A Dataset for Interactive Vision-Language Navigation with Unknown Command Feasibility Supplementary

Andrea Burns[1], Deniz Arsan[2], Sanjna Agrawal[1], Ranjitha Kumar[2], Kate Saenko[1,3], and Bryan A. Plummer[1]

[1] Boston University, Boston MA 02215, USA
{aburns4,sanjna,saenko,bplum}@bu.edu
[2] University of Illinois Urbana-Champaign, Champaign IL 61820, USA
{darsan2,ranjitha}@illinois.edu
[3] MIT-IBM Watson AI Lab, Cambridge MA 02142, USA

Table of Contents

1 MoTIF Collection

For data collection, we use UpWork[4] as our crowd sourcing platform and hired 34 people to collect our dataset. Of the annotators, 21 identified as female and 13 identified as male. The median age of the annotators was 23.5 years old. Annotators were from 18 different states in the U.S. and had a range of education from a high school diploma to a master's degree (2 have high school degrees, 24 have bachelor's degrees, and 8 have master's degrees).

Annotators were selected on UpWork if their profile skills listed data entry. As the initial iteration of MoTIF is in English, we also required annotators be fluent in English, but did not require them to be native speakers. We posted separate job listings for the task writing (base rate $15/hr) and task demonstration (base rate $10/hr) portions of the data collection, having independent annotators for the two stages. Annotators hired for the task writing portion were not informed of our interest in potentially ambiguous or infeasible tasks.

For the annotators hired for task demonstration, we additionally required them to have personal experience with Android devices so that there was no additional noise introduced from people unfamiliar with Android apps. We created anonymized login information for annotators so that no personally identifiable information was collected. Additional interface details and an example of the interface used by the workers (Figure 1) is provided in Section 1.1.

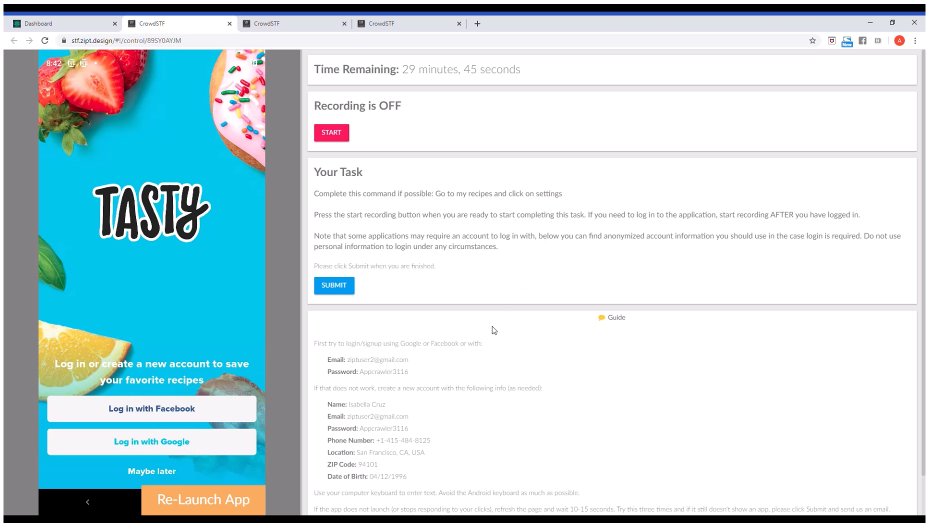
1.1 Data Collection Interface

We provide an example of what our data collection interface looks like for annotators while they explore an Android app and perform a task demonstration in Figure 1. Annotators are given the natural language task to attempt within the Android app in the 'Your Task' section on the right side of the interface. Below, we provide anonymized email login and password credentials for them to use if needed. The left hand side of the collection interface displays the phone screen from a physical Android device which is remotely connected to our collection website, from which we record all actions taken on the phone and the app modalities as described in the main text.

1.2 Application List

We include lists of all Android apps we collect demonstrations for in Tables 1-3. In addition to listing the app package name, we provide the corresponding Google Play Store Category and how that particular app's tasks were paired (app-specific, paired, or category-clustered). The apps selected for MoTIF were across fifteen app categories: lifestyle, communication, dating, food and drink, maps and navigation, news and magazines, productivity, shopping, social, travel, weather, tools, music and audio, entertainment, and education. For privacy, we do not intend to collect any demonstrations of natural language commands within dating

[4] https://www.upwork.com/



Time Remaining: 29 minutes, 45 seconds

Recording is OFF

Your Task

Complete this command if possible: Go to my recipes and click on settings

Press the start recording button when you are ready to start completing this task. If you need to log in to the application, start recording AFTER you have logged in.

Note that some applications may require an account to log in with, below you can find anonymized account information you should use in the case login is required. Do not use personal information to login under any circumstances.

Please click Submit when you are finished.

First try to login/signup using Google or Facebook or with:

Email: ziptuser2@gmail.com
Password: Appcrawler3116

If that does not work, create a new account with the following info (as needed):

Name: Isabella Cruz
Email: ziptuser2@gmail.com
Password: Appcrawler3116
Phone Number: +1-415-484-8125
Location: San Francisco, CA, USA
ZIP Code: 94101
Date of Birth: 04/12/1996

Use your computer keyboard to enter text. Avoid the Android keyboard as much as possible.

If the app does not launch (or stops responding to your clicks), refresh the page and wait 10-15 seconds. Try this three times and if it still doesn't show an app, please click Submit and send us an email.

