# Extracting Incomplete Planning Action Models from Unstructured Social Media Data to Support Decision Making

**Lydia Manikonda**[1], **Shirin Sohrabi**[2], **Kartik Talamadupula**[2], **Biplav Srivastava**[2], **Subbarao Kambhampati**[1]
[1]School of Computing, Informatics, and Decision Systems Engineering, Arizona State University
[2]IBM T. J. Watson Research Center
{lmanikonda, rao}@asu.edu, {ssohrab, krtalamad, biplavs}@us.ibm.com

### Abstract

Despite increasing interest in leveraging the wealth of online social media data to support data-based decision making, much work in this direction has focused on tasks with straightforward "labeling" decisions. A much richer class of tasks can benefit from the power of sequential decision making. However, supporting such tasks requires learning some form of action or decision models from unstructured data – a problem that had not received much attention. This paper leverages and extends machine learning techniques to learn decision models (incomplete action models) for planning from unstructured social media data. We provide evaluations showing the potential of unstructured data to build incomplete planning action models, which can further be extended to build PDDL-style action models for many real-world domains. Our models can be used to support novel quantitative analysis of online behaviors that can indirectly explain the offline behaviors of social media users.

## 1 Introduction

There is a growing interest in exploiting the burgeoning amount of user-generated data on the Internet – especially on social media platforms – to provide data-based decision support. While the initial wave of work in this direction was limited to support single *labeling* decisions (e.g. recommendations), there is an increasing interest in supporting more complex scenarios that require planning and other forms of sequential decision making. A prime example is the category of tasks that are classified as "self-help", and which involve a number of steps and often complicated sequences of actions. Examples here include quitting smoking, losing weight, or traveling the world. A number of online groups contain a plethora of crowd-generated wisdom about appropriate courses of action that have worked for a variety of different individuals. The main problem we consider in this paper is the extraction of such information so that it can be applied towards an automated way of helping new users with similar goals. Such automated approaches need not be restricted to plan synthesis alone; they can also include a number of other sequential decision making problems including plan critiquing, plan ranking, and even merely the extraction of plan traces that can be used as input to existing model-learning methods.

Although there exists a large body of literature on planning and decision making, almost all of it assumes that the action model has been specified *a priori*. This has turned into a very pressing bottleneck for the AI planning community as a whole, where planning techniques depend very heavily on the availability of *complete and correct* models (Kambhampati 2007). One of the challenges that must be overcome to tide over this problem is to *extract* usable causal relationships from unstructured natural language data on social media. This can be a very daunting problem, since social media posts are made in unrestricted natural language, and meant for human consumption. The text from these posts can be highly nuanced and extremely arbitrary, making automated extraction of causal relationships and action models an AI-complete task.

While parsing individual posts can be arbitrarily hard, our aim is to investigate if the massive scale and redundancy of the posts might nevertheless help extract reasonable approximations of causal and action models. We hypothesize that the feasibility of this endeavor might increase if we further focus on the so-called *shallow* models (c.f. (Kambhampati 2007; Tian, Zhuo, and Kambhampati 2016)). To this end, we propose and experiment with a six-phase pipeline that leverages shallow natural language processing (NLP) techniques to extract incomplete causal relationships. We envision that these relationships can be extended to generate complete PDDL-style domain models, in the spirit of (Srivastava and Kambhampati 2005) – however, this specifically is not the main focus of this paper.

As mentioned earlier, such approximate causal models can be utilized to automatically explain (c.f. *plan explanation, plan critiquing*) the experiences shared by users on online social media towards achieving their personal goals. In order to achieve this, our proposed pipeline addresses five main tasks: (1) *extract* actions from users' posts; (2) *process* the extracted actions to reduce redundancy; (3) *build* plan traces from the extracted actions; (4) *construct* an action precedence graph from these traces; and (5) *plan* using these precedence graphs. While we focus on an end-to-end solution, mid-stream output from our pipeline—e.g. plan traces—can also be fed to existing approaches for learning action models from complete, partial, or noisy plan traces (c.f. (Yang, Wu, and Jiang 2007; Tian, Zhuo, and Kambhampati 2016)). We evaluate the plans – which are represented as shallow workflows – to demonstrate the utility of

| subreddit | Main goal |
|---|---|
| **(/r/stopsmoking)** | How to *quit* smoking ? |
| **(/r/C25K)** | With no experience of running, how to *run* a 5K ? |
| **(/r/weddingplanning)** | How to *plan* for a wedding ? |

Table 1: Subreddits and their main goals

our novel six-phase pipeline. Within the current context, we define (shallow) *workflows* as a sequence of actions where the final action in the sequence is/achieves the user's goal.

