-world environments [147]. Evaluation typically focuses on task completion accuracy, generalization, and alignment between actions and language commands, with metrics like Task Success Rates, Sequence Completion, and Trajectory Prediction Accuracy.

Challenges. Robotic environments remain unsolved, with CALVIN [106] showing a success rate of only 42% [9] for long-horizon tasks as of October 2024. These studies aim to closely mimic real-world tasks, and some have potential applications in real environments, such as warehouse robots [106, 118]. However, issues like the need for better planning strategies, particularly for**long-horizon tasks**, and limited large-scale environments remain challenging [106, 109, 157]. Similar to simulation environments, LLMs are still underutilized in robotic tasks, with SayCan being a notable exception. More work is needed to improve **LLM integration** with robotic visual perception and decision-making [177].

5.2.4 Game Environments. Simulating real-world challenges in game environments, e.g., strategic planning in chess [170], collaborative building in Minecraft [61, 120], and procedural level generation in educational games [55], is common. We further divide game environments w.r.t. the task they focus on: **Structured, Strategic, and Predictive Environments** feature deterministic environments like chess and Othello [86, 170], where state, action, and observation spaces are defined by the positions and legal moves on the board. **Collaborative and Interactive Virtual Worlds** like Minecraft involve 3D environments where agents manipulate blocks and navigate using a limited view of the world [61, 120].

[9] http://calvin.cs.uni-freiburg.de

These environments focus on collaborative tasks. **Abstract and Procedural Knowledge Environments** such as HEXAGONS [80] and ScriptWorld [65] emphasize abstract reasoning and procedural knowledge. In ScriptWorld, tasks are defined by sequential textual scenarios, while HEXAGONS focuses on tile painting within a hexagonal grid. **Adaptive and Text-based Environments** like TextWorld [101] and SmartPlay [186] challenge AI systems with tasks requiring spatial reasoning and decision-making. TextWorld involves text-based navigation, while SmartPlay evaluates agents across diverse mini-games, including Rock-Paper-Scissors and Minecraft-like environments.

Methods and Evaluation. State spaces differ across environments, such as chessboard configurations [170], block positions in Minecraft [120], or hexagonal grids in HEXAGONS [80]. Observation spaces range from tile states [170] to visual data in Minecraft [186], while action spaces include moving game pieces or manipulating objects [120]. Approaches vary: Chess [170] and Othello [86] use GPT-2 and Othello-GPT for move prediction, while Minecraft uses Seq2Seq and BAP models for action prediction [120]. HEXAGONS [80] uses DeBERTa and T5 for abstract tasks, while reinforcement learning agents are used in ScriptWorld. TextWorld [101] tests RL agents like LSTM-DQN, while SmartPlay [186] evaluates large language models across games. A comprehensive overview of LLM performances on game environments can be found in [49]. For evaluation, in addition to traditional metrics such as task success, other metrics such as communication efficiency [120] are commonly used.

Challenges. Game environments present several challenges, particularly in handling dynamic and interactive tasks. One significant challenge is the adaptability across different types of environments—ranging from highly structured ones like chess to more open-ended worlds like Minecraft—poses difficulties, as each environment demands specialized reasoning and perception capabilities. The diversity of observation, state, and action spaces further complicates the development of generalized solutions. Another major challenge is integrating multimodal information, as environments like Minecraft or SmartPlay involve both visual and textual data, necessitating robust cross-modal understanding. For future work, developing more advanced models capable of abstract reasoning, improved collaboration in multi-agent environments, and enhanced multimodal fusion techniques is essential.

5.2.5 GUI Environments. Mobile environments for grounded task execution, also known as GUI environments, enable AI agents to interact with mobile app interfaces through actions like tapping, swiping, and navigating to perform diverse set of tasks. Similar to web environments, mobile environments focus on GUI interactions but extend to a broader range of applications, from system-level tasks to diverse fields like education [22], business [7, 198], and entertainment [90]. Key datasets and platforms for mobile environments share a focus on vision-language navigation and UI interaction. MoTIF [22] provides 6,100 natural language tasks with action localization, while AndroidEnv [171] offers a reinforcement learning platform for Android, with task-specific annotations. UICaption [7] pairs UI images with captions, though it lacks event annotations. PIXELHELP, ANDROIDHOWTO, and RICOSCA [90] map natural language to UI actions. Mobile-Env's WikiHow Task Set [198] offers a benchmark for LLM-based agents with detailed annotations for cross-page navigation and QA tasks.

Action, state, and observation spaces in mobile environments reflect real app interactions. MoTIF [22] and Mobile-Env [198] define the state space by app view hierarchy, and actions like tapping and swiping. On the other hand, AndroidEnv [171] focuses on pixel-based reinforcement learning interactions, while Lexi [7] expands UI understanding to both mobile and desktop. Observation spaces also vary, including UI object hierarchies [90], pixels, and screenshots [7], while actions involve diverse UI manipulations like dragging and typing. Finally, platforms like AndroidEnv [171] and Mobile-Env [198] simulate real-world operations on actual devices, including Pixel phones for realistic evaluations [90], while MoTIF [22] focuses on task feasibility but does not extend to real-world testing on physical devices, concentrating on controlled simulations.

Methods and Evaluation. Models like GPT-3.5-turbo, GPT-4, LLaMA 2 [198], and vision models like ViLBERT [7] are commonly used. Metrics include accuracy and recall, while reinforcement learning environments add completion rates and rewards [171, 198].

Challenges. Handling diverse app interfaces, generalizing across apps, and adapting to unseen environments remain challenging. Though AI models show promise, they lag behind humans in complex gesture handling and real-time execution. MoTIF [22] faces difficulties with app state variability, and AndroidEnv [171] struggles with complex gestures. Future work should focus on improving cross-app generalization and enhancing real-world performance, with potential for LLMs to play a greater role in these environments, as discussed in [91].

5.2.6 Relation to Program Synthesis. Apart from the environments mentioned in this section, one relevant literature to *grounded tasks* is **code generation**, i.e., program synthesis. Here, similar to Web or mobile environments, the goal is to generate code that can be executed by a symbolic interpreter, e.g., Python, SQL, etc., on a certain environment. However, the focus is learning the **conversion process in an offline manner through large annotated sets**, rather than through environment interaction. One such benchmark is InstructExcel [132] that focus on automating Excel operations by converting user-provided natural language descriptions into TypeScript code that can be executed on spreadsheets. The tasks range from conditional formatting and formula creation to data manipulation and charting, making the benchmark highly relevant to real-world Excel use cases. It offers an offline test bed to evaluate modern LLMs such as GPT-4, and GPT-3.5 Turbo. We refer the readers to other related surveys [63, 193] on program synthesis for more information.

5.2.7 Grounded Task-Oriented Dialogue. Task-oriented dialogue (TOD) systems facilitate specific tasks through conversational interactions. These systems use underlying resources such as documents, databases, or live environmental data to ensure accurate and contextually relevant responses. Researchers have extended TOD systems across various domains, tailoring them to function effectively within interactive and embodied environments, such as guiding users in navigation, construction, or household chores within simulated or virtual settings [50, 113, 120, 127, 159]. In customer support, TOD systems enhance technical assistance or customer service by utilizing documents or flowcharts [25, 45, 46, 136, 162]. Additionally, specialized TOD systems cater to specific needs, such as managing cooking recipes [64], supporting data visualization [151], simulating interactions in mobile GUIs [164], and facilitating storytelling within virtual worlds [2].

Data collection for these dialogue systems is generally expensive, as it typically requires two users to engage in a conversation. Researchers have adopted several approaches to address this challenge. For instance, Wizard-of-Oz methods have been employed in datasets such as META-GUI [164], ChartDialogs [151], and CookDial [64]. Human-human interaction is used in the creation of datasets like Minicraft [120], ABCD [25], Task2Dial [162], and TEACh [127]. Additionally, human annotation is used in datasets such as CraftAssist Instruction Parsing [159], DialFRED [50], Doc2Dial [46], MultiDoc2Dial [45], and FLONET [136].

Systems can be grounded on **documents** and other static data sources, such as flowcharts and knowledge bases [2, 25, 45, 46, 64, 113, 136, 151, 162], where the model has user utterances and grounding documents as input, to generate a sequence of actions or commands. Alternatively, systems can be grounded on the **environment** [50, 120, 127, 159, 164], where the input consists of user utterances plus real-time environmental state, and the output is dynamically responsive sequences of actions performed within that environment. This allows for interactions based on navigating physical spaces or manipulating objects within simulations.

Methods and Evaluation. Transformer-based models such as BERT dominate the field, where they are used for retrieval tasks [45, 46], action state tracking [25, 64], and to integrate language and visual inputs to handle complex task-oriented dialogues in both mobile GUIs and embodied environments [127, 164]. whereas Seq2Seq models are extensively used for generating instructions and parsing natural language [2, 50, 120, 151, 159]. Additionally, RAG models are used [45, 136] to effectively ground responses in specific document contexts while allowing for dynamic response generation. Building on these model architectures, the evaluation of these systems varies widely. Exact Match (EM) and F1 Score are frequently employed to measure the accuracy of span selection and response grounding [45, 46]. BLEU Score is another common metric used to evaluate the quality of generated responses and plot updates [25, 45, 46, 151]. Success Rate (SR) and Goal-Conditioned Success (GCS) are crucial metrics for evaluating task completion and goal achievement [25, 50, 127, 136]. Additionally, Recall metrics assess the accuracy of retrieval tasks [25, 45]. App. E showcases a variety of datasets used in grounded TOD research.