Fig. 1: The website interface annotators use to interact with an Android app and record their task demonstration. We provide anonymized information if needed for logging in or for forms at any point so that no personal identifying information is collected

apps, and will not be releasing any of the raw data collected when annotators decided on a list of natural language tasks for dating apps in the first stage of collection. We simply include dating apps as one Android category to see what kinds of tasks people would consider being automated in this setting. We will share the resulting natural language tasks, but no captured screen or view hierarchy data. The dating apps included com.wildec.dating.meet4u, com.once.android, emotion.onekm, ru.fotostrana.sweetmeet, com.mason.wooplus, and com.hitwe.android.

1.3 Dataset Examples

We include more example (app, task) pairs and their resulting action sequences from MoTIF. Figure 2 and 3 show samples for infeasible and feasible commands, respectively.

Table 1: A list of applications used in MoTIF, their Google Play Store Category, and how their submitted natural language tasks were grouped with applications in the (app, task) pairing stage. N/A refers to apps which has technical difficulties during the demonstration stage and we are working to resolve

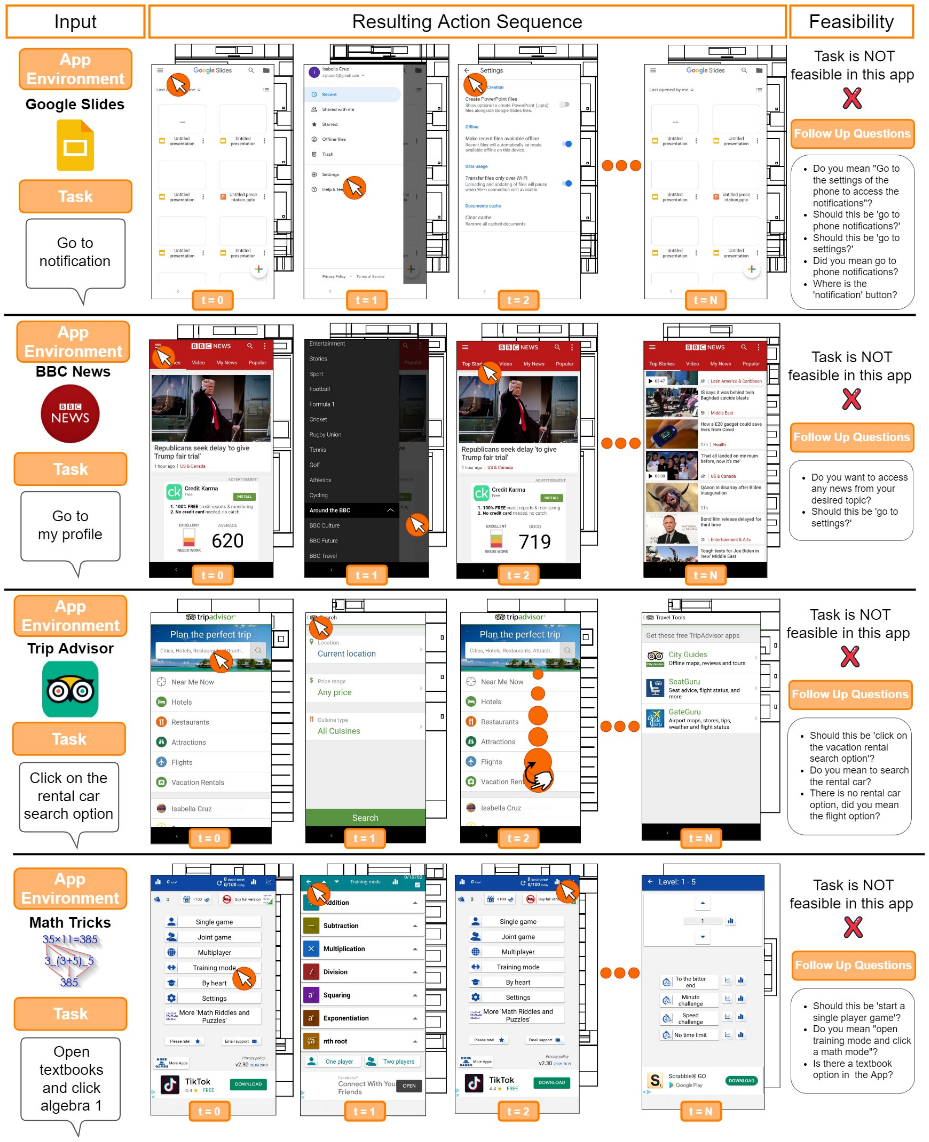
Google Play Store Category	App Name	(app, task) Pairing Method
Education	com.ted.android	*app-specific*
	gov.nasa	*app-specific*
	example.matharithmetics	*paired*
	org.khanacademy.android	*app-specific*
	com.duolingo	*app-specific*
	com.quizlet.quizletandroid	*app-specific*
	com.remind101	N/A
	org.coursera.android	N/A
	com.microblink.photomath	*paired*
Entertainment	com.megogo.application	*app-specific*
	com.app.emotes.dances.fortnite	*app-specific*
	com.scannerradio	*app-specific*
	com.google.android.youtube	*app-specific*
	com.zombodroid.MemeGenerator	*app-specific*
	tv.pluto.android	*app-specific*
	com.tubitv	*app-specific*
	com.imdb.mobile	*app-specific*
	com.eventbrite.attendee	*app-specific*
Communication	com.google.android.gm	*app-specific*
	com.sec.android.app.sbrowser	*paired*
	com.facebook.orca	N/A
	com.whatsapp	N/A
	org.mozilla.firefox	*paired*
	com.skype.raider	N/A
Food & Drinks	com.joelapenna.foursquared	*app-specific*
	com.yum.pizzahut	*app-specific*
	com.chickfila.cfaflagship	*app-specific*
	com.dominospizza	*paired*
	in.swiggy.android	*app-specific*
	com.opentable	*app-specific*
	com.starbucks.mobilecard	*app-specific*
	vivino.web.app	*app-specific*
Lifestyle	com.hm.goe	*app-specific*
	com.adpog.diary	*app-specific*
	com.aboutjsp.thedaybefore	*app-specific*
	info.androidz.horoscope	N/A
	ru.mail.horo.android	*paired*
	com.urbandroid.sleep	*app-specific*
	com.hundred.qibla	*app-specific*

