

Scaling Mobile Alternate Reality Games with Geo-Location Translation

Sanjeet Hajarnis, Brandon Headrick, Aziel Ferguson, and Mark O. Riedl

School of Interactive Computing, Georgia Institute of Technology
Atlanta, Georgia USA

{sanjeet,brandonheadrick,aziel.ferguson,riedl}@gatech.edu,

Abstract. Alternate Reality Games (ARGs) are interactive narrative experiences that engage the player by layering a fictional world over the real world. Mobile ARGs use geo-location aware devices to track players as they visit real-world locations to progress the story. ARG stories are often geo-specific, requiring players to visit specific locations in the world and, as a result, ARGs are played infrequently and only by those who live within proximity of the locations that the stories reference. We present a solution to the geo-specificity problem called *location translation*, which transforms ARG stories from one geographical location to another, making them playable anywhere. We show that location translation addresses fundamental scalability challenges that arise from geo-specificity.

1 Alternate Reality Games

Alternate Reality Games (ARGs) have recently emerged as a new genre of games. ARGs are interactive narrative experiences that engage the player by layering a fictional world over the real world; as players act in the real world their actions influence the state of the fictional world. With the advent of geo-location aware mobile devices, ARGs make use of the actual, physical world as the environment for which the game plays out [5]. Typically, a *game master* runs the game and monitors players from remote in order to make adjustments to the narrative arc or trigger branching points as necessary. Many ARGs utilize *confederate actors* planted throughout the physical world to interact with players.

The ARG genre is limited in two significant ways. First, supporting an ARG is effort-intensive on the part of human game masters and confederates. Second, ARG stories can be *geo-specific*—they reference real world geographical locations and landmarks requiring visits to these places to advance the narrative. Consequently, a particular ARG story is fixed to a specific region of the real world; a story set in New York City cannot be played in London without substantial re-authoring. Taken together, the scalability limitations result in a situation where ARGs are played infrequently and can only be played by those who live within proximity of the region in which the game story is set.

How can one reduce the need for human confederate actors and game master? The use of virtual agents, exemplified by the tour guides described by Lim and Aylett [3], can replace confederate actors. However, such systems do not

overcome geo-specificity limitations; agents can only perform in the vicinity of fixed landmarks. Efforts are under way to automate game mastering as well. The Spyfeet mobile ARG [6] uses a rule-base implemented in Inform7TM to control game progression. The Spyfeet story does not reference specific geographical locations and instead requires certain activities such as finding an NPC that has been mapped to an arbitrary geo-location. Likewise, the Backseat Playground [1] is a mobile ARG system that triggers story elements based on features of the local environment as one rides in the back seat of a car. Backseat Playground story content also does not make specific reference to location or landmark.

Location geo-specificity grounds the play experience by linking story content to physical space, but at the cost of limiting who can play. In this paper, we present an approach to overcoming the geo-specificity limitation through the use of *location translation*, an intelligent process in which the locations specifically referenced by a mobile ARG story are transformed to a new area so that any game instance can be played anywhere. Our location translation process is couched in the WEQUEST ARG platform [4] that automates the game master and confederate actor roles. WEQUEST allows ARG stories to reference *specific* geographical landmarks and uses location translation to make the game playable to people in other areas, thus directly resolving geo-specificity limitations.

2 Location Translation

In WEQUEST [4], ARG stories are represented by a *dependency graph*, a directed, acyclic graph (DAG) where the nodes correspond to story events and arcs impose constraints on story event visitation order. Inspired by classic Role-Playing Games (RPGs), story events involve engaging in dialogue with virtual Non-Player Characters (NPCs) and using or acquiring virtual inventory items. Story event nodes reference specific GPS coordinates that a player is required to be within a certain radius of for the interaction to occur. Arcs between nodes represent dependencies that must be fulfilled for a particular event to fire. A dependency graph is a basic technique for managing lock-and-key style game play; for an event to occur, it must be “unlocked” by completing all other events it depends upon. Unlike finite state machines, dependency graphs can easily support branching stories, partial ordering of events, and parallel multiplayer events.

Location translation maps locations in the old game story to analogous locations in a new city where the user intends to play. To formalize the problem, consider an original story set in one area as a number of locations L derived from a dependency graph. For each location L_i in the original story, there can be n_i analogous candidate locations in the vicinity of the target area, denoted $M_{i,j}$ for $j = 1..n_i$. The goal of the translation process is to select one location $M_{i,j}$ for each L_i such that: (a) the analogical similarity between any locations in the original and translated graphs is maximized, irrespective of geography, and (b) the difference in distances between adjacent locations in stories is minimized when geography is considered. These requirements are often in conflict as the most analogically similar locations may not be conveniently located relative to