In the rest of this paper, we will describe the details of our proposed pipeline; we first focus on explaining the data in Section 2. Details of the six-phase pipeline that we implemented, including the metrics for evaluating the action models we extracted to support sequential decision problems, are presented in Section 3. Section 4 describes the evaluation methodology and the results obtained through quantitative and qualitative analyses. Section 5 presents the related work focusing on how the existing literature and the proposed solution through the pipeline are different. Section 6 concludes the paper with a discussion on future work.

## 2 Data

In this paper, we utilize social media data to identify the important actions which are described by the users trying to achieve a goal. Towards this goal, we consider posts from the popular social news website called "Reddit" (https://www.reddit.com/) where the registered users submit content in different forms like web urls or text posts. Along with sharing content, users can comment and vote on a given post (up or down votes) that determines the popularity or rank of a post in a given thread. The content entries on this platform are designed in a tree format where each branch of a tree represents a sub-community referred as "Subreddit". Each subreddit is categorized to a particular domain that ranges from being very general to sometimes very personal.

We used the Python Wrapper for Reddit API[1] to crawl posts and their metadata from three different subreddits shown in Table 1. For the ease of reading, we represent the subreddit '/r/stopsmoking' as *Quit Smoking*; '/r/C25K' as *Couch to 5K*; '/r/weddingplanning' as *Wed. Planning*. Note that the entire pipeline is automated and there is no manual intervention in any of the processes. Table 2 provides the relevant statistics about the raw dataset and the actions extracted by the pipeline to build an action model.

## 3 Pipeline

We utilize the automated planning and NLP techniques to build a six-phase pipeline (as shown in Figure 1). This pipeline utilizes the raw unstructured social media data to extract structured shallow workflows. The main contributions or the challenges addressed by this pipeline are: 1) extract the plan traces from the raw unstructured data, 2) utilize the plan traces for building an incomplete action model that are capable of generating workflows that are near optimal. Our main contribution lies in considering the unstructured

[1]https://praw.readthedocs.io/en/latest/index.html

| | Domain Name | | |
|---|---|---|---|
| | **Quit Smoking** | **Couch to 5K** | **Wed. Planning** |
| **# Users** | 787 | 604 | 969 |
| **Tot. # of traces** | 1598 | 1131 | 3442 |
| **Avg. trace len.** | 17.97 | 16.7 | 21.29 |
| **# Unique Actions (orig)** | 1712 | 1299 | 2666 |
| **# Unique Actions (model)** | 234 | 194 | 355 |
| **# Pre-actions** | 1499 | 1060 | 2795 |
| | (117,6.4,16.9) | (84,5.5,13.8) | (170,7.9,22.5) |
| **# Post-actions** | 1398 | 982 | 2619 |
| | (31,6,6.5) | (29,5.1,5.6) | (37,7.4,8.2) |

Table 2: Statistics of users, plan traces and actions. Numbers in bracket are max, avg and std. dev.; min=1; Unique Actions (orig) are the set of actions that are extracted from the crawled raw data; Unique Actions (model) are the set of actions obtained after generalization.

social media data and building shallow models which are capable of generating plans that are near optimal.

Achieving the first goal is an important contribution of this paper, as most of the existing work (e.g., (Gregory and Lindsay 2016; Tian, Zhuo, and Kambhampati 2016; Yang, Wu, and Jiang 2007; Yoon 2007)) for domain model acquisition, assume that the plan traces required are readily available. Hence, these systems may not be functional when the traces are not available. To address the first challenge we mentioned earlier, the pipeline utilizes raw unstructured social media data that is processed to remove redundancies and repetitions to extract plan traces in lifted representations. To address the second challenge, these plan traces are utilized to compute the probabilities which determine the causal relationships between actions to finally build an incomplete action model. For each of the three domains we described in Section 2, we automatically extract the important actions to build a shallow model that is used to generate workflows.

The pipeline consists of six different components which are executed sequentially. The six different components of this pipeline shown in Figure 1 consists of: (1) fragment extractor - filtering the available data or posts to find the relevant posts, given a particular goal; (2) action extractor - identifying the candidate list of action names and their parameters as well as the initial plan fragment; (3) generalizer - grouping similar action names into the same cluster; (4) trace builder - converting the posts into plan traces; (5) sequential probability learner: learning the ordering among actions; (6) model validator: validating the extracted model. More details about each of the component are explained in detail below along with a running example.