Table 2: A list of applications used in MoTIF, their Google Play Store Category, and how their submitted natural language tasks were grouped with applications in the (app, task) pairing stage. N/A refers to apps which has technical difficulties during the demonstration stage and we are working to resolve

Google Play Store Category	App Name	(app, task) Pairing Method
Maps & Navigation	com.tranzmate	*category-clustered*
	com.mapfactor.navigator	*category-clustered*
	com.thetrainline	*category-clustered*
	com.citymapper.app.release	*app-specific*
	com.prime.studio.apps.route.finder.map	*category-clustered*
	com.waze	*category-clustered*
	com.nyctrans.it	*category-clustered*
Music & Audio	com.radio.fmradio	*app-specific*
	deezer.android.app	*app-specific*
	com.spotify.music	*category-clustered*
	com.pandora.android	*category-clustered*
	com.springwalk.mediaconverter	*category-clustered*
	com.google.android.music	*category-clustered*
	com.clearchannel.iheartradio.controller	*category-clustered*
	com.melodis.midimiMusicIdentifier.freemium	*category-clustered*
News & Magazines	fm.castbox.audiobook.radio.podcast	*category-clustered*
	com.ss.android.article.master	N/A
	com.opera.app.news	*category-clustered*
	bbc.mobile.news.ww	*category-clustered*
	com.quora.android	N/A
	com.google.android.apps.magazines	*category-clustered*
	com.reddit.frontpage	*app-specific*
	com.sony.nfx.app.sfrc	*category-clustered*
Shopping	com.amazon.mShop.android.shopping	*app-specific*
	com.abtnprojects.ambatana	*category-clustered*
	com.contextlogic.wish	*category-clustered*
	com.joom	*category-clustered*
	com.ebay.mobile	*category-clustered*
	com.walmart.android	*category-clustered*
	club.fromfactory	*app-specific*
	com.zzkko	*app-specific*
	com.groupon	*category-clustered*
Productivity	cn.wps.moffice_eng	*category-clustered*
	com.google.android.apps.docs.editors.sheets	*category-clustered*
	com.google.android.apps.docs	N/A
	com.microsoft.office.outlook	*category-clustered*
	com.google.android.calendar	*category-clustered*
	com.google.android.apps.docs.editors.slides	*category-clustered*
	com.dropbox.android	N/A

Table 3: A list of applications used in MoTIF, their Google Play Store Category, and how their submitted natural language tasks were grouped with applications in the (app, task) pairing stage. N/A refers to apps which has technical difficulties during the demonstration stage and we are working to resolve

Google Play Store Category	App Name	(app, task) Pairing Method
Tools	com.lenovo.anyshare.gps	*app-specific*
	com.antivirus	*paired*
	com.google.android.calculator	*paired*
	com.miui.calculator	*paired*
	com.google.android.apps.translate	*app-specific*
	com.avast.android.mobilesecurity	*paired*
Travel	com.kayak.android	*paired*
	com.tripadvisor.tripadvisor	*paired*
	com.trivago	*paired*
	com.google.android.apps.maps	*paired*
	com.yelp.android	*app-specific*
	com.booking	N/A
	com.google.earth	*paired*
	com.mapswithme.maps.pro	*app-specific*
	com.google.android.street	*paired*
	com.yellowpages.android.ypmobile	*app-specific*
Weather	com.gau.go.launcherex.gowidget.weatherwidget	N/A
	com.devexpert.weather	*category-clustered*
	com.chanel.weather.forecast.accu	*category-clustered*
	com.weather.Weather	*category-clustered*
	com.droid27.transparentclockweather	*app-specific*
	aplicacion.tiempo	*category-clustered*
	com.accuweather.android	*category-clustered*
	com.windyty.android	*category-clustered*
	com.handmark.expressweather	*category-clustered*
Social	com.zhiliaoapp.musically	*category-clustered*
	com.pinterest	*category-clustered*
	com.instagram.android	*category-clustered*
	com.facebook.katana	*category-clustered*
	com.sgiggle.production	*app-specific*
	com.snapchat.android	*app-specific*
	com.ss.android.ugc.boom	*category-clustered*
	com.lazygeniouz.saveit	*category-clustered*



- Do you mean "Go to the settings of the phone to access the notifications"?
- Should this be 'go to phone notifications?'
- Should this be 'go to settings?'
- Did you mean go to phone notifications?
- Where is the 'notification' button?

- Do you want to access any news from your desired topic?
- Should this be 'go to settings?'

- Should this be 'click on the vacation rental search option'?
- Do you mean to search the rental car?
- There is no rental car option, did you mean the flight option?

- Should this be 'start a single player game'?
- Do you mean "open training mode and click a math mode"?
- Is there a textbook option in the App?

