In Proceedings of ACL 2018

Learning Prototypical Goal Activities for Locations

Tianyu Jiang and Ellen Riloff
School of Computing
University of Utah
Salt Lake City, UT 84112
{tianyu, riloff}@cs.utah.edu

Abstract

People go to different places to engage in activities that reflect their goals. For example, people go to restaurants to eat, libraries to study, and churches to pray. We refer to an activity that represents a common reason *why* people typically go to a location as a *prototypical goal activity (goal-act)*. Our research aims to learn goal-acts for specific locations using a text corpus and semi-supervised learning. First, we extract activities and locations that co-occur in goal-oriented syntactic patterns. Next, we create an *activity profile matrix* and apply a semi-supervised label propagation algorithm to iteratively revise the activity strengths for different locations using a small set of labeled data. We show that this approach outperforms several baseline methods when judged against goal-acts identified by human annotators.

1 Introduction

Every day, people go to different places to accomplish goals. People go to stores to buy clothing, go to restaurants to eat, and go to the doctor for medical services. People travel to specific destinations to enjoy the beach, go skiing, or see historical sites. For most places, people typically go there for a common set of reasons, which we will refer to as *prototypical goal activities (goal-acts)* for a location. For example, a prototypical goal-act for restaurants would be "*eat food*" and for IKEA would be "*buy furniture*".

Previous research has established that recognizing people's goals is essential for narrative text understanding and story comprehension (Schank and Abelson, 1977; Wilensky, 1978; Lehnert, 1981; Elson and McKeown, 2010; Goyal et al., 2013). Goals and plans are essential to understand people's behavior and we use our knowledge of prototypical goals to make inferences when reading. For example, consider the following pair of sentences: *"Mary went to the supermarket. She needed milk."* Most people will infer that Mary purchased milk, unless told otherwise. But a purchase event is not explicitly mentioned. In contrast, a similar sentence pair *"Mary went to the theatre. She needed milk."* feels incongruent and does not produce that inference. Recognizing goals is also critical for conversational dialogue systems. For example, if a friend tells you that they went to a restaurant, you might reply *"What did you eat?"*, but if a friend says that they went to Yosemite, a more appropriate response might be *"Did you hike?"* or *"Did you see the waterfalls?"*.

Our knowledge of prototypical goal activities also helps us resolve semantic ambiguity. For example, consider the following sentences:

(a) *She went to the kitchen and got chicken.*
(b) *She went to the supermarket and got chicken.*
(c) *She went to the restaurant and got chicken.*

In sentence (a), we infer that she retrieved chicken (e.g., from the refrigerator) but did not pay for it. In (b), we infer that she paid for the chicken but probably did not eat it at the supermarket. In (c), we infer that she ate the chicken at the restaurant. Note how the verb "*got*" maps to different presumed events depending on the location.

Our research aims to learn the prototypical goal-acts for locations using a text corpus. First, we extract activities that co-occur with locations in goal-oriented syntactic patterns. Next, we construct an *activity profile matrix* that consists of an activity vector (profile) for each of the locations. We then apply a semi-supervised label propagation algorithm to iteratively revise the activity profile strengths based on a small set of labeled locations.

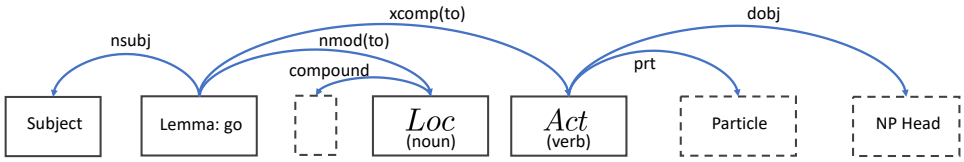

Figure 1: Dependency relation structure for "go to" pattern.

We also incorporate external resources to measure similarity between different activity expressions. Our results show that this semi-supervised learning approach outperforms several baseline methods in identifying the prototypical goal activities for locations.

2 Related Work

Recognizing plans and goals is fundamental to narrative story understanding (Schank and Abelson, 1977; Bower, 1982). Conceptual knowledge structures developed in prior work have shown the importance of this type of knowledge, including plans (Wilensky, 1978), goal trees (Carbonell, 1979), and plot units (Lehnert, 1981). Wilensky's research aimed to understand the actions of characters in stories by analyzing their goals, and their plans to accomplish those goals. For example, someone's goal might be to obtain food with a plan to go to a restaurant. Our work aims to learn prototypical goals associated with a location, to support similar inference capabilities during story understanding.

Goals and plans can also function to trigger *scripts* (Cullingford, 1978), such as the $RESTAURANT script. There has been growing interest in learning narrative event chains and script knowledge from large text corpora (e.g., (Chambers and Jurafsky, 2008, 2009; Jans et al., 2012; Pichotta and Mooney, 2014, 2016)). In addition, Goyal et al. (2010; 2013) developed a system to automatically produce plot unit representations for short stories. A manual analysis of their stories revealed that 61% of Positive/Negative Affect States originated from completed plans and goals, and 46% of Mental Affect States originated from explicitly stated or inferred plans and goals.