Challenges. While these works represent substantial advancements in task-oriented dialogue systems, they also underscore ongoing challenges in achieving robust, generalizable solutions [25, 45, 46, 50]. Recent successes, such as employing fine-tuned models like T5 and leveraging graph-based predictive techniques [138] and leveraging LLMs [58] improved the Success Rate by 20% and 33.4% on ABCD ad FLODial datasets, respectively. However, continued innovation is necessary to address issues related to contextual understanding, multi-modal integration, and real-world applicability, as these tasks remain far from fully resolved [2, 127, 164]. Moreover, many systems still lack system-level evaluation metrics, such as success and inform rate, which are crucial as they directly reflect the user's perceived outcomes [41, 45, 164]. Additionally, support for multilinguality is limited, constraining applicability in global contexts where users may interact in various languages.

6 CONCLUSION

In this work, we provide a comprehensive, systematic and interdisciplinary survey on existing representation formats and downstream tasks for complex, long-form instructional text. We identify the bibliographic properties and trends for the topic. Furthermore, we highlight the challenges and gaps in the literature for representation formats and tasks separately in §4 and §5.

ACKNOWLEDGMENTS

This work has been supported by the Scientific and Technological Research Council of Türkiye (TÜBİTAK) as part of the project "Automatic Learning of Procedural Language from Natural Language Instructions for Intelligent Assistance" with the number 121C132. We also gratefully acknowledge KUIS AI Lab for providing computational support. Special thanks to Müge Kural for her valuable assistance during the paper identification stage.

REFERENCES

[1] Michael Ahn, Anthony Brohan, Noah Brown, et al. 2022. Do As I Can, Not As I Say: Grounding Language in Robotic Affordances. In *Conference on Robot Learning*. https://api.semanticscholar.org/CorpusID:247939706

[2] Prithviraj Ammanabrolu and Mark O. Riedl. 2021. Modeling Worlds in Text. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual.* https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/1f0e3dad99908345f7439f8ffabdffc4-Abstract-round1.html

[3] Peter Anderson, Qi Wu, Damien Teney, et al. 2017. Vision-and-Language Navigation: Interpreting Visually-Grounded Navigation Instructions in Real Environments. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2017), 3674–3683. https://api.semanticscholar.org/CorpusID:4673790

[4] Talita Anthonio, Irshad Bhat, and Michael Roth. 2020. wikiHowToImprove: A Resource and Analyses on Edits in Instructional Texts. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*. European Language Resources Association, 5721–5729. https://aclanthology.org/2020.lrec-1.702/

[5] Yoav Artzi and Luke Zettlemoyer. 2013. Weakly Supervised Learning of Semantic Parsers for Mapping Instructions to Actions. *Transactions of the Association for Computational Linguistics* 1 (2013), 49–62. https://doi.org/10.1162/tacl_a_00209

[6] Mohannad Babli and Eva Onaindía. 2019. A context-aware knowledge acquisition for planning applications using ontologies. *arXiv* 1904.09845 (2019). arXiv:1904.09845 https://arxiv.org/abs/1904.09845

[7] Pratyay Banerjee, Shweti Mahajan, Kushal Arora, et al. 2023. Lexi: Self-Supervised Learning of the UI Language. *arXiv* 2301.10165 (2023). arXiv:2301.10165 https://arxiv.org/abs/2301.10165

[8] Devansh Batra, Nirav Diwan, Utkarsh Upadhyay, et al. 2020. RecipeDB: a resource for exploring recipes. *Database* 2020 (2020), baaa077. https://doi.org/10.1093/database/baaa077 arXiv:https://academic.oup.com/database/article-pdf/doi/10.1093/database/baaa077/34526636/baaa077.pdf

[9] Lisa Bauer and Mohit Bansal. 2021. Identify, Align, and Integrate: Matching Knowledge Graphs to Commonsense Reasoning Tasks. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Association for Computational Linguistics, 2259–2272. https://doi.org/10.18653/v1/2021.eacl-main.192

[10] Patrizio Bellan, Han van der Aa, Mauro Dragoni, et al. 2023. PET: An Annotated Dataset for Process Extraction from Natural Language Text Tasks. In *Business Process Management Workshops*. Springer International Publishing, 315–321.

[11] Dhaivat J. Bhatt, Seyed Ahmad Abdollahpouri Hosseini, Federico Fancellu, et al. 2024. End-to-end Parsing of Procedural Text into Flow Graphs. In *LREC:2024:main*. ELRA and ICCL, 5833–5842.

[12] Michał Bień, Michał Gilski, Martyna Maciejewska, et al. 2020. RecipeNLG: A Cooking Recipes Dataset for Semi-Structured Text Generation. In *Proceedings of the 13th International Conference on Natural Language Generation" month = dec.* 22–28. https://doi.org/10.18653/v1/2020.inlg-1.4

[13] Yonatan Bisk, Jan Buys, Karl Pichotta, et al. 2019. Benchmarking Hierarchical Script Knowledge. 4077–4085. https://doi.org/10.18653/v1/N19-1412

[14] Yonatan Bisk, Deniz Yuret, and Daniel Marcu. 2016. Natural Language Communication with Robots. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 751–761. https://doi.org/10.18653/v1/N16-1089

[15] Valeria Bolotova-Baranova, Vladislav Blinov, Sofya Filippova, et al. 2023. WikiHowQA: A Comprehensive Benchmark for Multi-Document Non-Factoid Question Answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*. Association for Computational Linguistics, 5291–5314. https://doi.org/10.18653/V1/2023.ACL-LONG.290

[16] Marco Bombieri, Daniele Meli, Diego Dall'Alba, et al. 2023. Mapping natural language procedures descriptions to linear temporal logic templates: an application in the surgical robotic domain. *Applied Intelligence* 53, 22 (2023), 26351–26363.

[17] Odellia Boni, Guy Feigenblat, Guy Lev, et al. 2021. HowSumm: A Multi-Document Summarization Dataset Derived from WikiHow Articles. *arXiv* 2110.03179 (2021). arXiv:2110.03179 https://arxiv.org/abs/2110.03179

[18] Antoine Bosselut, Omer Levy, Ari Holtzman, et al. 2018. Simulating Action Dynamics with Neural Process Networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=rJYFzMZC-

[19] S.R.K. Branavan, Harr Chen, Luke Zettlemoyer, et al. 2009. Reinforcement Learning for Mapping Instructions to Actions. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. Association for Computational Linguistics, 82–90. https://aclanthology.org/P09-1010

[20] Anna Breit, Laura Waltersdorfer, Fajar J. Ekaputra, et al. 2023. Combining Machine Learning and Semantic Web: A Systematic Mapping Study. *Comput. Surveys* 55 (2023), 1 – 41. https://api.semanticscholar.org/CorpusID:257406478

[21] Arthur Fender C. Bucker, Luis F. C. Figueredo, Sami Haddadin, et al. 2022. Reshaping Robot Trajectories Using Natural Language Commands: A Study of Multi-Modal Data Alignment Using Transformers. *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2022), 978–984. https://api.semanticscholar.org/CorpusID:247749081

[22] Andrea Burns, Deniz Arsan, Sanjna Agrawal, et al. 2022. A Dataset for Interactive Vision-Language Navigation with Unknown Command Feasibility. In *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part VIII (Lecture Notes in Computer Science, Vol. 13668)*. Springer, 312–328. https://doi.org/10.1007/978-3-031-20074-8_18

[23] Khyathi Raghavi Ch, u, Yonatan Bisk, et al. 2021. Grounding 'Grounding' in NLP. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021 (Findings of ACL, Vol. ACL/IJCNLP 2021)*. Association for Computational Linguistics, 4283–4305. https://doi.org/10.18653/v1/2021.findings-acl.375

[24] Derek Chen, Howard Chen, Yi Yang, et al. 2021. Action-Based Conversations Dataset: A Corpus for Building More In-Depth Task-Oriented Dialogue Systems. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*. Association for Computational Linguistics, 3002–3017. https://doi.org/10.18653/v1/2021.naacl-main.239

[25] Derek Chen, Howard Chen, Yi Yang, et al. 2021. Action-Based Conversations Dataset: A Corpus for Building More In-Depth Task-Oriented Dialogue Systems. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 3002–3017. https://doi.org/10.18653/v1/2021.naacl-main.239

[26] Guangyong Chen, Pengfei Chen, Chang-Yu Hsieh, et al. 2019. Alchemy: A Quantum Chemistry Dataset for Benchmarking AI Models. *arXiv* 1906.09427 (2019). arXiv:1906.09427 https://arxiv.org/abs/1906.09427

[27] Muhao Chen, Hongming Zhang, Qiang Ning, et al. 2021. Event-Centric Natural Language Processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Tutorial Abstracts*. Association for Computational Linguistics, 6–14. https://doi.org/10.18653/v1/2021.acl-tutorials.2

[28] Muhao Chen, Hongming Zhang, Haoyu Wang, et al. 2020. What Are You Trying to Do? Semantic Typing of Event Processes. In *CONLL:2020:1*. 531–542. https://doi.org/10.18653/v1/2020.conll-1.43

[29] Pengxiang Cheng and Katrin Erk. 2018. Implicit Argument Prediction with Event Knowledge. 831–840. https://doi.org/10.18653/v1/N18-1076

[30] Cuong Xuan Chu, Niket T, on, et al. 2017. Distilling Task Knowledge from How-To Communities. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*. ACM, 805–814. https://doi.org/10.1145/3038912.3052715

[31] Peter Clark, Bhavana Dalvi, Niket T, et al. 2018. What Happened? Leveraging VerbNet to Predict the Effects of Actions in Procedural Text. *arXiv* 1804.05435 (2018). arXiv:1804.05435 https://arxiv.org/abs/1804.05435

[32] Vanya Cohen, Jason Xinyu Liu, Raymond Mooney, et al. 2024. A Survey of Robotic Language Grounding: Tradeoffs Between Symbols and Embeddings. *arXiv* 2405.13245 (2024). https://doi.org/10.48550/ARXIV.2405.13245 arXiv:2405.13245