Fig. 2: Example tasks from MoTIF deemed infeasible by annotators. We show the input (app, task) pair for task demonstration, the resulting task demo (which captures the rendered screen, app view hierarchy, and action localization), and the feasibility annotations and follow up questions posed by annotators

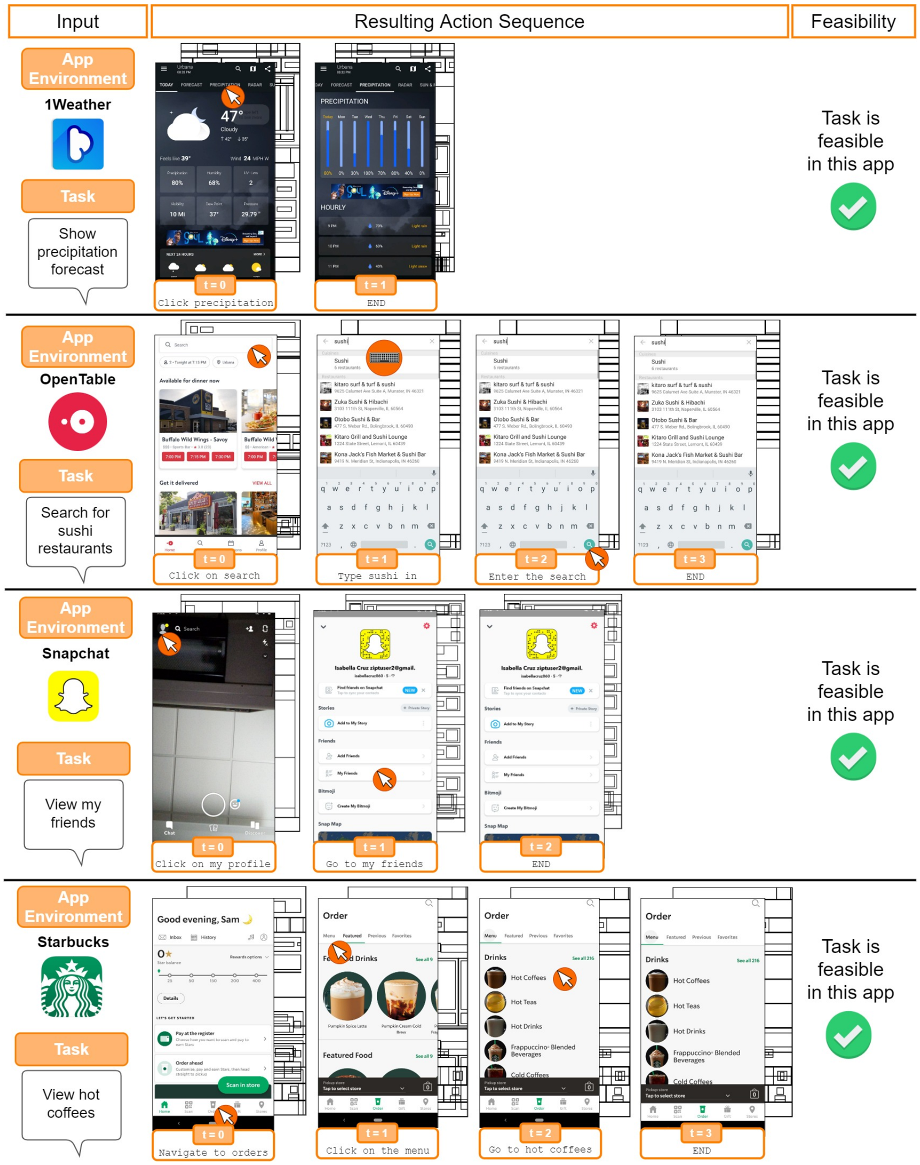
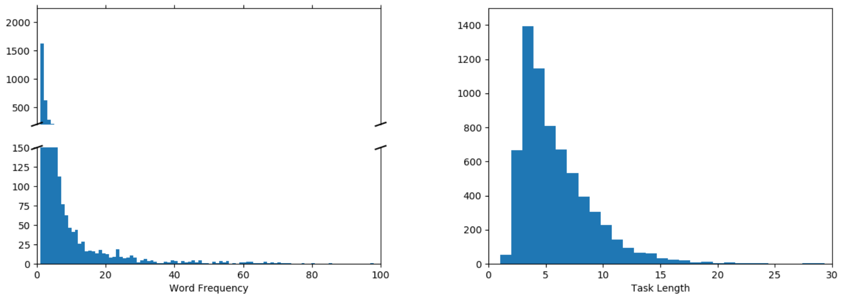


Fig. 3: Example tasks from MoTIF deemed feasible by annotators. We show the input (app, task) pair for task demonstration, the resulting task demo (which captures the rendered screen, app view hierarchy, and action localization), and the feasibility annotations and follow up questions posed by annotators



(a) The word frequency distribution of MoTIF's vocabulary

(b) The length (number of words per task) distribution of MoTIF's tasks

Fig. 4: Additional statistics on MoTIF's language tasks

2 MoTIF Statistics

We include statistics over the high-level goals collected for MoTIF in Section 2.1 and word cloud visualizations over all commands and per category in Section 2.2. We discuss annotator agreement when determining command feasibility in Section 2.3. Lastly, the cluster visualizations used to define (app, task) pairs in MoTIF are illustrated in Section 2.4.

2.1 Natural Language Command Statistics

We provide additional statistics on the natural language high-level goals in MoTIF in Figure 4. In Figure 4a we plot a histogram over the word frequency of the command vocabulary and Figure 4b shows a histogram over the task length (*i.e.*, how many words a task consists of) across all collected natural language tasks. Both reflect a long tail distribution, which is common for word frequency, and follows Zipf's Law. For task length, the distribution is skewed towards shorter length tasks (nearly all collected tasks have fewer than ten words), which aligns with MoTIF's natural language commands mostly capturing high-level goals.

2.2 Word Cloud Visualizations

We include a word cloud illustration over all high-level commands in MoTIF in Figure 5. The larger the word in the word cloud, the more often it occurs in MoTIF's collected tasks. As we compute the word cloud over all tasks (which span fifteen different Google Play Store app categories) we can see the largest words are those that are action or instruction oriented words, like 'click,' 'search,' or 'show.' In Figure 6, we show word clouds for tasks per app category.