Elson & McKeown (2010) included plans and goals in their work on creating extensive story bank annotations that capture the knowledge needed to understand narrative structure. Researchers have also begun to explore NLP methods for recognizing the goals, desires, and plans of characters in stories. Recent work has explored techniques to detect wishes (desires) in natural language text (Goldberg et al., 2009) and identify desire fulfillment (Chaturvedi et al., 2016; Rahimtoroghi et al., 2017).

Graph-based semi-supervised learning has been successfully used for many tasks, including sentiment analysis (Rao and Ravichandran, 2009; Feng et al., 2013), affective event recognition (Ding and Riloff, 2016) and class-instance extraction (Talukdar and Pereira, 2010). The semi-supervised learning algorithm used in this paper is modeled after a framework developed by Zhu et al. (2003) based on harmonic energy minimization and a label propagation algorithm described in (Zhu and Ghahramani, 2002).

3 Learning Prototypical Goal Activities

Our aim is to learn the most prototypical goal-acts for locations. To tackle this problem, we first extract locations and related activities from a large text corpus. Then we use a semi-supervised learning method to identify the goal activities for individual locations. In the following sections we describe these processes in detail.

3.1 Location and Activity Extraction

To collect information about locations and activities, we use the 2011 Spinn3r dataset (Burton et al., 2011). Since our interest is learning about the activities of ordinary people in their daily lives, we use the Weblog subset of the Spinn3r corpus, which contains over 133 million blog posts.

We use the text data to identify activities that are potential goal-acts for a location. However we also need to identify locations and want to include both proper names (e.g., Disneyland) as well as nominals (e.g., store, beach), so Named Entity Recognition will not suffice. Consequently, we extract (Loc, Act) pairs using syntactic patterns.

First, we apply the Stanford dependency parser (Manning et al., 2014). We then extract sentences that match the pattern "go to X to Y" with the

	a_1 = buy book	a_2 = eat burger	...	a_m = pray
l_1 = McDonald's	.10	.30		.01
l_2 = Burger King	.12	.50		.02
l_3 = bookstore	.40	.02		.04
⋮		⋮		
l_n = church	.05	.01		.70

Table 1: An illustration of the activity profile matrix Y.

following conditions: (1) there exists a subject connecting to "go", (2) X has an **nmod** (nominal modifier) relation to "go" (lemma), (3) X is a noun or noun compound, (4) Y has an **xcomp** relation (open clausal complement) with "go", and (5) Y is a verb. Figure 1 depicts the intended syntactic structure, which we will informally call the "go to" pattern. For sentences that match this pattern, we extract X as a location and Y as an activity. If the verb is followed by a particle and/or noun phrase (NP), then we also include the particle and head noun of the NP. For example, we extract activities such as "*pray*", "*clean up*", and "*buy sweater*".

This syntactic structure was chosen to identify activities that are described as being the reason why someone went to the location. However it is not perfect. In some cases, X is not a location (e.g., *"go to great lengths to ..."* yields "*lengths*" as a location), or Y is not a goal-act for X (e.g., *"go to the office to retrieve my briefcase ..."* yields "*retrieve briefcase*" which is not a prototypical goal for "*office*"). Interestingly, the pattern extracts some nominals that are not locations in a strict sense, but behave as locations. For example, *"go to the doctor"* extracts "*doctor*" as a location. Literally a doctor is a person, but in this context it really refers to the doctor's office, which is a location. The pattern also extracts entities such as "*roof*", which are not generally thought of as locations but do have a fixed physical location. Other extracted entities are virtual but function as locations, such as "*Internet*". For the purposes of this work, we use the term **location** in a general sense to include any place or object that has a physical, virtual or implied location.

The "go to" pattern worked quite well at extracting (Loc, Act) pairs, but in relatively small quantities due to the very specific nature of the syntactic structure. So we tried to find additional activities for those locations. Initially, we tried harvesting activities that occurred in close proximity (within 5 words) to a known location, but the results were too noisy. Instead, we used the pattern "Y in/at X" with the same syntactic constraints for Y (the extracted activity) and X (a location extracted by the "go to" pattern).

We discovered many sentences in the corpus that were exactly or nearly the same, differing only by a few words, which resulted in artificially high frequency counts for some (Loc, Act) pairs. So we filtered duplicate or near-duplicate sentences by computing the longest common substring of sentence pairs that extracted the same (Loc, Act). If the shared substring had length ≥ 5, then we discarded the "duplicate" sentence.

Finally, we applied three filters. To keep the size of the data manageable, we discarded locations and activities that were each extracted with frequency < 30 by our patterns. And we manually filtered locations that are Named Entities corresponding to cities or larger geo-political regions (e.g., provinces or countries). Large regions defined by government boundaries fall outside the scope of our task because the set of activities that typically occur in (say) a city or country is so broad. Finally, we added a filter to try to remove extremely general activities that can occur almost anywhere (e.g., visit). If an activity co-occurred with $> 20\%$ of the extracted (distinct) locations, then we discarded it.