[33] Carlos M. Costa, Germano Veiga, Arm Sousa, et al. 2017. Evaluation of Stanford NER for extraction of assembly information from instruction manuals. In *2017 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*. 302–309. https://doi.org/10.1109/ICARSC.2017.7964092

[34] Nico Daheim, David Thulke, Christian Dugast, et al. 2021. Cascaded Span Extraction and Response Generation for Document-Grounded Dialog. In *DIALDOC:2021:1*. 57–62. https://doi.org/10.18653/v1/2021.dialdoc-1.8

[35] Nico Daheim, David Thulke, Christian Dugast, et al. 2022. Controllable Factuality in Document-Grounded Dialog Systems Using a Noisy Channel Model. In *FINDINGS:2022:emnlp*. 1365–1381. https://doi.org/10.18653/v1/2022.findings-emnlp.98

[36] Bhavana Dalvi, Lifu Huang, Niket T, et al. 2018. Tracking State Changes in Procedural Text: a Challenge Dataset and Models for Process Paragraph Comprehension. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*. Association for Computational Linguistics, 1595–1604. https://doi.org/10.18653/v1/n18-1144

[37] Bhavana Dalvi, Niket T, on, et al. 2019. Everything Happens for a Reason: Discovering the Purpose of Actions in Procedural Text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*. Association for Computational Linguistics, 4495–4504. https://doi.org/10.18653/v1/D19-1457

[38] Abhishek Das, Samyak Datta, Georgia Gkioxari, et al. 2017. Embodied Question Answering. *arXiv* 1711.11543 (2017). arXiv:1711.11543 https://arxiv.org/abs/1711.11543

[39] Chad DeChant and Daniel Bauer. 2022. Summarizing a virtual robot's past actions in natural language. *arXiv* 2203.06671 (2022). arXiv:2203.06671 https://arxiv.org/abs/2203.06671

[40] Xiang Deng, Yu Gu, Boyuan Zheng, et al. 2023. Mind2Web: Towards a Generalist Agent for the Web. *arXiv* 2306.06070 (2023). https://doi.org/10.48550/ARXIV.2306.06070 arXiv:2306.06070

[41] Yang Deng, Xuan Zhang, Wenxuan Zhang, et al. 2024. On the Multi-turn Instruction Following for Conversational Web Agents. In *ACL:2024:long*. 8795–8812. https://doi.org/10.18653/v1/2024.acl-long.477

[42] Nirav Diwan, Devansh Batra, and Ganesh Bagler. 2020. A named entity based approach to model recipes. In *2020 IEEE 36th International Conference on Data Engineering Workshops (ICDEW)*. 88–93.

[43] Lucia Donatelli, Theresa Schmidt, Debanjali Biswas, et al. 2021. Aligning Actions Across Recipe Graphs. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*. Association for Computational Linguistics, 6930–6942. https://doi.org/10.18653/V1/2021.EMNLP-MAIN.554

[44] Yi Fan and Anthony Hunter. 2023. Understanding the Cooking Process with English Recipe Text. In *FINDINGS:2023:acl*. 4244–4264. https://doi.org/10.18653/v1/2023.findings-acl.261

[45] Song Feng, Siva Sankalp Patel, Hui Wan, et al. 2021. MultiDoc2Dial: Modeling Dialogues Grounded in Multiple Documents. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

[46] Song Feng, Hui Wan, Chulaka Gunasekara, et al. 2020. doc2dial: A Goal-Oriented Document-Grounded Dialogue Dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 8118–8128. https://doi.org/10.18653/v1/2020.emnlp-main.652

[47] Weixi Feng, Tsu-Jui Fu, Yujie Lu, et al. 2022. ULN: Towards Underspecified Vision-and-Language Navigation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*. Association for Computational Linguistics, 6394–6412. https://doi.org/10.18653/V1/2022.EMNLP-MAIN.429

[48] Wenfeng Feng, Hankz Hankui Zhuo, and Subbarao Kambhampati. 2018. Extracting Action Sequences from Texts Based on Deep Reinforcement Learning. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden.* ijcai.org, 4064–4070. https://doi.org/10.24963/ijcai.2018/565
[49] Roberto Gallotta, Graham Todd, Marvin Zammit, et al. 2024. Large Language Models and Games: A Survey and Roadmap. *arXiv* 2402.18659 (2024). arXiv:2402.18659 https://arxiv.org/abs/2402.18659
[50] Xiaofeng Gao, Qiaozi Gao, Ran Gong, et al. 2022. DialFRED: Dialogue-Enabled Agents for Embodied Instruction Following. *IEEE Robotics and Automation Letters* 7 (2022), 10049–10056. https://api.semanticscholar.org/CorpusID:247158852
[51] Saransh Goyal, Pratyush P, ey, et al. 2021. Tracking entities in technical procedures - a new dataset and baselines. *arXiv* 2104.07378 (2021). arXiv:2104.07378 https://arxiv.org/abs/2104.07378
[52] Izzeddin Gur, Hiroki Furuta, Austin Huang, et al. 2024. A Real-World WebAgent with Planning, Long Context Understanding, and Program Synthesis. arXiv:2307.12856
[53] Jun Harashima and Makoto Hiramatsu. 2020. Cookpad Parsed Corpus: Linguistic Annotations of Japanese Recipes. In *Proceedings of the 14th Linguistic Annotation Workshop*. Association for Computational Linguistics, 87–92. https://aclanthology.org/2020.law-1.8
[54] Qianyu He, Jie Zeng, Wenhao Huang, et al. 2024. Can Large Language Models Understand Real-World Complex Instructions? arXiv:2309.09150 https://arxiv.org/abs/2309.09150
[55] Danial Hooshyar, Moslem Yousefi, Minhong Wang, et al. 2018. A data-driven procedural-content-generation approach for educational games. *J. Comput. Assist. Learn.* 34 (2018), 731–739. https://api.semanticscholar.org/CorpusID:54449301
[56] Thomas M. Howard, Stefanie Tellex, and Nicholas Roy. 2014. A natural language planner interface for mobile manipulators. *2014 IEEE International Conference on Robotics and Automation (ICRA)* (2014), 6652–6659. https://api.semanticscholar.org/CorpusID:15011253
[57] Hengyuan Hu, Denis Yarats, Qucheng Gong, et al. 2019. Hierarchical Decision Making by Generating and Following Natural Language Instructions. *arXiv* 1906.00744 (2019). arXiv:1906.00744 https://arxiv.org/abs/1906.00744
[58] Zhiyuan Hu, Chumin Liu, Xidong Feng, et al. 2024. Uncertainty of Thoughts: Uncertainty-Aware Planning Enhances Information Seeking in Large Language Models. arXiv:2402.03271
[59] Nobuhiro Ito, Yuya Suzuki, and Akiko Aizawa. 2020. From Natural Language Instructions to Complex Processes: Issues in Chaining Trigger Action Rules. arXiv:2001.02462
[60] Ayush Jain, Pushkal Katara, Nikolaos Gkanatsios, et al. 2024. ODIN: A Single Model for 2D and 3D Perception. arXiv:2401.02416
[61] Prashant Jayannavar, Anjali Narayan-Chen, and Julia Hockenmaier. 2020. Learning to execute instructions in a Minecraft dialogue. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2589–2602. https://doi.org/10.18653/v1/2020.acl-main.232
[62] Jermsak Jermsurawong and Nizar Habash. 2015. Predicting the Structure of Cooking Recipes. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 781–786. https://doi.org/10.18653/v1/D15-1090
[63] Juyong Jiang, Fan Wang, Jiasi Shen, et al. 2024. A Survey on Large Language Models for Code Generation. *arXiv* 2406.00515 (2024). https://doi.org/10.48550/ARXIV.2406.00515 arXiv:2406.00515
[64] Yiwei Jiang, Klim Zaporojets, Johannes Deleu, et al. 2022. CookDial: a dataset for task-oriented dialogs grounded in procedural documents. *Applied Intelligence* 53, 4 (2022), 4748–4766. https://doi.org/10.1007/s10489-022-03692-0
[65] Abhinav Joshi, Areeb Ahmad, P, et al. 2023. ScriptWorld: text based environment for learning procedural knowledge. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI '23)*. https://doi.org/10.24963/ijcai.2023/566
[66] Yuchul Jung, Jihee Ryu, Kyung-Min Kim, et al. 2010. Automatic construction of a large-scale situation ontology by mining how-to instructions from the web. *J. Web Semant.* 8, 2-3 (2010), 110–124. https://doi.org/10.1016/J.WEBSEM.2010.04.006
[67] Yuchul Jung, Jihee Ryu, Kyung-min Kim, et al. 2010. Automatic construction of a large-scale situation ontology by mining how-to instructions from the web. *Web Semant.* 8, 2–3 (2010), 110–124. https://doi.org/10.1016/j.websem.2010.04.006
[68] Ghazaleh Kazeminejad and Martha Palmer. 2023. Event Semantic Knowledge in Procedural Text Understanding. 388–398. https://doi.org/10.18653/v1/2023.starsem-1.33
[69] Md. Al-Masrur Khan, Md Rashed Jaowad Khan, Abul Tooshil, et al. 2020. A Systematic Review on Reinforcement Learning-Based Robotics Within the Last Decade. *IEEE Access* 8 (2020), 176598–176623. https://doi.org/10.1109/ACCESS.2020.3027152
[70] Chloé Kiddon, Ganesa Th Ponnuraj, avam, et al. 2015. Mise en Place: Unsupervised Interpretation of Instructional Recipes. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 982–992. https://doi.org/10.18653/v1/D15-1114
[71] Chloé Kiddon, Luke Zettlemoyer, and Yejin Choi. 2016. Globally Coherent Text Generation with Neural Checklist Models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016.* The Association for Computational Linguistics, 329–339. https://doi.org/10.18653/v1/d16-1032
[72] Geunwoo Kim, Pierre Baldi, and Stephen Marcus McAleer. 2023. Language Models can Solve Computer Tasks. *arXiv* 2303.17491 (2023). arXiv:2303.17491 https://arxiv.org/abs/2303.17491
[73] Najoung Kim and Sebastian Schuster. 2023. Entity Tracking in Language Models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023.* Association for Computational Linguistics, 3835–3855. https://aclanthology.org/2023.acl-long.213