While there are some common words with high frequency across all app categories (like the action oriented words largest in Figure 5), there are other

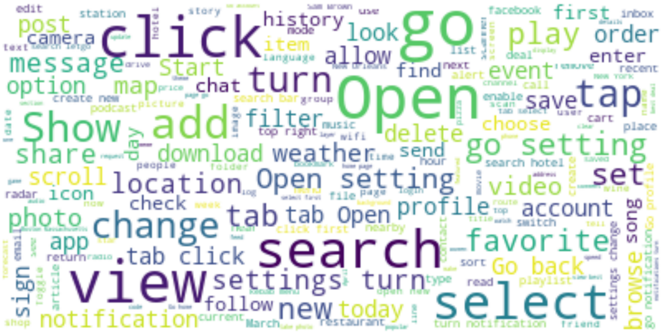


Fig. 5: Word cloud visualization over all MoTIF high-level language commands. The larger the word is illustrated, the more often it occurs

words illustrated that reflect each app category and functionality specific to that topic. For example, in the Education word cloud in the top left of Figure 6, we see words ‘lesson,’ ‘math,’ and ‘history.’ In contrast, the Shopping category in Figure 6 shows words like ‘deal,’ ‘search,’ and ‘cart’ with high frequency.

The word cloud visualizations also show the density of words for each Android app category’s collected tasks. The Food & Drink, Productivity, and Music & Audio app categories have the smallest vocabularies, with less densely populated word clouds. This reflects there being lower diversity in the kinds of requests asked by people for these app categories. On the other hand, Maps & Navigation, Weather, and Travel are examples of Android app categories with larger task vocabularies. This can reflect greater diversity in app requests collected, which may be due to the diversity of functionality in these app categories, or the fact that these apps can have highly specific, *i.e.*, very fine-grained, requests (like searching for one location’s weather out of the nearly unlimited locations one could request).

2.3 Annotator Feasibility Agreement

We define annotator feasibility labeling agreement as the fraction of the number of votes for the majority voted label ($max(C_{yes}, C_{no})$) over all votes ($C_{yes}+C_{no}$) for an (app, task) pair in MoTIF, where C_{yes} is the count of votes for feasible and C_{no} is the count of votes for infeasible. In Figure 7, we bin different degrees of annotator agreement and plot each bin’s counts over all (app, task) pairs with demonstrations in MoTIF. The minimum agreement is 50% and maximum agreement is 100%. The majority of our (app, task) pairs have annotation agreement between 90-100%, with 296 (app, task) pairs falling in this maximal bin.

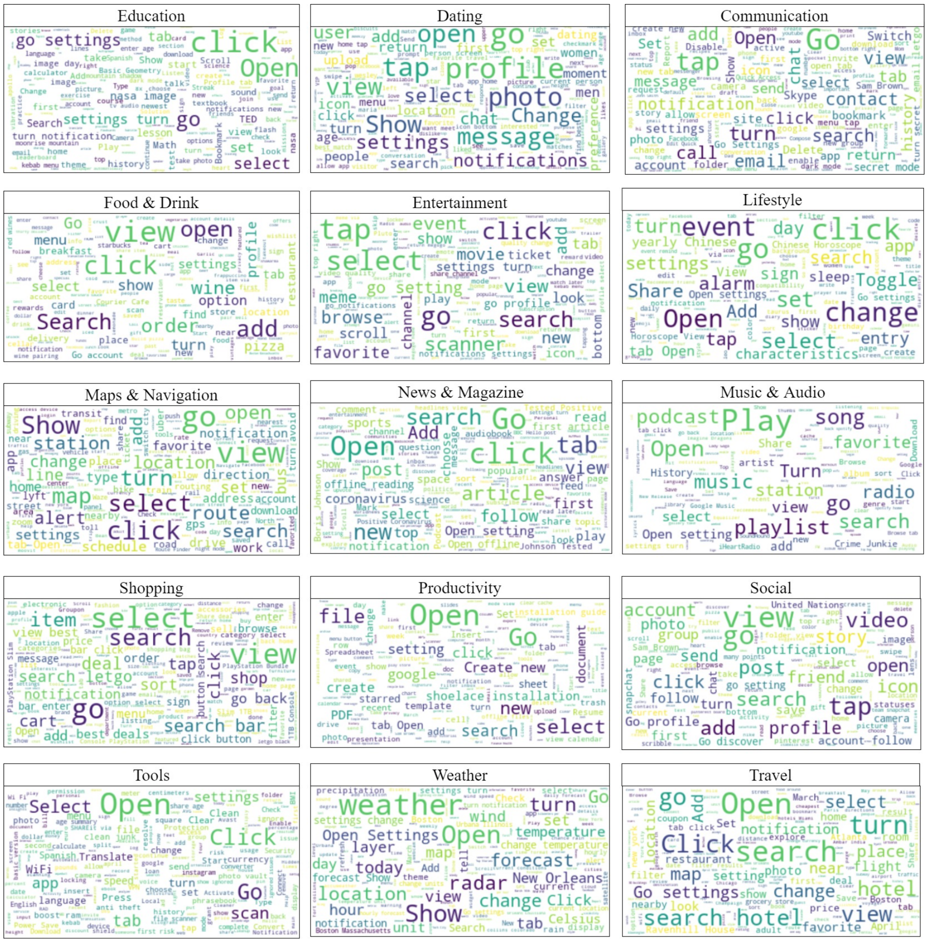


Fig. 6: Word cloud visualization of MoTIF high-level language tasks per Android app category. There are fifteen total categories: Education, Dating, Communication, Food & Drink, Entertainment, Lifestyle, Maps & Navigation, News & Magazine, Music & Audio, Shopping, Productivity, Social, Tools, Weather, and Travel. The larger the word is illustrated, the more often it occurs