### 3.1 Phase-1: Fragment Extractor

The main goal of this component is to extract the fragments from the corresponding subreddit. We define fragment as the relevant post that contains information about achieving the given goal of the subreddit. An individual fragment may contain more than one action that helps achieve the goal. To do this, we first crawl the individuals who are actively participating on a subreddit associated with a given goal. We


Figure 1: Six-phase pipeline

crawl the timelines of these individuals that we assume are the sequence of actions or a workflow that helps these individuals to achieve the given goal. We define *timeline* as the set of goal-related posts shared by the same user chronologically.

*Running example [relevant posts]: I spent few weeks drinking and partying. In a similar situation in the past, I take a cigarette and used to smoke pretty much non-stop. But this season I was assaulted by the triggers. Smoking in restaurants, communal areas. Many times I thought I can get a cigarette now. But those thoughts were always chased by reason and the power of conviction I have to quit smoking.*

The running example is taken from the Quit Smoking domain. This example is an excerpt of a post shared by a user on Reddit whose main goal is to quit smoking. In this study, we consider each post made by a user as a plan trace. Posts made by all users on this subreddit are aggregated to build the model in latter steps.

### 3.2 Phase-2: Action Extractor

Each post may have more than one sentence, where each sentence may have more than one verb. For each sentence, we extract the verbs and their corresponding nouns using the Stanford part of speech tagger (Toutanova et al. 2003), a state of the art tagger with reported 97.32% accuracy. The extracted verbs are the candidate list of action names. We assume that the order of sentences in a post is indicative of the order of actions we extract from them. In the plan trace, the extracted action names from the first sentence will appear before the extracted action names from the second sentence.

Along with the action names (verbs), we also extract the action parameters (nouns) using the similar strategy and attach the most frequently co-occurring action parameter (noun) with a given action name (verb). For this pipeline, we assume that each action (verb) will have only one action parameter (noun) and two action words can have the same action parameter. For example, assume that there is an action $a_i$ in our dataset which occurs in multiple plan traces and co-occurs with nouns $n_a$, $n_b$ and $n_c$. Noun with the largest co-occurrence frequency with $a_i$ is chosen to be the action parameter for $a_i$. In the examples provided in this paper, action parameters are attached to an action as *<action_name>_<action_parameter_name>* (or sometimes we use *<action_name>* (*<action_parameter_name>*) interchangeably). Since certain English words can be classified as multiple parts of speech tags, we make similar assumptions.

*Running example: [action names]: spent_smoke drink_beer party_hard take_day smoke_day assault_trigger smoke_day thought_smoke chase_life quit_smoke*

From the post made by the user obtained in phase-1, we extract all the verbs and their associated nouns. We assume that the sequentiality among actions is pre-established in the original post made by the user. This assumption sets a constraint that all the verbs extracted are ordered in the same way they occur in the post made by the Reddit user. Since the word 'smoke' can be either a noun or verb, we see the similar pattern in this extracted set of actions and their corresponding parameters.

### 3.3 Phase-3: Generalizer

Since we are handling unrestricted natural language text, it is normal that a same action is used to represent this action's synonyms. Across the aggregated set of posts, there might be verbs that can summarize or subsume a given verb. This motivated us to utilize hierarchical agglomerative clustering approach where low level actions are expressed in high level format. Performing this operation helps reduce the redundancy of actions.

To remove redundant actions, we utilize the agglomerative clustering approach to group semantically similar actions. When clustering the actions, only the action names are considered and their parameters are ignored. This approach (Han, Kamber, and Pei 2011) utilizes Leacock Chodorow similarity metric (*lch* for short)[2] to measure the distance between any two given actions ($W_i$ and $W_j$ – action words). This is one of the popular metrics utilized to compute the semantic similarity between pairs of words. The *l*ch similarity is computed as follows:

$$Sim(W_i, W_j) = Max[log2D - logDist(c_i, c_j)] \qquad (1)$$

where *Dist*($c_i$, $c_j$) is the shortest distance between concepts $c_i$ and $c_j$ (a concept is the general notion or abstract idea) and *D* is the maximum depth of a taxonomy.