[74] Jens Kober and Jan Peters. 2014. *Reinforcement Learning in Robotics: A Survey.* Springer International Publishing, 9–67. https://doi.org/10.1007/978-3-319-03194-1_2

[75] Mateja Kocbek, Gregor Jost, Marjan Hericko, et al. 2015. Business process model and notation: The current state of affairs. *Comput. Sci. Inf. Syst.* 12, 2 (2015), 509–539. https://doi.org/10.2298/CSIS140610006K

[76] Mahnaz Koupaee and William Yang Wang. 2018. WikiHow: A Large Scale Text Summarization Dataset. *arXiv* 1810.09305 (2018). arXiv:1810.09305 https://arxiv.org/abs/1810.09305

[77] Humam Kourani, Aless, ro Berti, et al. 2024. Process Modeling With Large Language Models. *arXiv* 2403.07541 (2024). https://doi.org/10.48550/ARXIV.2403.07541 arXiv:2403.07541

[78] Ranjay Krishna, Kenji Hata, Frederic Ren, et al. 2017. Dense-Captioning Events in Videos. In *2017 IEEE International Conference on Computer Vision (ICCV).* 706–715. https://doi.org/10.1109/ICCV.2017.83

[79] Vikram Kumaran, Dan Carpenter, Jonathan Rowe, et al. 2024. Procedural Level Generation in Educational Games from Natural Language Instruction. *IEEE Transactions on Games* (2024).

[80] Royi Lachmy, Valentina Pyatkin, Avshalom Manevich, et al. 2022. Draw Me a Flower: Processing and Grounding Abstraction in Natural Language. *Trans. Assoc. Comput. Linguistics* 10 (2022), 1341–1356. https://transacl.org/ojs/index.php/tacl/article/view/3961

[81] Faisal Ladhak, Esin Durmus, Claire Cardie, et al. 2020. WikiLingua: A New Benchmark Dataset for Cross-Lingual Abstractive Summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2020.* Association for Computational Linguistics, 4034–4048. https://doi.org/10.18653/v1/2020.findings-emnlp.360

[82] Yash Kumar Lal, Vanya Cohen, Nathanael Chambers, et al. 2024. CaT-BENCH: Benchmarking Language Model Understanding of Causal and Temporal Dependencies in Plans. *arXiv preprint arXiv:2406.15823* (2024).

[83] Thang Le and Anh Tuan Luu. 2024. Extractive Summarization with Text Generator. 157–174. https://doi.org/10.18653/v1/2024.naacl-long.9

[84] Thang Le and Luu Anh Tuan. 2024. Extractive Summarization with Text Generator. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.* acl.

[85] Belinda Z. Li, Maxwell Nye, and Jacob Andreas. 2021. Implicit Representations of Meaning in Neural Language Models. 1813–1827. https://doi.org/10.18653/v1/2021.acl-long.143

[86] Kenneth Li, Aspen K. Hopkins, David Bau, et al. 2022. Emergent World Representations: Exploring a Sequence Model Trained on a Synthetic Task. *arXiv* 2210.13382 (2022). arXiv:2210.13382 https://arxiv.org/abs/2210.13382

[87] Kailin Li, Lixin Yang, Zenan Lin, et al. 2024. FAVOR: Full-Body AR-Driven Virtual Object Rearrangement Guided by Instruction Text. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada.* AAAI Press, 3136–3144. https://doi.org/10.1609/AAAI.V38I4.28097

[88] Qian Li, Jianxin Li, Jiawei Sheng, et al. 2022. A survey on deep learning event extraction: Approaches and applications. *IEEE Transactions on Neural Networks and Learning Systems* (2022).

[89] Xinghang Li, Di Guo, Huaping Liu, et al. 2022. REVE-CE: Remote Embodied Visual Referring Expression in Continuous Environment. *IEEE Robotics and Automation Letters* 7, 2 (2022), 1494–1501. https://doi.org/10.1109/LRA.2022.3141150

[90] Yang Li, Jiacong He, Xin Zhou, et al. 2020. Mapping Natural Language Instructions to Mobile UI Action Sequences. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020.* Association for Computational Linguistics, 8198–8210. https://doi.org/10.18653/v1/2020.acl-main.729

[91] Yuanchun Li, Hao Wen, Weijun Wang, et al. 2024. Personal LLM Agents: Insights and Survey about the Capability, Efficiency and Security. *arXiv* 2401.05459 (2024). arXiv:2401.05459 https://arxiv.org/abs/2401.05459

[92] Angela Lin, Sudha Rao, Asli Celikyilmaz, et al. 2020. A Recipe for Creating Multimodal Aligned Datasets for Sequential Tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics.* Association for Computational Linguistics, 4871–4884. https://doi.org/10.18653/v1/2020.acl-main.440

[93] Jinzhou Lin, Han Gao, Rongtao Xu, et al. 2023. Advances in Embodied Navigation Using Large Language Models: A Survey. https://api.semanticscholar.org/CorpusID:264832983

[94] Evan Zheran Liu, Kelvin Guu, Panupong Pasupat, et al. 2018. Reinforcement Learning on Web Interfaces using Workflow-Guided Exploration. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings.* OpenReview.net. https://openreview.net/forum?id=ryTp3f-0-

[95] Xiao Liu, Yansong Feng, Jizhi Tang, et al. 2022. Counterfactual Recipe Generation: Exploring Compositional Generalization in a Realistic Scenario. 7354–7370. https://doi.org/10.18653/v1/2022.emnlp-main.497

[96] Reginald Long, Panupong Pasupat, and Percy Liang. 2016. Simpler Context-Dependent Logical Forms via Model Projections. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers.* The Association for Computer Linguistics. https://doi.org/10.18653/v1/p16-1138

[97] Yujie Lu, Pan Lu, Zhiyu Chen, et al. 2023. Multimodal Procedural Planning via Dual Text-Image Prompting. arXiv:2305.01795

[98] Qing Lyu, Li Zhang, and Chris Callison-Burch. 2021. Goal-Oriented Script Construction. In *Proceedings of the 14th International Conference on Natural Language Generation.* Association for Computational Linguistics, 184–200. https://aclanthology.org/2021.inlg-1.19

[99] Matt MacMahon, Brian Stankiewicz, and Benjamin Kuipers. 2006. Walk the Talk: Connecting Language, Knowledge, and Action in Route Instructions. In *Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, July 16-20, 2006, Boston, Massachusetts, USA*. AAAI Press, 1475–1482. http://www.aaai.org/Library/AAAI/2006/aaai06-232.php

[100] Hirokuni Maeta, Tetsuro Sasada, and Shinsuke Mori. 2015. A Framework for Procedural Text Understanding. 50–60. https://doi.org/10.18653/v1/W15-2206

[101] Marc-Alex, re Côté, Ákos Kádár, et al. 2018. TextWorld: A Learning Environment for Text-Based Games. In *Computer Games - 7th Workshop, CGW 2018, Held in Conjunction with the 27th International Conference on Artificial Intelligence, IJCAI 2018, Stockholm, Sweden, July 13, 2018, Revised Selected Papers (Communications in Computer and Information Science, Vol. 1017)*. Springer, 41–75. https://doi.org/10.1007/978-3-030-24337-1_3

[102] Javier Marin, Aritro Biswas, Ferda Ofli, Nicholas Hynes, Amaia Salvador, Yusuf Aytar, Ingmar Weber, and Antonio Torralba. 2019. Recipe1M+: A Dataset for Learning Cross-Modal Embeddings for Cooking Recipes and Food Images. *IEEE Trans. Pattern Anal. Mach. Intell.* (2019).

[103] Cynthia Matuszek, Evan V. Herbst, Luke Zettlemoyer, et al. 2012. Learning to Parse Natural Language Commands to a Robot Control System. In *International Symposium on Experimental Robotics*. https://api.semanticscholar.org/CorpusID:1658890

[104] Effrosyni Mavroudi, Triantafyllos Afouras, and Lorenzo Torresani. 2023. Learning to Ground Instructional Articles in Videos through Narrations. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)* (2023), 15155–15167. https://api.semanticscholar.org/CorpusID:259088701

[105] Sahisnu Mazumder and Oriana Riva. 2021. FLIN: A Flexible Natural Language Interface for Web Navigation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*. Association for Computational Linguistics, 2777–2788. https://doi.org/10.18653/V1/2021.NAACL-MAIN.222

[106] Oier Mees, Lukás Hermann, Erick Rosete-Beas, et al. 2021. CALVIN: A Benchmark for Language-Conditioned Policy Learning for Long-Horizon Robot Manipulation Tasks. *IEEE Robotics and Automation Letters* 7 (2021), 7327–7334. https://api.semanticscholar.org/CorpusID:244908821

[107] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, et al. 2019. HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips. In *ICCV*.

[108] Shivam Miglani and Neil Yorke-Smith. 2020. NLtoPDDL: One-Shot Learning of PDDL Models from Natural Language Process Manuals. In *ICAPS'20 Workshop on Knowledge Engineering for Planning and Scheduling (KEPS'20)*.