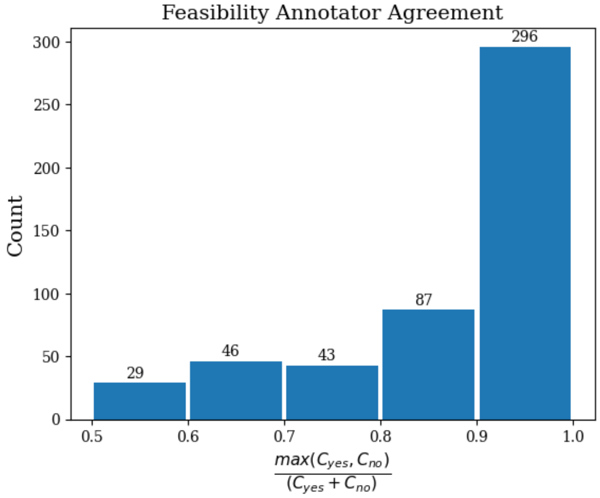


Fig. 7: The annotator feasibility labeling agreement for (app, task) pairs with demonstrations in MoTIF

2.4 App Category Clustering Visualizations

We provide the K-Means T-SNE cluster visualizations used in the (app, task) pairing process for each category of apps in Figure 8. These clusters decide whether an app's tasks are kept app-specific, paired to one or two other apps, or are category clustered. We zoom into the cluster visualization for the Weather Android app category in Figure 9. On the left, we see the cluster output for K-Means on the average task embedding (using FastText representations) for the commands written for weather apps. On the right we show the exact same clustering, but now color the points (*i.e.*, the written tasks) by which app they were originally written for. In the lower left corner of the cluster visualization is an isolated cluster for the com.droid27.transparentclockweather app. As its tasks form an isolated cluster, they are kept app-specific, while all other apps have (app, task) pairs obtained from the category clustering.

To actually select the category clustered tasks, we select natural language commands near each cluster's centroid. These serve as cluster representatives for our task demonstration data collection. So, for every Google Play Store app category, we perform K-Means with K=5, as we start by collecting demonstrations for five commands per app. Then, for apps that are chosen to be category clustered, we select the cluster representatives and collect demonstrations of these representatives for each weather app. For additional clarity, see Tables 1-3 for the (app, task) pairing method per app. Eventually, the goal is to collect all possible combinations of (app, task) pairs within a category.

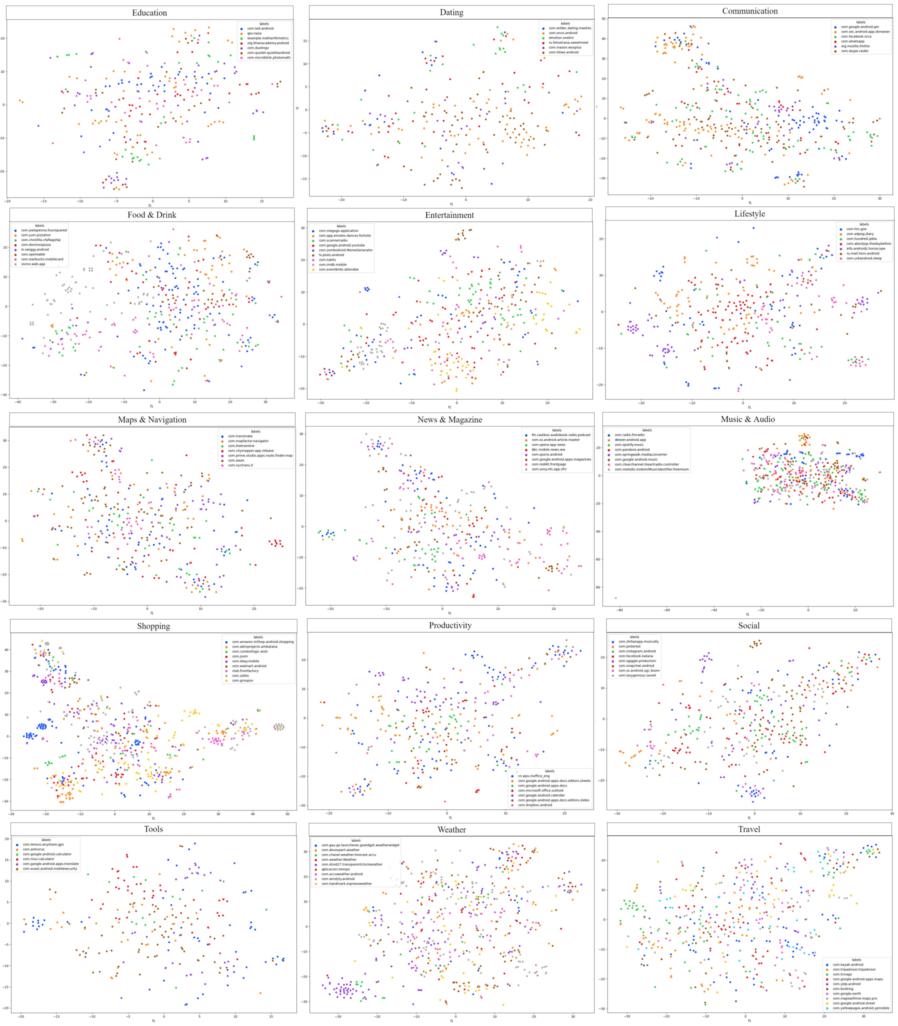


Fig. 8: T-SNE visualization of K-Means clusters for each Android Google Play Store Category. The visualizations are colored with the originating app label (and not the K-Means cluster label). These visualizations are used to inspect which apps should retain their app-specific tasks during the action sequence demonstration stage