After these filters, we extracted 451 distinct locations, 5143 distinct activities, roughly $200,000$ distinct (Loc, Act) pairs, and roughly $500,000$ instances of (Loc, Act) pairs.

3.2 Activity Profiles for Locations

We define an *activity profile matrix* Y of size $n \times m$, where n is the number of distinct locations and m is the number of distinct activities. $Y_{i,j}$ represents the strength of the jth activity a_j being a goal-act for l_i. We use $\mathbf{y}_i \in \mathbb{R}^m$ to denote the ith row of Y. Table 1 shows an illustration of (partial) activity profiles for four locations.[1] Our goal is

[1]Not actual values, for illustration only.

to learn the $Y_{i,j}$ values so that activities with high strength are truly goal-acts for location l_i.

We could build the activity profile for location l_i using the co-occurrence data extracted from the blog corpus. For example, we could estimate $P(a_j \mid l_i)$ directly from the frequency counts of the activities extracted for l_i. However, a high co-occurrence frequency doesn't necessarily mean that the activity represents a prototypical goal. For example, the activity *"have appointment"* frequently co-occurs with *"clinic"* but doesn't reveal the underlying reason for going to the clinic (e.g., probably to see a doctor or undergo a medical test). To appreciate the distinction, imagine that you asked a friend why she went to a health clinic, and she responded with "because I had an appointment". You would likely view her response as being snarky or evasive (i.e., she didn't want to tell you the reason). In Section 4, we will evaluate this approach as a baseline and show that it does not perform well.

3.3 Semi-Supervised Learning of Goal-Act Probabilities

Our aim is to learn the activity profiles for locations using a small amount of labeled data, so we frame this problem as a semi-supervised learning task. Given a small number of "seed" locations coupled with predefined goal-acts, we want to learn the goal-acts for new locations.

3.3.1 Location Similarity Graph

We use $l_i \in L$ to represent location l_i, where $|L| = n$. We define an undirected graph $G = (V, E)$ with vertices representing locations ($|V| = n$) and edges $E = V \times V$, such that each pair of vertices v_i and v_k is connected with an edge e_{ik} whose weight represents the similarity between l_i and l_k.

We can then represent the edge weights as an $n \times n$ symmetric weight matrix W indicating the similarity between locations. There could be many ways to define the weights, but for now we use the following definition from (Zhu et al., 2003), where σ^2 is a hyper-parameter[2]:

$$W_{i,k} = \exp\left(-\frac{1}{\sigma^2}\left(1 - sim\left(l_i, l_k\right)\right)\right) \quad (1)$$

To assess the similarity between locations, we measure the cosine similarity between vectors of their co-occurrence frequencies with activities. Specifically, let matrix $F_{n\times m} = [\mathbf{f}_1, ..., \mathbf{f}_n]^{\mathrm{T}}$ where $\mathbf{f}_i$ is a vector of length m capturing the co-occurrence frequencies between location l_i and each activity a_j in the extracted data (i.e., $F_{i,j}$ is the number of times that activity a_j occurred with location l_i). We then define location similarity as:

$$sim(l_i, l_k) = \frac{\mathbf{f}_i^{\mathrm{T}}\mathbf{f}_k}{\|\mathbf{f}_i\|\|\mathbf{f}_k\|} \quad (2)$$

3.3.2 Initializing Activity Profiles

We use semi-supervised learning with a set of "seed" locations from human annotations, and another set of locations that are unlabeled. So we subdivide the set of locations into $S = \{l_1, ..., l_s\}$, which are the seed locations, and $U = \{l_{s+1}, ..., l_{s+u}\}$, which are the unlabeled locations, such that $s + u = n$. For an unlabeled location $l_i \in U$, the initial activity profile is the normalized co-occurrence frequency vector $\overline{\mathbf{f}}_i$.

For each seed location $l_i \in S$, we first automatically construct an activity profile vector $\overline{\mathbf{h}}_i$ based on the gold goal-acts which were obtained from human annotators as described in Section 4.1. All activities not in the gold set are assigned a value of zero. Each activity a_j in the gold set is assigned a probability $P(a_j \mid l_i)$ based on the gold answers. However, the gold goal-acts may not match the activity phrases found in the corpus (see discussion in Section 4.3), so we smooth the vector created with the gold goal-acts by averaging it with the normalized co-occurrence frequency vector $\overline{\mathbf{f}}_i$ extracted from the corpus.

The activity profiles of seed locations stay constant through the learning process. We use $\mathbf{y}_i^0$ to denote the initial activity profiles. So when $l_i \in S$, $\mathbf{y}_i^0 = (\overline{\mathbf{f}}_i + \overline{\mathbf{h}}_i)/2$.