[109] Dipendra Misra, Jaeyong Sung, Kevin Lee, et al. 2014. Tell me Dave: Context-sensitive grounding of natural language to manipulation instructions. *The International Journal of Robotics Research* 35 (2014), 281 – 300. https://api.semanticscholar.org/CorpusID:8297420

[110] Dipendra Kumar Misra, Andrew Bennett, Valts Blukis, et al. 2018. Mapping Instructions to Actions in 3D Environments with Visual Goal Prediction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*. Association for Computational Linguistics, 2667–2678. https://doi.org/10.18653/v1/d18-1287

[111] Dipendra Kumar Misra, Kejia Tao, Percy Liang, et al. 2015. Environment-Driven Lexicon Induction for High-Level Instructions. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, 992–1002. https://doi.org/10.3115/v1/P15-1096

[112] Ashutosh Modi, Tatjana Anikina, Simon Ostermann, et al. 2016. InScript: Narrative texts annotated with script information. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. European Language Resources Association (ELRA).

[113] Nikita Moghe, Patrick Xia, Jacob Andreas, et al. 2024. Interpreting User Requests in the Context of Natural Language Standing Instructions. 4043–4060. https://doi.org/10.18653/v1/2024.findings-naacl.255

[114] Sheshera Mysore, Zach Jensen, Edward Kim, et al. [n. d.]. The Materials Science Procedural Text Corpus: Annotating Materials Synthesis Procedures with Shallow Semantic Structures. In *Proceedings of the 13th Linguistic Annotation Workshop, LAW*.

[115] N, Abhilash y, Manav Kapadnis, et al. 2023. CLMSM: A Multi-Task Learning Framework for Pre-training on Procedural Text. 8793–8806. https://doi.org/10.18653/v1/2023.findings-emnlp.589

[116] N, Abhilash y, Yash Kulkarni, et al. 2024. Order-Based Pre-training Strategies for Procedural Text Understanding. 821–828. https://doi.org/10.18653/v1/2024.naacl-short.74

[117] Nima Nabizadeh, Dorothea Kolossa, and Martin Heckmann. 2020. MyFixit: An Annotated Dataset, Annotation Tool, and Baseline Methods for Information Extraction from Repair Manuals. European Language Resources Association, 2120–2128.

[118] Suraj Nair, Eric Mitchell, Kevin Chen, et al. 2021. Learning Language-Conditioned Robot Behavior from Offline Data and Crowd-Sourced Annotation. arXiv:2109.01115

[119] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, et al. 2021. WebGPT: Browser-assisted question-answering with human feedback. *arXiv* 2112.09332 (2021). arXiv:2112.09332 https://arxiv.org/abs/2112.09332

[120] Anjali Narayan-Chen, Prashant Jayannavar, and Julia Hockenmaier. 2019. Collaborative Dialogue in Minecraft. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*. Association for Computational Linguistics, 5405–5415. https://doi.org/10.18653/v1/p19-1537

[121] Humza Naveed, Asad Ullah Khan, Shi Qiu, et al. 2024. A Comprehensive Overview of Large Language Models. arXiv:2307.06435 https://arxiv.org/abs/2307.06435

[122] Dai Quoc Nguyen, Dat Quoc Nguyen, Cuong Xuan Chu, et al. 2017. Sequence to Sequence Learning for Event Prediction. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing, IJCNLP 2017, Taipei, Taiwan, November 27 - December 1, 2017, Volume 2: Short Papers*. Asian Federation of Natural Language Processing, 37–42. https://aclanthology.org/I17-2007/

[123] Taichi Nishimura, Atsushi Hashimoto, Yoshitaka Ushiku, et al. 2021. Structure-Aware Procedural Text Generation From an Image Sequence. *IEEE Access* 9 (2021), 2125–2141. https://doi.org/10.1109/ACCESS.2020.3043452

[124] Simon Ostermann, Ashutosh Modi, Michael Roth, et al. 2018. MCScript: A Novel Dataset for Assessing Machine Comprehension Using Script Knowledge. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. European Language Resources Association (ELRA). https://aclanthology.org/L18-1564

[125] Simon Ostermann, Michael Roth, and Manfred Pinkal. 2019. MCScript2.0: A Machine Comprehension Corpus Focused on Script Events and Participants. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*. Association for Computational Linguistics, 103–117. https://doi.org/10.18653/v1/S19-1012

[126] Jiefu Ou, Arda Uzunoglu, Benjamin Van Durme, et al. 2024. WorldAPIs: The World Is Worth How Many APIs? A Thought Experiment. arXiv:2407.07778 https://arxiv.org/abs/2407.07778

[127] Aishwarya Padmakumar, Jesse Thomason, Ayush Shrivastava, et al. 2022. TEACh: Task-Driven Embodied Agents That Chat. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*. AAAI Press, 2017–2025. https://ojs.aaai.org/index.php/AAAI/article/view/20097

[128] Matthew J Page, Joanne E McKenzie, Patrick M Bossuyt, et al. 2021. The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. *International journal of surgery* 88 (2021), 105906.

[129] Matthew J Page, David Moher, Patrick M Bossuyt, et al. 2021. PRISMA 2020 explanation and elaboration: updated guidance and exemplars for reporting systematic reviews. *bmj* 372 (2021).

[130] Liang-Ming Pan, Jingjing Chen, Jianlong Wu, et al. 2020. Multi-modal Cooking Workflow Construction for Food Recipes. In *Proceedings of the 28th ACM International Conference on Multimedia (MM '20)*. Association for Computing Machinery, 1132–1141. https://doi.org/10.1145/3394171.3413765

[131] Hogun Park and Hamid Reza Motahari-Nezhad. 2018. Learning Procedures from Text: Codifying How-to Procedures in Deep Neural Networks. In *Companion of the The Web Conference 2018 on The Web Conference 2018, WWW 2018, Lyon , France, April 23-27, 2018*. ACM, 351–358. https://doi.org/10.1145/3184558.3186347

[132] Justin Payan, Swaroop Mishra, Mukul Singh, et al. 2023. InstructExcel: A Benchmark for Natural Language Instruction in Excel. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*. Association for Computational Linguistics, 4026–4043. https://doi.org/10.18653/V1/2023.FINDINGS-EMNLP.265

[133] Xavier Puig, Kevin Ra, Marko Boben, et al. 2018. VirtualHome: Simulating Household Activities via Programs. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. Computer Vision Foundation / IEEE Computer Society, 8494–8502. https://doi.org/10.1109/CVPR.2018.00886

[134] Y. Qi, Q. Wu, P. Anderson, et al. 2020. REVERIE: Remote Embodied Visual Referring Expression in Real Indoor Environments. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 9979–9988. https://doi.org/10.1109/CVPR42600.2020.01000

[135] Chen Qian, Lijie Wen, Akhil Kumar, et al. 2020. An Approach for Process Model Extraction by Multi-grained Text Classification. In *Advanced Information Systems Engineering*. Springer International Publishing, 268–282.

[136] Dinesh Raghu, Shantanu Agarwal, Sachindra Joshi, et al. 2021. End-to-End Learning of Flowchart Grounded Task-Oriented Dialogs. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*. Association for Computational Linguistics, 4348–4366. https://doi.org/10.18653/v1/2021.emnlp-main.357

[137] Abhinav Rajvanshi, Karan Sikka, Xiao Lin, et al. 2023. SayNav: Grounding Large Language Models for Dynamic Planning to Navigation in New Environments. In *International Conference on Automated Planning and Scheduling*. https://api.semanticscholar.org/CorpusID:261660608

[138] Ramya Ramakrishnan, Ethan Elenberg, Hashan Narangodage, et al. 2023. Multi-Step Dialogue Workflow Action Prediction. *arXiv preprint arXiv:2311.09593* (2023).

[139] Michaela Regneri, Alex Koller, er, et al. 2010. Learning Script Knowledge with Web Experiments. 979–988.

[140] Ryokan Ri, Yufang Hou, Radu Marinescu, et al. 2022. Finding Sub-task Structure with Natural Language Instruction. 1–9. https://doi.org/10.18653/v1/2022.lnls-1.1

[141] Kyeongmin Rim, Jingxuan Tu, Bingyang Ye, et al. 2023. The Coreference under Transformation Labeling Dataset: Entity Tracking in Procedural Texts Using Event Models. 12448–12460. https://doi.org/10.18653/v1/2023.findings-acl.788

[142] Robin Rojowiec, Jana Götze, Philipp Sadler, et al. 2020. From "Before" to "After": Generating Natural Language Instructions from Image Pairs in a Simple Visual Domain. 316–326. https://doi.org/10.18653/v1/2020.inlg-1.38

[143] Michael Roth and Talita Anthonio. 2021. UnImplicit Shared Task Report: Detecting Clarification Requirements in Instructional Text. In *Proceedings of the 1st Workshop on Understanding Implicit and Underspecified Language*. Association for Computational Linguistics, 28–32. https://doi.org/10.18653/v1/2021.unimplicit-1.4

[144] Anisa Rula and Jennifer D'Souza. 2023. Procedural Text Mining with Large Language Models. In *Proceedings of the 12th Knowledge Capture Conference 2023 (K-CAP '23)*. Association for Computing Machinery, 9–16. https://doi.org/10.1145/3587259.3627572

[145] Keisuke Sakaguchi, Ch, ra Bhagavatula, et al. 2021. proScript: Partially Ordered Scripts Generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*. Association for Computational Linguistics, 2138–2149. https://doi.org/10.18653/v1/2021.findings-emnlp.184

[146] Amaia Salvador, Nicholas Hynes, Yusuf Aytar, et al. 2017. Learning Cross-Modal Embeddings for Cooking Recipes and Food Images. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 3068–3076. https://doi.org/10.1109/CVPR.2017.327

[147] Rosario Scalise, Shen Li, Henny Admoni, et al. 2018. Natural language instructions for human–robot collaborative manipulation. *The International Journal of Robotics Research* 37 (2018), 558 – 565. https://api.semanticscholar.org/CorpusID:49380266

[148] Roger C Schank and Robert P Abelson. 2013. *Scripts, plans, goals, and understanding: An inquiry into human knowledge structures.* Psychology press.