We consider a threshold metric (or closeness metric) $\alpha$ to verify the quality and stop the process of agglomeration. The agglomerative clustering algorithm terminates when the closeness metric is greater than the linkage metric at any given point of time. In hierarchical clustering, there are three different types of linkage metrics – *single*, *complete* and *average*. In this paper, we utilize the *complete* linkage metric as the Clustering Quality (we refer to as $cq$) measured is higher ($cq$=8.33) compared to the other linkage metrics (single ($cq$=5.23) and average ($cq$=7.17)). The formal equation to compute the complete linkage metric is $max\{d(a,b) : a \in A, b \in B\}$ where, $d(a,b)$ is the distance metric, *A* and *B* are two separate clusters. When the algorithm terminates, semantically similar actions will be grouped into the same cluster.

Each cluster may have more than a single action that requires us to find a unique cluster representative. To do

[2] http://www.nltk.org/howto/wordnet.html

this, we utilize a popular statistic from Information Retrieval community – *Term Frequency–Inverse Document Frequency* to rank all the actions present in a cluster based on this metric. For each cluster, we choose the top-ranked action with the highest *tfidf* value to be the representative of the respective cluster. The original parameter associated with this action word is continued to be the action parameter after this action word is chosen to be the cluster representative. The statistic can be computed as shown in equation 4 that uses the *TF* and *IDF* equations in 2 and 3 respectively.

$$tf(t,d) = \frac{f_{t,d}}{max f_{t',d} : t' \in d} \tag{2}$$

$$idf(t,D) = log \frac{N}{|d \in D : t \in d|} \tag{3}$$

$$tfidf(t,d,D) = tf(t,d) * idf(t,D) \tag{4}$$

where $t$ is the given action; $d$ is the set of raw posts shared by a given user; $D$ is the super set of all sets of raw posts made by all the users in our raw dataset (where $|D|$ will be equal to the number of unique users in our dataset). Each cluster will be represented by a unique top-ranked action word.

*Running example: [clustering]: We map the action names to the cluster representatives of their corresponding cluster. spent $\rightarrow$ spend, drink $\rightarrow$ party, take $\rightarrow$ taken*

### 3.4 Phase-4: Trace Builder

The initial plan fragments are converted into plan traces by replacing the action names with their corresponding cluster representatives[3]. This process is repeated on all the posts to build the traces.

*Running example: [rebuilding plan traces]: initial plan fragment: [spent_smoke, drink_beer, party_hard, take_day, smoke_day, assault_trigger, smoke_day, thought_smoke, chase_life, quit_smoke], plan trace: [spend_time, party_hard, taken_hold, smoke_day, assault_trigger, smoke_day, thought_smoke, chase_life, quit_smoke]*

The actions in the running example such as *spent*, *drink*, *take* are represented in their corresponding high level mapped format. Since *drink* is mapped to *party*, *drink_beer* is represented as *party_hard*. In the plan fragment, [*spend_time*, *party_hard*, *party_hard*, *taken_hold*, . . . ], two '*party*' actions are occurring sequentially. Hence, we remove repetitions and include only one such instance. However, in general a plan may include repeated actions and we acknowledge that our system may miss out on those plans with repeated actions.

### 3.5 Phase-5: Sequence Probability Learner

After extracting the plan traces, we then extract the pre-actions and post-actions for any given action. Due to co-occurrence in the plan traces, actions are inter-related to other actions with a probability ($p(a_i, a_j)$) describing the chance of action $a_j$ following action $a_i$. This probability is computed purely in a data-driven fashion. This procedure considers a constraint metric $\theta$ that decides whether a co-occurring relation should be included in the model. We compute the conditional probability $p(a_i \mid a_j)$ using the following equation:

$$p(a_i \mid a_j) = \frac{p(a_i \cap^* a_j)}{p(a_j)} \tag{5}$$

Let $a_1$ and $a_2$ are two actions where $a_1$ is an effect of $a_2$ which indicates that unless $a_2$ is executed, $a_1$ cannot be executed. The support-based probability then is computed where $a_1$ will be the postcondition of $a_2$ if $p(a_1 \mid a_2) > \theta$ and $a_2$ will be the precondition of $a_1$. The $\cap^*$ in Equation 5 represents an ordered conjunction that considers the sequentiality of $a_1$ and $a_2$ while computing the frequency of their occurrence together.

Once we extract the pre-actions and post-actions for every action in the data set, we represent the relationships in the form of an Action Graph (*AG*) as our incomplete action model (*M*) to generate shallow workflows. *AG* consists of actions as *nodes* and each *edge* is a transition between two actions $a_i$ and $a_j$. The edge weight between any two nodes $a_i$ and $a_j$ is the support-based probability $p(a_i, a_j)$.