3.3.3 Learning Goal-Act Strengths

We apply a learning framework developed by (Zhu et al., 2003) based on harmonic energy minimization and extend it to multiple labels. Intuitively, we assume that similar locations should share similar activity profiles,[3] which motivates the following objective function over matrix Y:

$$\begin{aligned} &\arg\min_{Y} \sum_{i,k} W_{i,k}\|\mathbf{y}_i - \mathbf{y}_k\|^2, \\ &\text{s.t. } \mathbf{y}_i = \mathbf{y}_i^0 \text{ for each } l_i \in S \end{aligned} \quad (3)$$

Let $D = (d_i)$ denote an $n \times n$ diagonal matrix where $d_i = \sum_{k=1}^{n} W_{i,k}$. Let's split Y by the sth

[2] We use the same value $\sigma^2 = 0.03$ as (Zhu et al., 2003).

[3] This is a heuristic but is not always true.

row: $Y = \begin{bmatrix} Y_s \\ Y_u \end{bmatrix}$, then split W(similarly for D) into four blocks by the sth row and column:

$$W = \begin{bmatrix} W_{ss} & W_{su} \\ W_{us} & W_{uu} \end{bmatrix} \quad (4)$$

From (Zhu et al., 2003), Eq (3) is given by:

$$Y_u = (D_{uu} - W_{uu})^{-1} W_{us} Y_s \quad (5)$$

We then use the label propagation algorithm described in (Zhu and Ghahramani, 2002) to compute Y:

Algorithm 1

repeat
  $Y \leftarrow D^{-1}WY$
  Clamp $\mathbf{y}_i = \mathbf{y}_i^0$ for each $l_i \in S$
until convergence

3.3.4 Activity Similarity

One problem with the above algorithm is that it only takes advantage of relations between vertices (i.e., locations). If there are intrinsic relations between activities, they could be exploited as a complementary source of information to benefit the learning. Intuitively, different pairs of activities share different similarities, e.g., "*eat burgers*" should be more similar to "*have lunch*" than "*read books*".

Under this idea, similar to the previous location similarity weight matrix W, we want to define an activity similarity weight matrix $A_{m \times m}$ where $A_{i,k}$ indicates the similarity weight between activity a_i and a_k:

$$A_{i,k} = \exp\left(-\frac{1}{\sigma^2}\left(1 - sim\left(a_i, a_k\right)\right)\right) \quad (6)$$

where σ^2 is the same as in Eq (1).

We explore 3 different similarity functions $sim(a_i, a_k)$ based on co-occurrence with locations, word matching, and embedding similarities.

First, similar to Eq (2), we can use each activity's co-occurrence frequency with all locations as its location profile and define a similarity score based on cosine values of location profile vectors:

$$sim^L(a_i, a_k) = \frac{\mathbf{g}_i^{\mathrm{T}} \mathbf{g}_k}{\|\mathbf{g}_i\| \|\mathbf{g}_k\|} \quad (7)$$

where the predefined co-occurrence frequency matrix $F = [\mathbf{f}_1, ..., \mathbf{f}_n]^{\mathrm{T}} = [\mathbf{g}_1, ..., \mathbf{g}_m]$.

As a second option, the similarity between activities can often be implied by their lexical overlap, e.g., two activities sharing the same verb or noun might be related. For each word belonging to any of our activities, we use WordNet (Miller, 1995) to find its synonyms. We also include the word itself in the synonym set. If the synonym sets of two words overlap, we call these two words "**match**". Then we define the lexical overlap similarity function between a_i and a_k:

$$sim^O(a_i, a_k) = \begin{cases} 1 & \text{if verb and noun match} \\ 0.5 & \text{if verb or noun match} \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

As a third option, we can use 300-dimension word embedding vectors (Pennington et al., 2014) trained on 840 billion tokens of web data to compute semantic similarity. We compute an activity's embedding as the average of its words' embeddings. Let $sim^E(a_i, a_k)$ be the cosine value between the embedding vectors of a_i and a_k:

$$sim^E(a_i, a_k) = cos\langle \text{Embed}(a_i), \text{Embed}(a_k) \rangle \quad (9)$$

Finally, we can plug these similarity functions into Eq (6). We use A^L, A^O, A^E to denote the corresponding matrix. We can also plug in multiple similarity metrics such as $(sim^L + sim^E)/2$ and use combination symbols A^{L+E} to denote the matrix.

3.3.5 Injecting Activity Similarity

Once we have a similarity matrix for activities, the next question is how will it help with the activity profile computation? Recall from Eq (5), we know that the activity profile of an unlabeled location can be represented by a linear combination of other locations' activity profiles. The activity profile matrix Y is an $n \times m$ matrix where each row is the activity profile for a location. We can also view Y as a matrix whose each column is the location profile for an activity. Using the same idea, we can make each column approximate a linear combination of its highly related columns (i.e., the location profile of an activity will become more similar to the location profiles of its similar activities). Our expectation is that this approximation will help improve the quality of Y.