[149] Thomas Scialom, Sylvain Lamprier, Benjamin Piwowarski, et al. 2019. Answers Unite! Unsupervised Metrics for Reinforced Summarization Models. 3246–3256. https://doi.org/10.18653/v1/D19-1320

[150] Procheta Sen, Xi Wang, Ruiqing Xu, et al. 2023. Task2KB: A Public Task-Oriented Knowledge Base. arXiv:2302.00654

[151] Yutong Shao and Ndapa Nakashole. 2020. ChartDialogs: Plotting from Natural Language Instructions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 3559–3574. https://doi.org/10.18653/v1/2020.acl-main.328

[152] Haochen Shi, Zhiyuan Sun, Xingdi Yuan, et al. 2024. OPEx: A Component-Wise Analysis of LLM-Centric Agents in Embodied Instruction Following. 622–636. https://doi.org/10.18653/v1/2024.acl-long.37

[153] Qi Shi, Qian Liu, Bei Chen, et al. 2022. LEMON: Language-Based Environment Manipulation via Execution-Guided Pre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*. Association for Computational Linguistics, 471–485. https://doi.org/10.18653/V1/2022.FINDINGS-EMNLP.33

[154] Tianlin Shi, Andrej Karpathy, Linxi Fan, et al. 2017. World of Bits: An Open-Domain Platform for Web-Based Agents. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017 (Proceedings of Machine Learning Research, Vol. 70)*. PMLR, 3135–3144. http://proceedings.mlr.press/v70/shi17a.html

[155] Mohit Shridhar, Jesse Thomason, Daniel Gordon, et al. 2020. ALFRED: A Benchmark for Interpreting Grounded Instructions for Everyday Tasks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. https://arxiv.org/abs/1912.01734

[156] Bharat Singh, Rajesh Kumar, and Vinay Pratap Singh. 2022. Reinforcement learning in robotic applications: a comprehensive survey. *Artif. Intell. Rev.* 55, 2 (2022), 945–990. https://doi.org/10.1007/s10462-021-09997-9

[157] Himanshu Gaurav Singh, Vishal Bindal, Arnav Tuli, et al. 2022. Learning Neuro-symbolic Programs for Language-Guided Robotic Manipulation. https://api.semanticscholar.org/CorpusID:260706314

[158] Chan Hee Song, Jiaman Wu, Clayton Washington, et al. 2023. LLM-Planner: Few-Shot Grounded Planning for Embodied Agents with Large Language Models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2998–3009.

[159] Kavya Srinet, Yacine Jernite, Jonathan Gray, et al. 2020. CraftAssist Instruction Parsing: Semantic Parsing for a Voxel-World Assistant. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 4693–4714. https://doi.org/10.18653/v1/2020.acl-main.427

[160] Maurício Steinert and Felipe Rech Meneguzzi. 2020. Planning domain generation from natural language step-by-step instructions. In *2020 Workshop on Knowledge Engineering for Planning and Scheduling (KEPS@ ICAPS), 2020, França.*

[161] Shane Storks, Qiaozi Gao, Yichi Zhang, et al. 2021. Tiered Reasoning for Intuitive Physics: Toward Verifiable Commonsense Language Understanding. 4902–4918. https://doi.org/10.18653/v1/2021.findings-emnlp.422

[162] Carl Strathearn and Dimitra Gkatzia. 2022. Task2Dial: A Novel Task and Dataset for Commonsense-enhanced Task-based Dialogue Grounded in Documents. 187–196. https://doi.org/10.18653/v1/2022.dialdoc-1.21

[163] Chenkai Sun, Tie Xu, ChengXiang Zhai, et al. 2023. Incorporating Task-Specific Concept Knowledge into Script Learning. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*. 3026–3040. https://doi.org/10.18653/v1/2023.eacl-main.220

[164] Liangtai Sun, Xingyu Chen, Lu Chen, et al. 2022. META-GUI: Towards Multi-modal Conversational Agents on Mobile GUI. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*. Association for Computational Linguistics, 6699–6712. https://doi.org/10.18653/V1/2022.EMNLP-MAIN.449

[165] T, Niket on, Bhavana Dalvi, et al. 2019. WIQA: A dataset for "What if..." reasoning over procedural text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, 6076–6085. https://doi.org/10.18653/v1/D19-1629

[166] T, Niket on, Keisuke Sakaguchi, et al. 2020. A Dataset for Tracking Entities in Open Domain Procedural Text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 6408–6417. https://doi.org/10.18653/v1/2020.emnlp-main.520

[167] Chen Tang, Ben Abbatematteo, Jiaheng Hu, et al. 2024. Deep Reinforcement Learning for Robotics: A Survey of Real-World Successes. arXiv:2408.03539 https://arxiv.org/abs/2408.03539

[168] Stefanie Tellex, Thomas Kollar, Steven Dickerson, et al. 2011. Approaching the Symbol Grounding Problem with Probabilistic Graphical Models. *AI Mag.* 32 (2011), 64–76. https://api.semanticscholar.org/CorpusID:8477711

[169] Stefanie Tellex, Thomas Kollar, Steven Dickerson, et al. 2011. Understanding Natural Language Commands for Robotic Navigation and Mobile Manipulation. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, August 7-11, 2011*. AAAI Press, 1507–1514. https://doi.org/10.1609/AAAI.V25I1.7979

[170] Shubham Toshniwal, Sam Wiseman, Karen Livescu, et al. 2021. Chess as a Testbed for Language Model State Tracking. In *AAAI Conference on Artificial Intelligence*. https://api.semanticscholar.org/CorpusID:248811258

[171] Daniel Toyama, Philippe Hamel, Anita Gergely, et al. 2021. AndroidEnv: A Reinforcement Learning Platform for Android. *arXiv* 2105.13231 (2021). arXiv:2105.13231 https://arxiv.org/abs/2105.13231

[172] Arda Uzunoglu, Gözde Şahin, and Abdulfattah Safa. 2024. PARADISE: Evaluating Implicit Planning Skills of Language Models with Procedural Warnings and Tips Dataset. In *Findings of the Association for Computational Linguistics ACL 2024*. 10085–10102. https://doi.org/10.18653/v1/2024.findings-acl.599

[173] Arda Uzunoglu and Gözde Gül Sahin. 2023. Benchmarking Procedural Language Understanding for Low-Resource Languages: A Case Study on Turkish. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, IJCNLP 2023 -Volume 1: Long Papers, Nusa Dua, Bali, November 1 - 4, 2023*. Association for Computational Linguistics, 804–819. https://doi.org/10.18653/V1/2023.IJCNLP-MAIN.52

[174] Adam Vogel and Daniel Jurafsky. 2010. Learning to Follow Navigational Directions. In *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden*. The Association for Computer Linguistics, 806–814. https://aclanthology.org/P10-1083/

[175] Hao Wang, Guosheng Lin, Steven CH Hoi, et al. 2022. Learning structural representations for recipe generation and food retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 3 (2022), 3363–3377.

[176] Haoyu Wang, Hongming Zhang, Yueguan Wang, et al. 2023. Are All Steps Equally Important? Benchmarking Essentiality Detection in Event Processes. 4048–4056. https://doi.org/10.18653/v1/2023.emnlp-main.246

[177] Jiaqi Wang, Zihao Wu, Yiwei Li, et al. 2024. Large Language Models for Robotics: Opportunities, Challenges, and Perspectives. *arXiv* 2401.04334 (2024). arXiv:2401.04334 https://arxiv.org/abs/2401.04334

[178] Lei Wang, Chen Ma, Xueyang Feng, et al. 2024. A survey on large language model based autonomous agents. *Frontiers Comput. Sci.* 18, 6 (2024), 186345. https://doi.org/10.1007/S11704-024-40231-1

[179] Lilian Wanzare, Aless Zarcone, ra, et al. 2017. Inducing Script Structure from Crowdsourced Event Descriptions via Semi-Supervised Clustering. 1–11. https://doi.org/10.18653/v1/W17-0901

[180] Lilian D. A. Wanzare, Aless, ra Zarcone, et al. 2016. A Crowdsourced Database of Event Sequence Descriptions for the Acquisition of High-quality Script Knowledge. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. European Language Resources Association (ELRA).

[181] Jason Weston, Antoine Bordes, Sumit Chopra, et al. 2016. Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*. http://arxiv.org/abs/1502.05698

[182] Wikipedia. 2024. Business Process Model and Notation.