*Running example: [data-driven probabilities]: In the plan trace dataset, we compute the sequential probabilities for any given action pairs. This results in generating a precedence graph shown in Figure 2 for the "Quit Smoking" domain, where the sink node is the action quit with no subsequent effect.*

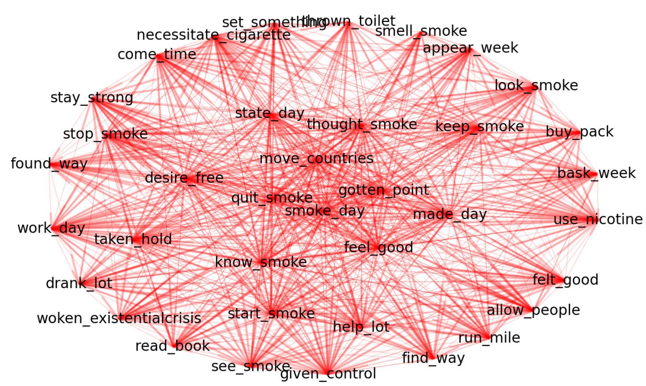


Figure 2: Part of the directed graph $G$ for the $QuitSmoking$ domain with *quit_smoke* as the sink node.

### 3.6 Phase-6: Model Validator

We divide the set of plan traces $D$ into training data, $D_{tr}$, and testing data, $D_{te}$. By this step, we have the set of plan traces represented in the lifted action format. This common lifted representation ensures that a given action uses the same name in both $D_{tr}$ and $D_{te}$. We use $D_{tr}$ to build the model *M*. Let *T* be the set of transitions present in *M* and

[3] Note, it is possible to replace two sequential action names by the same cluster representative. In that case we remove the repeated action name; hence, reducing the length of the plan trace. For example, if the representation of a plan trace is: $[a_1, a_2, a_2, a_5, a_8]$, after post-processing it will be converted to $[a_1, a_2, a_5, a_8]$.

$T^{'}$ be the set of transitions in test dataset. Since *M* is used to generate workflows, the goodness of this model should be measured to trust the quality of these plans. To determine goodness of *M*, we define a new metric called *explainability* that can be computed as shown in Equation 6.

$$T^{''} = T \cap T^{'}$$
$$Explainability = \frac{|T^{''}|}{|T^{'}|} \tag{6}$$

# 4 Evaluation Methodology

We evaluate the pipeline from two perspectives: 1) data and approach employed to construct the incomplete action model in terms of explainability 2) workflows generated by the incomplete action model in terms of soundness and completeness. We evaluate the data utilized by the pipeline followed by the extracted plans represented as shallow workflows. Although we have the incomplete action model in the *pre-action* $\longrightarrow$ *action* $\longrightarrow$ *post-action* format (a sample of these models extracted for the three domains is show in Table 3), we are still in the process of attempting to convert and refine this incomplete model to a PDDL-style model. This attempt could be a valuable contribution to the automated planning community (Srivastava and Kambhampati 2005). Planning community can no longer depend on a fixed set of domains for the International Planning Competition (IPC) challenges but instead expand the domains to any real-world scenarios.

| **Quit Smoking** |
|---|
| (:action change(ability) |
| [:pre-action eat(gross) crave(succeed) dealt(reality)] |
| [:post-action set(goal) run(mile) quit(smoke)] ) |
| Possible explanation: Someone is craving for success and is dealing with the reality of eating gross food who wants to change his abilities that led that person to set some goals, run miles and quit smoking. |
| **Couch to 5K** |
| (:action sign(race) |
| [:pre-action recommend(c25k) push(run) refer(program) ] |
| [:post-action begin(week) run(minute) cover(mile) know(battle) kept(pace) ] ) |
| Possible explanation: A person was recommended the couch to 5K reddit forum and was being pushed to run. So, he refers to a program and signs up for the race. After this, he begins from the next week to run few minutes and cover few miles. The person knows the battle but he kept the pace. |
| **Wedding Planning** |
| (:action hate(dress) |
| [:pre-action pick(dress) saw(dress) blame(problem) cost(much) prove(difficult)] |
| [:post-action kill(wed) find(dress) move(wed)]) |
| Possible explanation: The person sees and picks her dress. It may cost a lot but starts blaming someone for the problem and now hates the dress. The next steps could be to kill the wedding at the moment, find a new dress and move the wedding date. |

Table 3: Sample actions from the incomplete models extracted for the 3 domains **automatically** by this pipeline and their possible explanations provided by the human subjects.