By being right multiplied by matrix A, Y gets updated from manipulating its columns (activities) as well. We modify the algorithm accordingly as below:

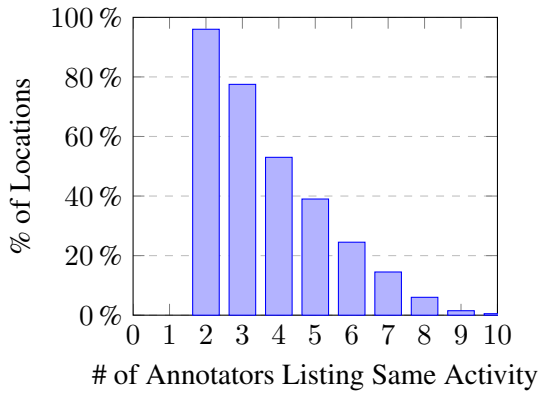

Figure 2: Percentage of locations that have at least one goal-act assigned by multiple annotators.

Algorithm 2

repeat
$Y \leftarrow D^{-1}WYA$
Clamp $\mathbf{y}_i = \mathbf{y}_i^0$ for each $l_i \in S$
until convergence

4 Evaluation

4.1 Gold Standard Data

Since this is a new task and there is no existing dataset for evaluation, we use crowd-sourcing via Amazon Mechanical Turk (AMT) to acquire gold standard data. First, we released a qualification test containing 15 locations along with detailed annotation guidelines. 25 AMT workers finished our assignment, and we chose 15 of them who did the best job following our guidelines to continue. We gave the 15 qualified workers 200 new locations, consisting of 152 nominals and 48 proper names,[4] randomly selected from our extracted data and set aside as test data. For each location, we asked the AMT workers to complete the following sentence:

People go to *LOC* to _____ _____
VERB NOUN

LOC was replaced by one of the 200 locations. Annotators were asked to provide an activity that is the primary reason why a person would go to that location, in the form of just a VERB or a VERB NOUN pair. Annotators also had the option to label a location as an "ERROR" if they felt that the provided term is not a location, since our location extraction was not perfect.

[4]Same distribution as in the whole location set.

Only 10 annotators finished labeling our test cases, so we used their answers as the gold standard. We discarded 12 locations that were labeled as an "ERROR" by ≥ 3 workers.[5] This resulted in a test set of 188 locations paired with 10 manually defined goal-acts for each one.

A key question that we wanted to investigate through this manual annotation effort is to know whether people truly do associate the same prototypical goal activities with locations. To what extent do people agree when asked to list goal-acts? Also, some places clearly have a smaller set of goal-acts than others. For example, the primary reason to go to an airport is to catch a flight, but there's a larger set of common reasons why people go to Yosemite (e.g.,"*hiking camping*", "*rock climbing*", "*see waterfalls*", etc.).

Complicating matters, the AMT workers often described the same activity with different words (e.g., "*buy book*" vs. "*purchase book*"). Automatically recognizing synonymous event phrases is a difficult NLP problem in its own right.[6] So solely for the purpose of analysis, we manually merged activities that have a nearly identical meaning. We were extremely conservative and did not merge similar or related phrases that were not synonymous because the granularity of terms may matter for this task (e.g., we did not merge "*eat burger*" and "*eat lunch*" because one may apply to a specific location while the other does not).

Figure 2 shows the results of our analysis. Only 1 location was assigned exactly the same goal-act by all 10 annotators. But at least half (5) of the annotators listed the same goal-act for 40% of the locations. And nearly 80% of locations had one or more goal-acts listed by ≥ 3 people. These results show that people often do share the same associations between prototypical goal-acts and locations. These results are also very conservative because many different answers were also similar (e.g. "*eat burger*", "*eat meal*").

In Table 2 we show examples of locations and the goal-acts listed for them by the human annotators. If multiple people gave the same answer, we show the number in parentheses. For example, given the location "Toys R Us", 9 people listed "*buy toys*" as a goal-act and 1 person listed "*browse gifts*". We see from Table 2 that

[5]We found that the workers rarely used the "ERROR" label, so setting this threshold to be 3 was a strong signal.

[6]We tried using WordNet synsets to conflate phrases, but it didn't help much.

Location	Gold Goal-Acts
Toys R Us	buy toys (9), browse gifts
sink	wash hands (7), wash dishes (3)
airport	catch flight (7), board planes, take airplane, take trips
bookstore	buy books (6), browse books (2), browse bestsellers, read book
lake	go fishing (3), go swimming (3), drive boat (2), ride boat, see scenery
chiropractor	get treatment (3), adjust backs (3), alleviate pain (2), get adjustment, get aligned
Chinatown	buy goods (2), buy duck, buy souvenirs, eat dim sum, eat rice, eat wontons, find Chinese, speak Chinese, visit restaurants

Table 2: Goal-acts provided by human annotators.

some locations yield very similar sets of goal-acts (e.g., sink, airport, bookstore), while other locations show more diversity (e.g., lake, chiropractor, Chinatown).