[183] Gu Wu, e, Jing Qian, et al. 2024. ARTiST: Automated Text Simplification for Task Guidance in Augmented Reality. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24)*. Association for Computing Machinery. https://doi.org/10.1145/3613904.3642772

[184] Te-Lin Wu, Caiqi Zhang, Qingyuan Hu, et al. 2023. Learning Action Conditions from Instructional Manuals for Instruction Understanding. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*. Association for Computational Linguistics, 3023–3043. https://doi.org/10.18653/V1/2023.ACL-LONG.170

[185] Xueqing Wu, Sha Li, and Heng Ji. 2023. OpenPI-C: A Better Benchmark and Stronger Baseline for Open-Vocabulary State Tracking. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*. Association for Computational Linguistics, 7213–7222. https://doi.org/10.18653/V1/2023.FINDINGS-ACL.452

[186] Yue Wu, Xuan Tang, Tom M. Mitchell, et al. 2023. SmartPlay : A Benchmark for LLMs as Intelligent Agents. *arXiv* 2310.01557 (2023). https://doi.org/10.48550/ARXIV.2310.01557 arXiv:2310.01557

[187] Wei Xiang and Bang Wang. 2019. A Survey of Event Extraction From Text. *IEEE Access* 7 (2019), 173111–173137. https://doi.org/10.1109/ACCESS.2019.2956831

[188] Nancy Xu, Sam Masling, Michael Du, et al. 2021. Grounding Open-Domain Instructions to Automate Web Support Tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*. Association for Computational Linguistics, 1022–1032. https://doi.org/10.18653/V1/2021.NAACL-MAIN.80

[189] Semih Yagcioglu, Aykut Erdem, Erkut Erdem, et al. 2018. RecipeQA: A Challenge Dataset for Multimodal Comprehension of Cooking Recipes. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*. Association for Computational Linguistics, 1358–1368. https://doi.org/10.18653/v1/d18-1166

[190] Yue Yang, Artemis Panagopoulou, Qing Lyu, et al. 2021. Visual Goal-Step Inference using wikiHow. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2167–2179. https://doi.org/10.18653/v1/2021.emnlp-main.165

[191] Shunyu Yao, Howard Chen, John Yang, et al. 2022. WebShop: Towards Scalable Real-World Web Interaction with Grounded Language Agents. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*. http://papers.nips.cc/paper_files/paper/2022/hash/82ad13ec01f9fe44c01cb91814fd7b8c-Abstract-Conference.html

[192] Siyu Yuan, Jiangjie Chen, Ziquan Fu, et al. 2023. Distilling Script Knowledge from Large Language Models for Constrained Language Planning. 4303–4325. https://doi.org/10.18653/v1/2023.acl-long.236

[193] Daoguang Zan, Bei Chen, Fengji Zhang, et al. 2023. Large Language Models Meet NL2Code: A Survey. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*. Association for Computational Linguistics, 7443–7464. https://doi.org/10.18653/V1/2023.ACL-LONG.411

[194] Xiaoxue Zang, Ashwini Pokle, Marynel Vázquez, et al. 2018. Translating Navigation Instructions in Natural Language to a High-Level Plan for Behavioral Robot Navigation. 2657–2666. https://doi.org/10.18653/v1/D18-1286

[195] Rowan Zellers, Ari Holtzman, Yonatan Bisk, et al. 2019. HellaSwag: Can a Machine Really Finish Your Sentence? 4791–4800. https://doi.org/10.18653/v1/P19-1472

[196] Qingtian Zeng, Xiaojie Tang, Weijian Ni, et al. 2020. Missing Procedural Texts Repairing Based on Process Model and Activity Description Templates. *IEEE Access* 8 (2020), 12999–13010. https://doi.org/10.1109/ACCESS.2020.2965160

[197] Qingtian Zeng, Xiaojie Tang, Weijian Ni, et al. 2020. Missing Procedural Texts Repairing Based on Process Model and Activity Description Templates. *IEEE Access* 8 (2020), 12999–13010. https://doi.org/10.1109/ACCESS.2020.2965160

[198] Danyang Zhang, Zhennan Shen, Rui Xie, et al. 2023. Mobile-Env: Building Qualified Evaluation Benchmarks for LLM-GUI Interaction. *arXiv* 2305.08144 (2023). arXiv:2305.08144 https://arxiv.org/abs/2305.08144

[199] Hongming Zhang, Muhao Chen, Haoyu Wang, et al. 2020. Analogous Process Structure Induction for Sub-event Sequence Prediction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 1541–1550. https://doi.org/10.18653/v1/2020.emnlp-main.119

[200] Jingwei Zhang, Xin Wu, and Yi Cai. 2023. CLEVR-Implicit: A Diagnostic Dataset for Implicit Reasoning in Referring Expression Comprehension. 12820–12830. https://doi.org/10.18653/v1/2023.emnlp-main.791

[201] Li Zhang. 2022. Reasoning about Procedures with Natural Language Processing: A Tutorial. *arXiv* 2205.07455 (2022). https://doi.org/10.48550/arXiv.2205.07455 arXiv:2205.07455

[202] Li Zhang, Qing Lyu, and Chris Callison-Burch. 2020. Reasoning about Goals, Steps, and Temporal Ordering with WikiHow. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*. Association for Computational Linguistics, 4630–4639. https://doi.org/10.18653/v1/2020.emnlp-main.374

[203] Li Zhang, Hainiu Xu, Abhinav Kommula, et al. 2024. OpenPI2.0: An Improved Dataset for Entity Tracking in Texts. 166–178.

[204] Li Zhang, Hainiu Xu, Abhinav Kommula, et al. 2024. OpenPI2.0: An Improved Dataset for Entity Tracking in Texts. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2024 - Volume 1: Long Papers, St. Julian's, Malta, March 17-22, 2024*. Association for Computational Linguistics, 166–178. https://aclanthology.org/2024.eacl-long.10

[205] Li Zhang, Hainiu Xu, Yue Yang, et al. 2023. Causal Reasoning of Entities and Events in Procedural Texts. In *Findings of the Association for Computational Linguistics: EACL 2023*. Association for Computational Linguistics, 415–431. https://aclanthology.org/2023.findings-eacl.31

[206] Shengyu Zhang, Linfeng Dong, Xiaoya Li, et al. 2023. Instruction Tuning for Large Language Models: A Survey. *arXiv preprint arXiv:2308.10792* (2023).

[207] Tianyi Zhang, Li Zhang, Zhaoyi Hou, et al. 2024. PROC2PDDL: Open-Domain Planning Representations from Texts. 13–24.

[208] Yi Zhang, Sujay Kumar Jauhar, Julia Kiseleva, et al. 2021. Learning to Decompose and Organize Complex Tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2726–2735. https://doi.org/10.18653/v1/2021.naacl-main.217

[209] Yizhe Zhang, Jiarui Lu, and Navdeep Jaitly. 2024. Probing the Multi-turn Planning Capabilities of LLMs via 20 Question Games. *ACL*.

[210] Yixin Zhang, Yoko Yamakata, and Keishi Tajima. 2021. Supplementing Omitted Named Entities in Cooking Procedural Text with Attached Images. In *2021 IEEE 4th International Conference on Multimedia Information Processing and Retrieval (MIPR)*. 199–205. https://doi.org/10.1109/MIPR51284.2021.00037

[211] Yixin Zhang, Yoko Yamakata, and Keishi Tajima. 2022. MIAIS: A Multimedia Recipe Dataset with Ingredient Annotation at Each Instructional Step. In *Proceedings of the 1st International Workshop on Multimedia for Cooking, Eating, and Related APPlications (CEA++ '22)*. Association for Computing Machinery, 49–52. https://doi.org/10.1145/3552485.3554938

[212] Zhihan Zhang, Xiubo Geng, Tao Qin, et al. 2021. Knowledge-Aware Procedural Text Understanding with Multi-Stage Training. In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*. ACM.

[213] Yingxiu Zhao, Bowen Yu, Bowen Li, et al. 2023. Causal Document-Grounded Dialogue Pre-training. In *EMNLP:2023:main*. 7160–7174. https://doi.org/10.18653/v1/2023.emnlp-main.443

[214] Luowei Zhou, Chenliang Xu, and Jason J Corso. 2018. Towards Automatic Learning of Procedures From Web Instructional Videos. In *AAAI Conference on Artificial Intelligence*. 7590–7598. https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17344

[215] Shuyan Zhou, Frank F. Xu, Hao Zhu, et al. 2023. WebArena: A Realistic Web Environment for Building Autonomous Agents. *arXiv* 2307.13854 (2023). https://doi.org/10.48550/ARXIV.2307.13854 arXiv:2307.13854

[216] Shuyan Zhou, Pengcheng Yin, and Graham Neubig. 2022. Hierarchical Control of Situated Agents through Natural Language. 67–84. https://doi.org/10.18653/v1/2022.suki-1.8

[217] Shuyan Zhou, Li Zhang, Yue Yang, et al. 2022. Show Me More Details: Discovering Hierarchies of Procedures from Semi-structured Web Data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2998–3012. https://doi.org/10.18653/v1/2022.acl-long.214

[218] Yilun Zhou, Julie Shah, and Steven Schockaert. 2019. Learning Household Task Knowledge from WikiHow Descriptions. In *Proceedings of the 5th Workshop on Semantic Deep Learning (SemDeep-5)*. Association for Computational Linguistics, 50–56. https://aclanthology.org/W19-5808

[219] Yuqi Zhu, Shuofei Qiao, Yixin Ou, et al. 2024. KnowAgent: Knowledge-Augmented Planning for LLM-Based Agents. *arXiv preprint arXiv:2403.03101* (2024).

A DATABASE SPECIFIC KEYWORD SETTINGS

The query includes thirteen sub-queries focusing on frequent terms in procedural text research. Initially, we used broad keywords like “procedural text” and “instructional text,” then refined the search to cover common corpora, instruction forms, and tasks. The full list is given in Table 4.

<table>
<tr><td>Google Scholar, IEEE Xplore</td><td>DBLP</td></tr>
<tr><td>Exact Phrase Match (PM)</td><td>Exact Match per Word (EMW)</td></tr>
<tr><td colspan="2">“procedural text”
“instructional text”
“natural language instruction(s)”
“entity tracking”
“entity state tracking”
“wikihow”
“script knowledge”
“BPMN generation”
“action sequence natural language”
“action sequence annotation”
“instruction parsing”
“instruction manual” AND “parsing”
“recipe parsing”</td></tr>
</table>

Table 4. Keywords

B EXAMPLE OF EVENT CENTRIC REPRESENTATIONS

Fig. 9 illustrates the diversity of structures used to represent the process of baking a cake through three different formalisms. While all three approaches capture the same goal, the way in which they structure the actions and entities involved is distinct. Fig. 10 shows an example of BPMN for a “Sending an issue list” process.

C LIST OF GROUNDED AND UNGROUNDED TASK PAPERS

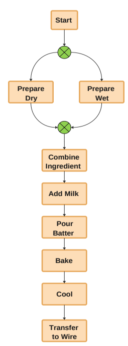
List of all papers for grounded and ungrounded tasks are given in Table 5.