## 4.1 Evaluation-1 – Explainability

Prior to analyzing the pipeline, it is important to examine whether the data we are utilizing to construct the incomplete action models is consistent across all the experiential posts shared online by the users. To evaluate this, we measure the explainability of the incomplete action model by varying the $\alpha$ value (clustering threshold). $\alpha$ decides on the amount of redundancy to be removed from the posts. The smaller the value of $\alpha$, the larger the redundancy present in the data considered. We fix the size of the training data ($D_{tr}$) to 80% of the entire set of plan traces and the remaining as the test data set ($D_{te}$) and conduct experiments on all the three domains separately. The dataset from each domain consists of a set of plans that are aimed at achieving the primary goal of the corresponding domain. The pipeline first utilizes $D_{tr}$ to build the incomplete action model *M* and then use the test dataset $D_{te}$ to evaluate the explainability of *M*.

| $\alpha$ | **Quit Smoking** | **Couch to 5K** | **Wed. Planning** |
|---|---|---|---|
| 2.50 | 65.66% | 64.5% | 73.39% |
| 2.25 | 65.66% | 64.59% | 73.39% |
| 2.0 | 68.41% | 69.78% | 77.7% |
| 1.75 | 69.33% | 70.67% | 78.39% |
| 1.50 | 80.58% | 82.06% | 84.68% |
| 1.25 | 90.42% | 89.43% | 91.6% |
| 1.0 | 89.31% | 89.91% | 91.04% |

Table 4: Average explainability measured by Eq. 6 as we vary $\alpha$ through 10-fold cross-validation

As shown in Table 4, the maximum explainability value was reached at $\alpha$ = 1.25. It is expected that if the data and the approach are correct, the explainability value should be directly proportional to the value of $\alpha$. This trend is clearly visible in the results shown in Table 4. This trend also positions more confidence in building the best incomplete model used to generate shallow workflows. Also, we focus on how well can these incomplete domain models explain the newly seen data to decide the consistency of goal-oriented experiences shared by users. The results obtained through 10-fold cross-validation show that *M* has the potential to obtain 90% accuracy. The results display the strength of unstructured data from social media platforms like Reddit could be employed to build incomplete models.

## 4.2 Evaluation-2 – Soundness & Completeness

Next, we examine the "goodness" of the incomplete action models by evaluating the generated shallow workflows. Each workflow is generated by representing the incomplete model as a graph and is the shortest path in this graph from a given source node to the goal node. For example, in *Quit Smoking* domain, the source node can be *start(smoke)* and the goal node is *quit(smoke)*. To identify the best path, we utilized the weight-based Djikstra's shortest path algorithm from the *NetworkX* (`https://networkx.github.io/`) Python library. We rate each plan on a binary-scale evaluating it's soundness and completeness metrics.

| Domain | Soundness | Completeness |
|---|---|---|
| **Quit Smoking** | 42% | 38% |
| **Couch to 5K** | 66% | 45% |
| **Wedding Planning** | 36% | 43% |

Table 5: Soundness and Completeness as evaluated by the human subjects. Note that higher the percentage values, the better the workflows that are generated.

**Soundness:** is defined as whether a given shallow workflow is meaningful and can help achieve the goal.

**Completeness:** is defined as if a given shallow workflow is missing any important actions to achieve the goal.

We recruited 10 human test subjects who evaluated the top-5 workflows generated by $M$. We provide instructions to the test subjects and ask them to rate the soundness and completeness of each workflow. Each subject evaluates all the top-5 workflows from the three domains and the combined statistics are shown in Table 5. Each percentage value in this table is the average value of all the votes gathered by the plans in a given domain.

The best plan among these 15 plans (combined all top-5 plans from the 3 domains considered) is from the *Couch to 5K* domain – *inhale(nose)* $\rightarrow$ *exhale(mouth)* $\rightarrow$ *aid(loss)* $\rightarrow$ *transform(life)* $\rightarrow$ *outpaced(brain)* $\rightarrow$ *slow(pace)* $\rightarrow$ *run(minutes)*. This shallow workflow was described by the human subjects as “If you inhale through nose and exhale from mouth (a powerful breathing pattern[4]) that will help you relax and transforms by keeping your slow pace to run the 5K in minutes.” Notice that these workflows are not partially meaningful. However, the evaluation results showed that they make sense to humans as shown by the results presented in Table 5. The table shows that the *Couch to 5K* domain has highest soundness and completeness values which might be due to the fact that the number of original number of actions are relatively lower that led to a model with less redundancy. Another reason could be the workflows generated from this domain are more meaningful to the human test subjects. With regard to completeness, test subjects expressed the difficulty of not being completely aware of the domains and so by default assumed that there should be a missing action in the plan.