4.2 Baselines

To assess the difficulty of this NLP task, we created 3 baseline systems for comparison with our learning approach. All of these methods take the list of activities that co-occurred with a location l_i in our extracted data and rank them.

The first baseline, **FREQ**, ranks the activities based on the co-occurrence frequency $F_{i,j}$ between l_i and a_j in our patterns. The second baseline, **PMI**, ranks the activities using point-wise mutual information. The third baseline, **EMBED**, ranks the activities based on the cosine similarity of the semantic embedding vectors for l_i and a_j. We use GloVe (Pennington et al., 2014) 300-dimension embedding vectors pre-trained on 840 billion tokens of web data. For locations and activities with multiple words, we create an embedding by averaging the vectors of their constituent words.

4.3 Matching Activities

The gold standard contains a set of goal-acts for each location. Since the same activity can be expressed with many different phrases, the only way to truly know whether two phrases refer to the same activity is manual evaluation, which is expensive. Furthermore, many activities are very similar or highly related, but not exactly the same. For example, "*eat burger*" and "*eat food*" both describe eating activities, but the latter is more general than the former. Considering them to be the same is not always warranted (e.g., "*eat burger*" is a logical goal-act for McDonald's but not for Baskin-Robbins which primarily sells ice cream). As another example, "*buy chicken*" and "*eat chicken*" refer to different events (buying and eating) so they are clearly not the same semantically. But at a place like KFC, buying chicken implies eating chicken, and vice versa, so they seem like equally good answers as goal-acts for KFC. Due to the complexities of determining which gold standard answers belong in equivalence classes, we considered all of the goal-acts provided by the human annotators to be acceptable answers.

	MRR_E	MRR_P	TOP1	TOP2	TOP3
EMBED	0.02	0.09	0.05	0.08	0.12
PMI	0.20	0.33	0.25	0.36	0.41
FREQ	0.23	0.34	0.23	0.32	0.40
AP	0.28	0.38	0.29	0.41	0.47
AP+A^L	0.28	0.40	0.32	**0.44**	0.49
AP+A^O	0.23	0.33	0.24	0.35	0.43
AP+A^E	0.25	0.36	0.28	0.40	0.47
AP+A^{L+E}	**0.29**	**0.42**	**0.35**	**0.44**	**0.52**

Table 3: Scores for MRR and Top k results.

To determine whether an activity a_j produced by our system matches any of the gold goal-acts for a location l_i, we report results using two types of matching criteria. **Exact Match** judges a_j to be a correct answer for l_i if (1) it exactly matches (after lemmatization) any activity in l_i's gold set, or (2) a_j's *verb* and *noun* both appear in l_i's gold set, though possibly in different phrases. For example, if a gold set contains "*buy novels*" and "*browse books*", then "*buy books*" will be a match.

Since Exact Match is very conservative, we also define a **Partial Match** criterion to give 50% credit for answers that partially overlap with a gold answer. An activity a_j is a partial match for l_i if either its *verb* or *noun* matches any of the activities in l_i's gold set of goal-acts. For example, "*buy burger*" would be a partial match with "*buy food*" because their verbs match.

4.4 Evaluation Metrics

All of our methods produce a ranked list of hypothesized goal-acts for a location. So we use Mean Reciprocal Rank (MRR) to judge the quality of the top 10 activities in each ranked list. We report two types of MRR scores.

MRR based on the Exact Match criteria (MRR_E) is computed as follows, where n is the

number of locations in the test set:

$$\text{MRR}_\text{E} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{\text{rank of } 1^{st} \text{ Exact Match}} \quad (10)$$

We also compute MRR using both the Exact Match and Partial Match criteria. First, we need to identify the "best" answer among the 10 activities in the ranked list, which depends both on each activity's ranking and its matching score. The matching score for activity a_j is defined as:

$$\text{score}(a_j) = \begin{cases} 1 & \text{if } a_j \text{ is an Exact Match} \\ 0.5 & \text{if } a_j \text{ is a Partial Match} \\ 0 & \text{otherwise} \end{cases}$$

Given 10 ranked activities a_1 ... a_{10} for l_i, we then compute:

$$\text{best\_score}(l_i) = \max_{j=1..10} \frac{\text{score}(a_j)}{\text{rank}(a_j)}$$

And then finally define MRR_P as follows:

$$\text{MRR}_\text{P} = \frac{1}{n}\sum_{i=1}^{n}\text{best\_score}(l_i) \quad (11)$$

4.5 Experimental Results

Unless otherwise noted, all of our experiments report results using 4-fold cross-validation on the 200 locations in our test set. We used 4 folds to ensure 50 seed locations for each run (i.e., 1 fold for training and 3 folds for testing).

The first two columns of Table 3 show the MRR results under Exact Match and Partial Match conditions. The first 3 rows show the results for the baseline systems, and the remaining rows show results for our Activity Profile (AP) semi-supervised learning method. We show results for 5 variations of the algorithm: **AP** uses Algorithm 1, and the others use Algorithm 2 with different Activity Similarity measures: **AP+A$^\mathbf{L}$** (location profile similarity), **AP+A$^\mathbf{O}$** (overlap similarity), **AP+A$^\mathbf{E}$** (embedding similarity), and **AP+A$^{\mathbf{L+E}}$** (location profiles plus embeddings).