Ingredients:
1- 1 ½ cups all-purpose flour
2- 1 ½ teaspoons baking powder
3- ½ teaspoon salt
4- ½ cup unsalted butter, softened
5- 1 cup granulated sugar
6- 2 large eggs
7- 1 teaspoon vanilla extract
8- ½ cup milk
Instructions:
1. Prepare dry ingredients: Mix flour, baking powder, and salt.
2. Prepare wet ingredients: Cream butter and sugar, add eggs, and vanilla.
3. Combine ingredients: Gradually add dry ingredients to wet ingredients while mixing.
4. Add Milk.
5. Pour batter into a greased 8-inch cake pan.
6. Bake for 25-30 minutes at 350°F (175°C).
7. Let cool for 10 minutes
8. Transfer to a wire rack.

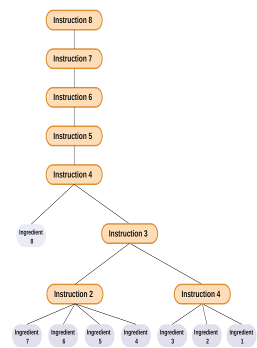
(a) Recipe Text

Prepare (1- $1\frac{1}{2}$ cups all-purpose flour, $1\frac{1}{2}$ teaspoons baking powder, $\frac{1}{2}$ teaspoon salt)
→
Prepare ($\frac{1}{2}$ cup unsalted butter, softened, 1 cup granulated sugar, 2 large eggs, 1 teaspoon vanilla extract)
→
Combine (ingredients)
→
Add ($\frac{1}{2}$ cup milk)
→
Pour (batter, into greased 8-inch cake pan)
→
Bake (25-30 minutes at 350°F)
→
Cool (10 minutes)
→
Transfer (to wire rack)

(b) Recipe Representation in Wiki HG [48]



(c) Recipe Representation as Process Graph [135]



(d) Recipe Representation in SIMMR [62]

Fig. 9. Vanilla cake recipe representations: (a) Recipe Text, (b) Text Representation from Wiki HG [48], (c) Process Graph [135], and (d) SIMMR structure [62].

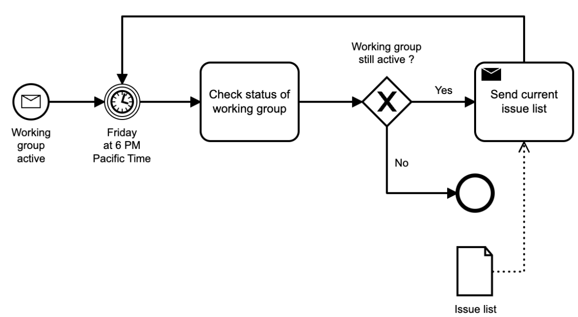


Fig. 10. Example of a Business Process Model and Notation for “Sending an issue list” process [182]

Category	Task/Environment	Papers
Ungrounded	Summarization	[17, 39, 76, 81, 83, 83, 149, 155]
	Event Alignment	[12, 36, 43, 78, 92, 102, 104, 115, 139, 149, 155, 179, 180, 214]
	Detecting and Correcting Implicit Instructions	[4, 13, 29, 31, 47, 143, 197, 197, 200, 210]
	Entity State Tracking	[18, 26, 29, 31, 36, 36, 37, 51, 68, 73, 85, 96, 115, 116, 141, 166, 203, 203, 205, 212]
	Instruction Parsing	[8, 10, 11, 33, 42, 44, 48, 53, 59, 70, 77, 79, 82, 94, 94, 100, 107, 108, 114, 117, 130, 131, 135, 139, 139, 140, 144, 146, 154, 175, 179, 179, 180, 180, 184, 199, 203, 207, 208, 211, 217]
	Process Generation	[71, 95, 97, 98, 112, 122, 123, 142, 145, 145, 163, 180, 192, 199]
	Question Answering	[9, 15, 124, 125, 161, 165, 172, 176, 181, 189, 190, 195, 202, 209, 218]
	Knowledge Acquisition/Mining	[6, 28, 30, 67, 108, 150, 160]
Grounded	Web	[40, 41, 52, 72, 94, 105, 119, 154, 188, 191, 215, 219]
	Simulations/Navigation	[3, 5, 38, 47, 87, 89, 99, 109–111, 133, 134, 137, 152, 174, 183, 194, 216]
	Robotic	[1, 14, 21, 56, 103, 106, 118, 147, 157, 168, 169]
	Game	[55, 57, 61, 65, 80, 86, 101, 120, 170, 186]
	GUI/Mobile	[7, 22, 90, 171, 198]
	Dialogue	[2, 25, 45, 46, 50, 64, 113, 120, 127, 136, 151, 159, 162, 164]

Table 5. Ungrounded and Grounded Tasks Papers

D GROUNDED ENVIRONMENT DETAILS

We have identified the following divergences across environments: *data instance* (typical components of the dataset), *dataset size*, *the action space*, *tasks performed* in the *environment*. The end tasks vary depending on the environment. For instance, simulating user actions on websites, such as filling forms or navigating menus [40, 191, 215] are common in Web environments, while gaming scenarios often involve character navigation and object interaction within virtual worlds [80, 120, 186, 186]. A summary is given in Table 6.

	Environment	Data instance	Task	Num of Actions	Size
PixelHelp [90]	Google Phone	Screen, instruction, grounded UI actions	Account configuration, Gmail, Chrome, Photos tasks	2-8 steps	187
WinHelp [19]	Windows 2000 VM	Screen, instruction, grounded UI actions	Windows troubleshooting	10,3	128
AndroidHowTo [90]	Google Phone	Screen, instruction, grounded UI actions	Account configuration, Gmail, Chrome, Photos tasks	2-8 steps	187
AndroidEnv [171]	Mobile	Emulated Android devices, touchscreen gestures, real-time interaction	UI navigation, game playing, and basic utilities	Varies	100 (tasks)
WebArena [215]	Web	Screenshot/HTML Dom Tree/Accessibility Tree, Multitab, (high-level) instruction, grounded UI actions	web-based tasks for e-commerce, social forums, collaborative software and content management	Varies	812
Mind2Web [40]	Web	Real-world websites, instructions, and action sequences	variety of tasks across different domains on any website	Average = 7.3	2,350 (from 137 websites across 31 domains)
World Of Bits [154]	Web	Screen pixels, DOM, keyboard and mouse actions	web-based tasks via natural language queries	Varies	100 (web tasks)
WOB++ [94]	Web	DOM elements, workflow steps (Click, Type)	Email processing, form filling, etc., in noisy environments	Varies	80 tasks
CrossBlock [94]	Grid	Grid state, instruction, action	Clear the grid by drawing lines	5,86	50
Hexagons (Draw me a flower) [80]	Grid	Grid state, instruction, action	Fill the grids with colors	6,73	620
Chess [170]	Grid	Chessboard state, instructions	Predicting legal chess moves	Varies	2.5M scenarios
SCONE [96]	Simulation	Instructions, Logical form	Infer the final environment state following the actions	5	4K
VirtualHome [133]	3D Simulation	Instructions, Scratch Program, Video	Household	11,6	2821
LANI, CHAI [110]	3D World w/Landmarks	Instructions, world states, actions	Navigation/Househould	4,7/7,7	6000/1596
CALVIN [106]	Robotics/Simulation	language instructions, multimodal sensors	Long-horizon tasks (e.g., open drawer, push block)	~5-10	20K language directives
Bucker et al. [21]	Robotics/Simulation	Predefined trajectories, object positions, natural language commands	Trajectory reshaping based on natural language commands	~ 5-10 steps	~ 10,000 trajectory labels
LOReL [118]	Robotics/Simulation	Vision-based interaction, natural language commands, multimodal sensors	Language-conditioned manipulation (e.g., open drawer, move stapler)	up to 150	53,000 scenarios
Meta-GUI [164]	Dialogue/Web	Screenshots, HTML Dom Tree/Accessibility Tree, touch and text inputs	Multi-modal conversational interactions (e.g., booking a hotel, checking the weather)	~ 4-5	1125 dialogues

Table 6. Grounded instructions. Screen: The structured UI state/tree not pixels.

E GROUNDED TASKS ORIENTED DIALOGUES DATASETS

Table 7 showcases a variety of datasets used in grounded task-oriented dialogue research, illustrating diverse tasks ranging from single subtasks, such as Knowledge Identification (which can be mentioned with different names, including User Utterance Grounding, Grounding Span Prediction, Flow-chart Retrieval) to full pipeline tasks like Cascade Dialogue Success, and employing different metrics.

Dataset	Docs	Dialogues	Evaluation			
			Task	Metric	Baseline	SoTA
Doc2Dial	480	4470	Knowledge Identification	EM	55.4	68.4 [34]
				F1	66.6	88.7 [35]
MultiDoc2Dial	488	4796	Knowledge Identification	EM	26.6	51.0 [213]
				F1	43.7	64.5 [213]
Task2Dial	353	478	-	-	-	-
CookDial	260	260	User Question Understanding	Accuracy	94.5	-
				F1	91.0	-
ABCD	55	10042	Cascade Dialgoue Success	Cascading Eval	31.9	60.7 [138]
FLODial	12	5,476	Knowledge Identification	Success Rate	33.7	67.1 [58]
				R@1	91.0	-
TEACh	12	3,047	Execution from Dialogue History	SR	7.06	18.60 [60]
				GC	9.57	18.60 [60]
			Trajectory from Dialogue	SR	0.51	-
				GC	20.3	-
			Two-Agent Task Completion	SR	24.4	-

Table 7. Commonly Used Grounded Task-Oriented Dialogue Datasets

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009