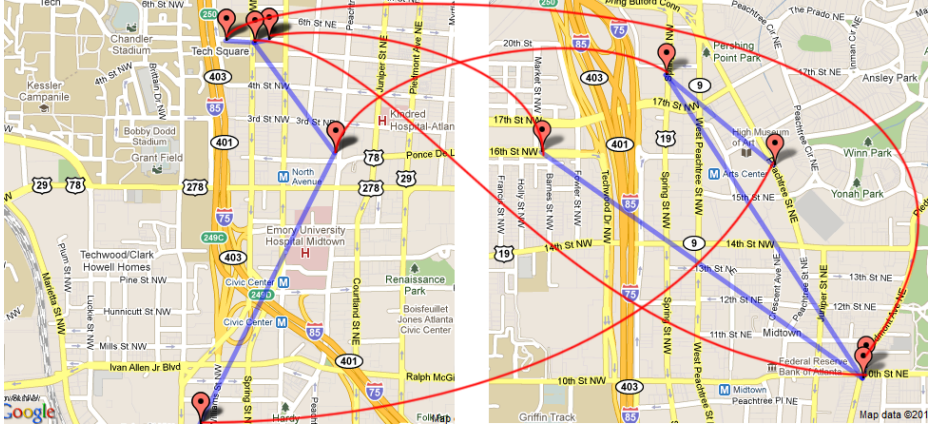


Fig. 1. An ARG story translated from one part of a city to another. Blue lines are dependency arcs. Red lines show analogical matches between locations.

any neighboring locations. Figure 1 shows an example of an ARG story and its translation to a different part of the same city (disallowing self-matching).

2.1 Translation Search Algorithm

Our location translation process searches for the optimal candidate $M_{i,j}$ for each location L_i , given a dependency graph. Viewing game instance translation as an optimization problem, we solve the location translation problem with *dynamic programming*, an optimization algorithm specifically designed to exploit the optimal substructure property through an inductive process that runs in $O(n_{\max} * |L|)$. The solutions to subproblems are cached to avoid repetitious computation. Our dynamic programming implementation determines the suitability of any given candidate $M_{i,j}$ for original location L_i by computing the cost of $M_{i,j}$ given the optimal solutions for locations prior to $M_{i,j}$ in the dependency graph. Because dependency graphs can branch arbitrarily, we extend dynamic programming to account for multiple branching subproblems. See Figure 2.

A cost function evaluates a candidate location $M_{i,j}$ based on similarity of $M_{i,j}$ to the original location L_i plus the difference in distances between the candidate and its dependency graph predecessors as compared to the original dependency graph when locations are positioned geographically. Specifically, $cost(M_{i,j}) = \sum_{d \in Dep(M_{i,j})} (|length(edge_{d,j}) - length(edge_{orig})|) + \frac{k}{sim}$ where $Dep(M_{i,j})$ returns the nodes that candidate $M_{i,j}$ are dependent on according to the dependency graph, $edge_{d,j}$ is a edge in the new graph between the current candidate location and the candidate selected as the solution to a subproblem, $edge_{orig}$ is the corresponding edge in the original dependency graph, and sim is the probability ($[0..1]$) that two locations in two different cities are similar. Thus, as similarity decreases, cost increases exponentially. The constant k is a tunable

```

Input: A list of locations  $L$ , origin and target cities  $city_1, city_2$ , and a set of similarity matrices.
Output: A list of locations  $S$  determined to be analogous to those in  $L$ .
let  $S \leftarrow C \leftarrow M \leftarrow \emptyset$ ;
for  $i = 1$  to number of locations  $|L|$ , consistent with the dependency graph do
  let  $M_i \leftarrow \text{candidates}(L_i, city_1, city_2, \text{type}(L_i))$ ;
  for  $j = 1$  to number of candidates  $|M_i|$  do
    let  $cost \leftarrow 0$ ;
    for  $l = 1$  to number of parents of  $M_{i,j}$  in dependency graph do
       $cost \leftarrow cost + \text{length difference of edge between } M_{i,j} \text{ and } S_l$ ;
     $cost \leftarrow cost + (k/\text{similarity}(L_i, M_{i,j}, city_1, city_2, \text{type}(L_i)))$ ;
    if  $cost < C_i$  then
       $S_i \leftarrow M_{i,j}$ ;
       $C_i \leftarrow cost$ ;

```

Fig. 2. Modified dynamic programming for location translation.

factor that penalizes dissimilarity relative to edge difference. The dynamic programming algorithm selects candidates in the target city that minimize cost.

2.2 Location Similarity

How do we compute the analogical similarity between locations in different cities? Our approach to finding analogies uses statistical correlations based on information about locations retrieved websites such as CitysearchTM and YelpTM that allow their users to write reviews of restaurants, shops, and other landmarks. We make the assumption that the words people use to describe their experiences at these locations captures latent (e.g., hidden) salient features (cf., [2]) of the place and that natural language processing algorithms can analyze and compare word usage to derive similarity between places. When this assumption holds, term-frequency vector similarity techniques can be used to compute the distance between texts—in our case, texts containing user reviews of locations. Our approach to identifying similar locations based on reviews is most similar to the phrase-similarity computation technique of Sahami and Heilman [7], which compares term-frequencies vectors between documents retrieved from GoogleTM. Our technique, however, uses web-retrieved reviews as a document corpus instead of the entire Web Wide Web, and compares locations instead of phrases.