## 5 Related Work

The work reported in this paper brings together work from three communities that are quite far apart – traditional (classical) planning, social computing and Natural Language Processing (NLP).

**Automated Planning & AI:** Classical planning techniques have sought to mostly ignore domain acquisition and maintenance issues in favor of search efficiency and plan synthesis. Towards this goal, multiple works have focused on learning the action models through inductive logic programming, from sets of successful plan traces (Yang, Wu, and Jiang 2007; Zhuo et al. 2010), improving partial models (Oates and Cohen 1996; Gil 1994), etc. Recent work on model-lite planning (Kambhampati 2007; Yoon 2007; Zhuo, Nguyen, and Kambhampati 2013) acknowledges that learned models may be forever plagued by incompleteness and laden with uncertainty, and plan synthesis techniques themselves may have to change in order to accommodate this reality. Other existing works (Addis and Borrajo 2011; Lindsay et al. 2017; Tenorth, Nyga, and Beetz 2010; Waibel et al. 2011) that includes literature from the field of robotics attempts at learning action models from the web where they consider plans recommended by websites like wikihow.com for a given task. This work focuses on carefully constructing well-curated complete domain models where as the work proposed in this paper emphasizes on building incomplete models that are efficient enough to perform automated planning tasks that include plan recognition and obtaining meaningful plans.

**Social Computing:** On the other hand, work in the social media field puts a premium on the analysis of data, but very little on complex decision-making that can build on the knowledge embedded within that data (Kiciman and Richardson 2015). Given the high level human engagement with these platforms, researchers have sought to utilize the data generated on them for various analyses that can help understand and predict users’ behaviors (Golder and Macy 2011; Sarker et al. 2015; Paul and Dredze 2011). Building on this theme of using human-generated data on social media, the crowd sourcing community realized that in addition to using the inadvertent by-product of user participation on social media, it could also directly utilize the “crowd” to prepare plans for goal-oriented tasks (Law and Zhang 2011; Manikonda et al. 2014). This work has gained traction in recent years in part as a response to the unavailability of good planning models for many real-world, everyday planning and scheduling tasks. Such “hybrid” intelligence systems utilize domain knowledge that is split between humans and machines, with each party possessing complementary information; unfortunately, these systems are still far from being scalable and cost-effective.

**NLP:** There is another set of work (Harabagiu and Maiorano 2002; Collier 1998) from the natural language processing (NLP) community which focus on extracting domain templates. The templates extracted from this literature capture most important information of a particular domain and they can be used across multiple instances of that domain especially in the field of information extraction. For example, the GISTexter summarization system considers semantic relations from WordNet along with summary statistics over an arbitrary document collection. This type of summarization could be at a disadvantage if there is only one instance of the domain as input as addressed by Filatova et. al 2006. This direction of work is later extended to identify event schema using count-bases statistics and by building formal generative models (Chambers and Jurafsky 2008; Chambers 2013). The main distinction of this line of research from our work is two fold: (1) the model: our main aim is to build shallow models where as, the existing literature aims at constructing full models; (2) the unstructured natural language data on online social media platforms: our proposed pipeline handles natural language with different

[4]https://goo.gl/BiKvGG

styles of language, where as the existing literature considers fairly structured text available online.

## 6 Conclusions and Future Work

To support sequential decision making, action models extracted from the unstructured data are very valuable. However, extracting these models from unstructured data is difficult. Towards exploring these challenges and to measure the feasibility of building usable action models, this paper proposes a novel six-phase pipeline. This pipeline takes as input the unstructured web data and automatically generates the incomplete action model. Through evaluations, we show the capability of utilizing shallow NLP techniques to overcome the challenges posed by various entities and successfully generate incomplete action models. We acknowledge that the workflows generated by the incomplete action models are shallow. However, the evaluations displayed the power of experiential statuses shared by the users on online social media platforms can be used to generate incomplete action model. Also, the capability of these models to generate plans as workflows are tagged by human subjects as sound to a certain extent.

As a future work, these incomplete action models can be translated to PDDL-style models. In addition, hierarchical representation of actions can be extracted in order to enhance the extracted models. We hope that this work inspires the research community to utilize the potential of incomplete action models to perform automated planning tasks. Also, considering the wealth of information present on online social media platforms especially the goal-oriented posts shared publicly, we envision that further action models are constructed towards supporting sequential decision making.