Table 3 shows that our AP algorithm outperforms all 3 baseline methods. When adding Activity Similarity into the algorithm, we find that A^L slightly improves performance, but A^O and A^E do not. However, we also tried combining them and obtained improved results by using A^L and A^E together, yielding an MRR_P score of 0.42.

To gain more insight about the behavior of the models, Table 3 also shows results for the top-ranked 1, 2, and 3 answers. For these experiments, the system gets full credit if any of its top k answers exactly matches the gold standard, or 50% credit if a partial match is among its top k answers. These results show that our AP method produces more correct answers at the top of the list than the baseline methods.

Table 4 shows six locations with their gold answers and the Top 3 goal-acts hypothesized by our best AP system and the PMI and FREQ baselines. The activities in **boldface** were deemed correct (including Partial Match). For "bookstore" and "pharmacy", all of the methods perform well. Note the challenge of recognizing that different phrases mean essentially the same thing (e.g., "*fill prescription*", "*pick up prescription*", "*find medicine*"). For "university" and "Meijer", the AP method produces more appropriate answers than the baseline methods. For "market" and "phone", all three methods struggle to produce good answers. Since "market" is polysemous, we see activities related to both stores and financial markets. And "phone" arguably is not a location at all, but most human annotators treated it as a virtual location, listing goal-acts related to telephones. However our algorithm considered phones to be similar to computers, which makes sense for today's smartphones. In general, we also observed that Internet sites behave as virtual locations in language (e.g., "*I went to YouTube...*").

4.6 Discussion

The goal-acts learned by our system were extracted from the Spinn3r dataset, while the gold standard answers were provided by human annotators, so the same (or very similar) activities are often expressed in different ways (see Section 4.3). This raises the question: what is the upper bound on system performance when evaluating against human-provided goal-acts? To answer this, we compared all of the activities that co-occurred with each location in the corpus against its gold goal-acts. Only 36% of locations had at least one gold goal-act among its extracted activities when matching identical strings (after lemmatization). Because of this issue, our Exact Match criteria also allowed for combining verbs and nouns from different gold answers. Under this Exact Match criteria, 73% of locations had at least one gold goal-act

Location	Gold Activity List	AP+A^{L+E} Top 3	PMI Top 3	FREQ Top 3
bookstore	buy book (6) browse book (2) browse bestseller read book	**buy book** **purchase book** **see book**	**buy copy** **purchase book** **buy book**	**buy book** **browse** **find book**
pharmacy	get drug (4) fill prescription (3) get prescription (2) buy medicine	**find medicine** **get prescription** **pick up prescription**	**buy pill** **fill prescription** **pick up prescription**	**buy pill** **fill prescription** **pick up prescription**
university	get degree (4) gain education (5) watch sport	**gain education** **further education** **gain knowledge**	study law study psychology pursue study	enrol[7] enroll take class
Meijer	buy grocery (8) buy cream obtain grocery	**buy item** go shopping get item	check out deal have shopping post today	get item save money check out
market	buy grocery (6) buy fresh, buy goods buy shirt, find produce	make money eat out eat lunch	have demand increase competition lead player	trade intervene make money
phone	make call (4), ERROR (2) answer call, call friend have conversation stop ring	play game browse website view website	put number have number put card	plug glance have number

Table 4: Examples of Top 3 hypothesized prototypical goal activities.

among the extracted activities, so this represents an upper bound on performance using this metric. Under the Partial Match criteria, 98% of locations had at least one gold goal-act among the extracted activities, but only 50% credit was awarded for these cases so the maximum score possible would be ∼86%.

We also manually inspected 200 gold locations to analyze their types. We discovered some related groups, but substantial diversity overall. The largest group contains ∼20% of the locations, which are many kinds of stores (e.g., Ikea, WalMart, Apple store, shoe store). Even within a group, different locations often have quite different sets of co-occurring activities. In fact, we discovered some spelling variants (e.g., "WalMart" and "wal mart"), but they also have substantially different activity vectors (e.g., because one spelling is much more frequent), so the model learns about them independently.[8] Other groups include restaurants (∼5%), home-related (e.g., bathroom) (∼5%), education (∼5%), virtual (e.g., Wikipedia) (∼3%), medical (∼3%) and landscape (e.g., hill) (∼3%). It is worth noting that our locations were extracted by two syntactic patterns and it remains to be seen if this has brought in any bias — detecting location nouns (especially nominals) is a challenging problem in its own right.

[7] A lemmatization error for the verb "enrolled".

[8] Of course normalizing location names beforehand may be beneficial in future work.

5 Conclusions and Future Work

We introduced the problem of learning prototypical goal activities for locations. We obtained human annotations and showed that people do associate prototypical goal-acts with locations. We then created an activity profile framework and applied a semi-supervised label propagation algorithm to iteratively update the activity strengths for locations. We demonstrated that our learning algorithm identifies goal-acts for locations more accurately than several baseline methods.