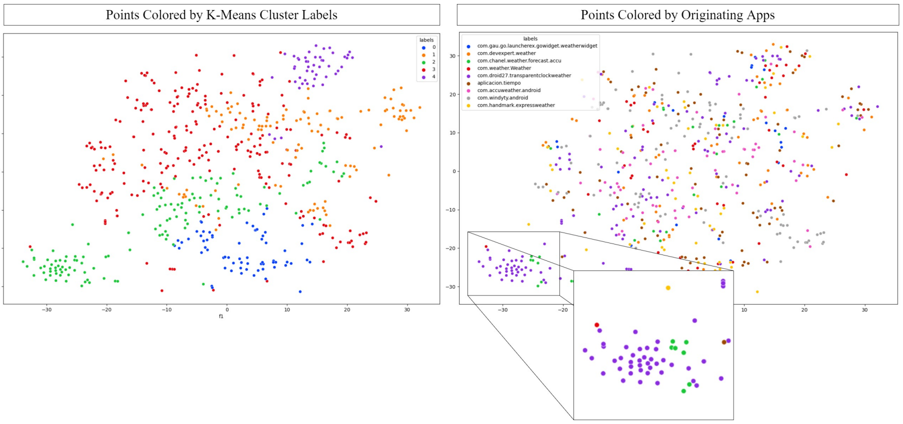


Fig. 9: T-SNE visualization of K-Means clusters on MoTIF commands from the Weather Google Play Store app category. Points represent MoTIF commands (represented by their mean FastText embedding). The left plot colors points by the clusters output by K-Means, while the right plot colors points by their originating app. In the lower left corner of both plots is a cluster (the green cluster on the left hand side), which when colored by the app the command was originally written for (on the right hand side), we see primarily comes from a single app, com.droid27.transparentclockweather. As a result, this app's commands will not be category clustered, and will stay paired with com.droid27.transparentclockweather

Table 4: Task feasibility F1 score using our MLP. We ablate input features and how action demonstration sequences are aggregated. The random baseline predicts a feasibility label given the train set distribution

$\mathbf{C}_{feas}$ Input Features	Demo Aggregation		
	Avg	Cat	LSTM
Random	20.1		
(a) View Hierarchy			
FastText			
ET	22.8	44.3	37.0
ET + ID	16.7	43.6	34.1
ET + ID + CLS	19.7	39.6	36.2
CLIP			
ET	27.0	48.4	35.9
ET + ID	28.0	50.9	36.2
ET + ID + CLS	29.6	49.2	35.2
Screen2Vec	25.9	33.7	36.0
(b) App Screen Image			
ResNet	31.3	41.9	35.9
Icons	0.4	40.0	15.2
CLIP	44.7	58.2	42.8
(c) Best Combination			
CLIP (Screen + ET + ID)	44.8	61.1	40.9

3 Task Feasibility Experiments

In Table 4(a), we have additional rows for which view hierarchy element attributes are included as input features to our feasibility classifier. The view hierarchy of an Android app contains several element attributes, including text (ET), resource-identifier (ID), and class (CLS) attributes. We ablate using one or multiple of these attributes and find that on average across demonstration aggregation type, the (ET + ID) input combination results in the best performance. Consequently, we keep it for our best results in the main text.

4 Task Automation Experiments

We further detail how task automation experiments are performed in a vision-language navigation paradigm in Section 4.1, where we describe the test-time environment. Then, we report performance when training VLN methods only on our data in Section 4.2. In Section 4.3, we evaluate our models from the main paper on different language inputs (high-level goal, low-level instruction, or both) at test-time and describe performance trends. Lastly, in Section 4.4 we include some additional results on generalization of tasks across apps for a subset of our baselines.

Table 5: Mobile app task complete and partial sequence accuracy on MoTIF when trained on MoTIF alone, or MoTIF and RicoSCA data for the Seq2Seq model. The training and testing language input are kept the same; input contains the high-level goal and low level step by step instructions

Model **Seq2Seq**	Train Data	MoTIF Test Split					
		App Seen			App Unseen		
		Action	Ground	Action + Ground	Action	Ground	Action + Ground
Complete	MoTIF	45.0	17.1	15.9	33.8	13.6	11.7
Partial		79.4	37.7	35.5	66.8	27.8	25.0
Complete	MoTIF + RicoSCA	68.5	22.5	22.5	54.3	18.0	17.7
Partial		89.5	40.4	40.1	81.7	31.3	30.6

Table 6: Mobile app task complete and partial sequence accuracy on MoTIF when trained on MoTIF alone, or MoTIF and RicoSCA data for the MOCA model. The training and testing language input are kept the same; input contains the high-level goal and low level step by step instructions

Model **MOCA**	Train Data	MoTIF Test Split					
		App Seen			App Unseen		
		Action	Ground	Action + Ground	Action	Ground	Action + Ground
Complete	MoTIF	37.8	16.2	12.3	24.6	17.0	13.2
Partial		66.0	34.9	29.9	60.4	32.0	27.7
Complete	MoTIF + RicoSCA	51.1	21.3	20.7	44.8	17.0	15.1
Partial		78.5	40.0	38.6	72.2	32.7	30.0

4.1 Test-time Evaluation of Seq2Seq and MOCA

We build an offline version of each Android app environment to approximate a complete state-action space graph at test time. We merge demonstrations we've collected across all samples. The nodes in this state-action space graph are unique 'views' of an application, *i.e.*, a particular screen within an action demonstration sequence. Nodes are connected by edges which represent the transition between any pair of screens. This transition is defined by the action class (clicking, typing, or swiping) and the location of the action taken at the current screen state (point or bounding box coordinates in the rendered app screen image).