## References


Addis, A., and Borrajo, D. 2011. From unstructured web knowledge to plan descriptions. *Information Retrieval and Mining in Distributed Environments* 41–59.

Chambers, N., and Jurafsky, D. 2008. Unsupervised learning of narrative event chains. In *Proc. ACL.*

Chambers, N. 2013. Event schema induction with a probabilistic entity-driven model. In *Proc. EMNLP*.

Collier, R. 1998. Automatic template creation for information extraction. In *Ph.D. thesis, Univerisyt of Sheffield*.

Filatova, E.; Hatzivassiloglou, V.; and Mckeown, K. 2006. Automatic creation of domain templates. In *Proc. COLING/ACL*.

Gil, Y. 1994. Learning by experimentation: Incremental refinement of incomplete planning domains. In *Proc. ICML*.

Golder, S. A., and Macy, M. W. 2011. Diurnal and seasonal mood vary with work, sleep, and daylength across diverse cultures. *Science* 333(6051).

Gregory, P., and Lindsay, A. 2016. Domain model acquisition in domains with action costs. In *Proc. ICAPS*.

Han, J.; Kamber, M.; and Pei, J. 2011. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., 3rd edition.

Harabagiu, A. M., and Maiorano, S. J. 2002. Multi-document summarization with gistexter. In *Proc. LREC*.

Kambhampati, S. 2007. Model-lite planning for the web age masses: The challenges of planning with incomplete and evolving domain models. In *Proc. AAAI*.

Kiciman, E., and Richardson, M. 2015. Towards decision support and goal achievement: Identifying action-outcome relationships from social media. In *Proc. KDD*.

Law, E., and Zhang, H. 2011. Towards large-scale collaborative planning: Answering high-level search queries using human computation. In *Proc. AAAI*.

Lindsay, A. A.; Read, J. J.; Ferreira, J. F. J.; Hayton, T. T.; Porteous, J. J.; and Gregory, P. J. P. 2017. Framer: Planning models from natural language action descriptions. In *Proc. ICAPS*.

Manikonda, L.; Chakraborti, T.; De, S.; Talamadupula, K.; and Kambhampati, S. 2014. AI-MIX: Using automated planning to steer human workers towards better crowd-sourced plans. In *Proc. IAAI*.

Oates, T., and Cohen, P. R. 1996. Searching for planning operators with context-dependent and probabilistic effects. In *Proc. AAAI*.

Paul, M., and Dredze, M. 2011. You are what you tweet: Analyzing twitter for public health. In *Proc. ICWSM*.

Sarker, A.; Ginn, R.; Nikfarjam, A.; OConnor, K.; Smith, K.; Jayaraman, S.; Upadhaya, T.; and Gonzalez, G. 2015. Utilizing social media data for pharmacovigilance: A review. *Journal of Biomedical Informatics* 54.

Srivastava, B., and Kambhampati, S. 2005. The case for automated planning in autonomic computing. In *Proc. ICAC*.

Tenorth, M.; Nyga, D.; and Beetz, M. 2010. Understanding and executing instructions for everyday manipulation tasks from the world wide web. In *Proc. ICRA*.

Tian, X.; Zhuo, H. H.; and Kambhampati, S. 2016. Discovering underlying plans based on distributed representations of actions. In *Proc. AAAMAS*.

Toutanova, K.; Klein, D.; Manning, C. D.; and Singer, Y. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proc. NAACL*.

Waibel, M.; Beetz, M.; Civera, J.; dAndrea, R.; Elfring, J.; Galvez-Lopez, D.; Haussermann, K.; Janssen, R.; Montiel, J.; Perzylo, A.; et al. 2011. A world wide web for robots. *IEEE Robotics & Automation Magazine* 18(2):69–82.

Yang, Q.; Wu, K.; and Jiang, Y. 2007. Learning action models from plan examples using weighted max-sat. *Artif. Intell.* 171(2-3).

Yoon, S. 2007. Towards model-lite planning: A proposal for learning and planning with incomplete domain models. In *Proc. Workshop on AI Planning and Learning, ICAPS*.

Zhuo, H. H.; Yang, Q.; Hu, D. H.; and Li, L. 2010. Learning complex action models with quantifiers and logical implications. *Artif. Intell.* 174(18).

Zhuo, H. H.; Nguyen, T.; and Kambhampati, S. 2013. Refining incomplete planning domain models through plan traces. In *Proc. IJCAI*.