However, this problem is far from solved. Challenges also remain in how to evaluate the accuracy of goal knowledge extracted from text corpora. Nevertheless, our work represents a first step toward learning goal knowledge about locations, and we believe that learning knowledge about plans and goals is an important direction for natural language understanding research. In future work, we hope to see if we can take advantage of more contextual information as well as other external knowledge to improve the recognition of goal-acts.

Acknowledgments

We are grateful to Haibo Ding for valuable comments on preliminary versions of this work.

References

Gordon H Bower. 1982. Plans and Goals in Understanding Episodes. *Advances in Psychology*, 8:2–15.

K. Burton, N. Kasch, and I. Soboroff. 2011. The ICWSM 2011 Spinn3r Dataset. In *Proceedings of the Fifth Annual Conference on Weblogs and Social Media (ICWSM-2011)*.

J. G. Carbonell. 1979. *Subjective Understanding: Computer Models of Belief Systems*. Ph.D. thesis, Yale University.

Nathanael Chambers and Dan Jurafsky. 2008. Unsupervised Learning of Narrative Event Chains. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL/HLT-2008)*.

Nathanael Chambers and Dan Jurafsky. 2009. Unsupervised Learning of Narrative Schemas and Their Participants. 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*.

Snigdha Chaturvedi, Dan Goldwasser, and Hal Daumé III. 2016. Ask, and Shall You Receive? Understanding Desire Fulfillment in Natural Language Text. In *Processings of the 30th AAAI Conference on Artificial Intelligence (AAAI-2016)*.

Richard Edward Cullingford. 1978. *Script Application: Computer Understanding of Newspaper Stories*. Ph.D. thesis, Yale University.

Haibo Ding and Ellen Riloff. 2016. Acquiring Knowledge of Affective Events from Blogs using Label Propagation. In *Processings of the 30th AAAI Conference on Artificial Intelligence (AAAI-2016)*.

David Elson and Kathleen McKeown. 2010. Building a Bank of Semantically Encoded Narratives. In *Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC-2010)*.

Song Feng, Jun Seok Kang, Polina Kuznetsova, and Yejin Choi. 2013. Connotation Lexicon: A Dash of Sentiment Beneath the Surface Meaning. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL-2013)*.

Andrew B Goldberg, Nathanael Fillmore, David Andrzejewski, Zhiting Xu, Bryan Gibson, and Xiaojin Zhu. 2009. May all your wishes come true: A study of wishes and how to recognize them. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT/NAACL-2009)*.

Amit Goyal, Ellen Riloff, and Hal Daumé III. 2010. Automatically producing plot unit representations for narrative text. In *Proceedings of the 2010 Conference on Empirical Methods on Natural Language Processing (EMNLP-2010)*.

Amit Goyal, Ellen Riloff, and Hal Daumé III. 2013. A Computational Model for Plot Units. *Computational Intelligence*, 29(3):466–488.

Bram Jans, Steven Bethard, Ivan Vulić, and Marie Francine Moens. 2012. Skip n-grams and ranking functions for predicting script events. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL-2012)*.

Wendy G Lehnert. 1981. Plot Units and Narrative Summarization. *Cognitive Science*, 5(4):293–331.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL-2014) System Demonstrations*.

George A Miller. 1995. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11):39–41.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods on Natural Language Processing (EMNLP-2014)*.

Karl Pichotta and Raymond Mooney. 2014. Statistical script learning with multi-argument events. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL-2014)*.

Karl Pichotta and Raymond J Mooney. 2016. Learning Statistical Scripts with LSTM Recurrent Neural Networks. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI-2016)*.

Elahe Rahimtoroghi, Jiaqi Wu, Ruimin Wang, Pranav Anand, and Marilyn Walker. 2017. Modelling Protagonist Goals and Desires in First-Person Narrative. In *Proceedings of the 18th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL-2017)*.

Delip Rao and Deepak Ravichandran. 2009. Semi-supervised Polarity Lexicon Induction. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL-2009)*.

Roger C Schank and Robert Abelson. 1977. *Scripts, Plans, Goals and Understanding*. Lawrence Erlbaum.

Partha Pratim Talukdar and Fernando Pereira. 2010. Experiments in Graph-based Semi-supervised Learning Methods for Class-instance Acquisition. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL-2010)*.

Robert Wilensky. 1978. *Understanding Goal-based Stories*. Ph.D. thesis, Yale University.

Xiaojin Zhu and Zoubin Ghahramani. 2002. Learning from Labeled and Unlabeled Data with Label Propagation. Technical report, Carnegie Mellon University.

Xiaojin Zhu, Zoubin Ghahramani, and John D Lafferty. 2003. Semi-supervised Learning Using Gaussian Fields and Harmonic Functions. In *Proceedings of the 20th International Conference on Machine Learning (ICML-2003)*.