4.2 Training Data Ablations

We also ran experiments with Seq2Seq and MOCA when trained only on MoTIF data instead of both MoTIF and RicoSCA. We include these comparisons for Seq2Seq and MOCA in Tables 5 and 6, respectively. Jointly training on both datasets consistently performs better across all metrics. Additionally, performance trends generally remain the same when comparing the app seen versus

Table 7: Mobile app task complete and partial sequence accuracy on MoTIF with various language inputs at test time for the Seq2Seq model. The training input contains the high level goal and low level step by step instructions

Model	Test	MoTIF Test Split					
Seq2Seq	Input	App Seen			App Unseen		
		Action	Ground	Action + Ground	Action	Ground	Action + Ground
Complete	High + Low	68.5	22.5	22.5	54.3	18.0	17.7
Partial		89.5	40.4	40.1	81.7	31.3	30.6
Complete	Low	47.1	18.6	18.0	27.1	13.9	13.9
Partial		73.7	36.6	33.9	43.6	22.6	21.2
Complete	High	30.9	15.3	14.7	18.9	11.7	8.8
Partial		68.1	31.6	29.5	59.1	24.0	19.8

app unseen test split: regardless of training data, accuracy is higher on the app seen test split. We report the joint training performance for these methods in the main text for a closer apples-to-apples comparison with Seq2Act.

4.3 Test-time Language Input Ablations

We include ablations for the trained models in the main text for all possible language inputs at test time. Seq2Seq and MOCA were trained on both high-level goal and low-level instructions, as their original models supported both inputs and obtained best performance with them in prior work. Seq2Act does not currently support high-level goal language input, so we cannot jointly evaluate both in a meaningful way. We benchmark models as close to their original architecture as possible, and leave adaptations to future work.

In the main text, all task automation results were reported on the same language input as was used during training to avoid confounding factors when analyzing generalization to new app environments. Thus, Seq2Seq and MOCA took both high-level and low-level command as input while Seq2Act took only low-level instruction. We now evaluate all possible input language ablations at test time. Evaluating the high-level goal input alone replicates what these models would be provided in practical application, as users would request high-level goals (and not provide step by step instruction). Our high-level input results are useful to evaluate generalization to downstream settings, but we also include results for low-level input alone or both high-level and low-level language instruction (where applicable, as Seq2Act cannot support both) in Tables 7, 8, and 9.

The Seq2Seq partial and complete sequence accuracy for action prediction show that having both high-level goal and low-level instruction inputs result in the best performance, followed by low-level instruction, and then high-level goal. On the other hand, MOCA performs quite similarly when both high-level goal and low-level instruction are input versus low-level instruction alone on action prediction. Additionally, there is less grounding performance degradation

Table 8: Mobile app task complete and partial sequence accuracy on MoTIF with various language inputs at test time for the MOCA model. The training input contains the high level goal and low level step by step instructions

Model **MOCA**	Test Input	MoTIF Test Split					
		App Seen			App Unseen		
		Action	Ground	Action + Ground	Action	Ground	Action + Ground
Complete Partial	High + Low	51.1 78.5	21.3 40.0	20.7 38.6	44.8 72.2	17.0 32.7	15.1 30.0
Complete Partial	Low	48.6 77.3	19.5 36.5	19.2 36.5	45.4 74.1	17.0 32.4	15.8 30.8
Complete Partial	High	13.5 43.6	19.5 38.8	8.4 26.1	11.4 41.1	18.6 33.5	6.9 21.2

Table 9: Mobile app task complete and partial sequence accuracy on MoTIF with various language inputs at test time for the Seq2Act model. The training input contains the low level step by step instructions

Model **Seq2Act**	Test Input	MoTIF Test Split					
		App Seen			App Unseen		
		Action	Ground	Action + Ground	Action	Ground	Action + Ground
Complete Partial	Low	98.8 99.7	27.6 64.4	27.6 64.3	94.9 98.9	23.5 62.2	23.5 61.7
Complete Partial	High	10.6 28.1	7.6 12.8	7.6 10.8	8.5 31.3	1.9 6.9	1.9 5.4

over the ablations, which may be a result of MOCA's more constrained test-time environment (which uses app type prediction to narrow the grounding prediction space).

Seq2Act performs best across all metrics when provided the low-level instruction at test time. This is expected, given that Seq2Act was trained on step by step instructions. For both test splits, the action and grounding accuracy is significantly higher with low-level input. As the VLN methods showed having both high-level and low-level inputs can improve performance, adapting Seq2Act to take both as input would be important in future work.

4.4 Generalization of Natural Language Commands across Apps

We lastly evaluate generalization of our task automation methods to natural language tasks. Specifically, we present results on two additional test splits: an app seen and task unseen app split (where the task was seen in other apps, but not the current) and an app unseen and task seen split. The former shows the easier setting of having seen the app environment with other tasks during training and the task with other apps during training, whereas the app unseen

Table 10: Mobile app task complete and partial sequence accuracy on MoTIF with various test splits for evaluating task generalization. The training and test-time input contains the high level goal and low level step by step instructions

Model	MoTIF Test Split					
	App Seen Task Unseen (Current App)			App Unseen Task Seen		
	Action	Ground	Action + Ground	Action	Ground	Action + Ground
Seq2Seq						
Complete	75.4	31.0	31.0	70.9	25.8	25.8
Partial	92.7	46.6	46.6	91.5	41.4	41.2
MOCA						
Complete	66.5	34.3	33.1	57.9	29.5	28.1
Partial	87.8	47.7	46.2	77.8	44.7	42.7

test split means the task was seen during training with other apps but the model has never seen any task in this particular app.

Intuitively, performance is consistently higher on the easier setting of app seen and task unseen (current app), as the model has had the chance to learn about both the app environment and task instruction, albeit independently. Comparing these task generalization results to the app generalization results in the main text (can also be found in Tables 7-9), the models can consistently generalize tasks across applications better than they can generalize to new environments.