Location translation begins with a pre-processing phase in which a *similarity matrix* is built that captures the probability that locations in disparate cities are analogous. We further specialize the similarity matrices by *type* of location (e.g., restaurant, park, salon, etc.). That is, each similarity matrix represents a combination of $City \times City \times Type$. For each location of each type in each city, we download all reviews from CitysearchTM through their API. Reviews are merged into a single document representing the location. We remove stop-words, words that are not nouns, (according to Wordnet), and common proper nouns (such as the names of credit card companies). Removing non-noun words from reviews avoids relating two places based on similar sentiment. While sentiment analysis is useful for product recommendation, we require an objective account; noun-only similarity is thus a simple form of feature-only comparison under the assumption that nouns identify salient features of a place.

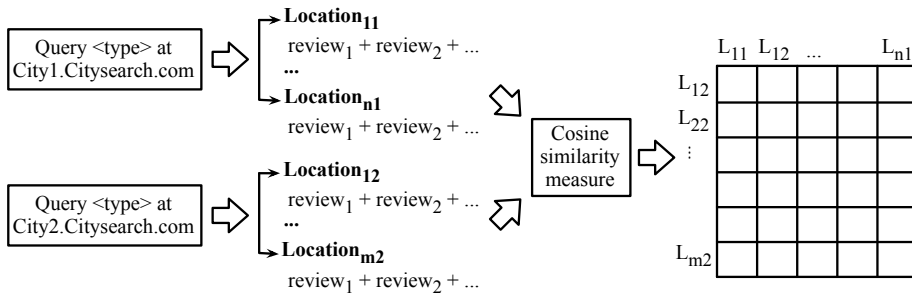


Fig. 3. Similarity matrix construction for a pair of cities and a type of location.

Review documents are converted into term frequency vectors where each dimension in the vector is a term and the value for each term is computed by Term-Frequency Inverse-Document Frequency (TFIDF), a common measure of term importance based on term uniqueness across documents. The Cosine similarity measure is used to determine similarity of document vectors by measuring the angle between each pair of vectors. Applied to all pairs of locations from two cities, the result is a similarity matrix with columns representing places in one city, rows representing places in the other city, and cells containing the probability that the two places are the same. See Figure 3. Repeating this process for all pairs of cities and all types of locations produces $|City \times City \times Type|$ matrices.

Reliable location translation requires reasonably accurate capture of semantic similarities between locations. To evaluate the quality of the similarity computations, we randomly sampled 10 *source* restaurants from our dataset of locations. For each source restaurant, we randomly sampled 10 *target* restaurants against which to evaluate similarity. We asked 5 participants familiar with the city to sort each list of targets based on their judgement of similarity to the source. From participant data, we computed a gold standard ranking as follows. Treating each participant’s trial as a competition amongst target restaurants to be the most similar to the source restaurant, we use the ELO tournament rating method to determine a total order of target restaurants for each source restaurant. The ELO rating for a target restaurant is the aggregate number of other restaurants ordered below it by participants. We then used similarity matrix lookups to generate an ordered list of targets for each source (geography was ignored). Thus, humans and WEQUEST performed the same ranking tasks.

To compare the WEQUEST similarity matrix ranking against the gold standard, we used the Kendall’s Tau rank correlation coefficient to assess the association between ranked lists. We calculated an average τ of 0.533 (where 1.0 indicates perfect agreement) across the 10 source restaurant comparisons, which is significant at $p = 0.0318$ indicating that the gold standard and generated rankings tend to be highly associated. We note that, anecdotally, human participant ranking becomes increasingly arbitrary when actual similarity between locations is low, making the gold standard ELO values for low-similarity restau-

rants unreliable. Thus, looking at the tops of the rankings, WEQUEST’s top pick concurred with the gold standard’s top-pick 60% of the time, was in the top two 80% of the time, and was in the top three 100% of the time. Thus, accuracy is highest when human-rated similarity is also high, which is significant considering that the optimization search must balance maximizing similarity with minimizing distances; it doesn’t always pick the most similar location.

3 Conclusions

There is a lot of semantic knowledge to be harnessed from human-generated natural language that can be brought to bear for the purposes of automating creative tasks. Location translation is a form of analogical reasoning where we promote game play by computing probabilistic similarities between locations in different cities based on the words that people use to describe those places on the World Wide Web. In WEQUEST we have applied location translation to automatically re-author ARG stories, making them playable anywhere, regardless of geo-specific references to locations. This allows WEQUEST to overcome a significant limitation to the adoption of ARGs as a mainstream form of interactive entertainment: content geo-specificity. The ability to translate ARG stories from one area to another, combined with end-user story authoring [4], has the potential to scale up the amount of content available to players, making ARGs more accessible to mainstream audiences who desire real-world gaming experiences.

References

1. Gustafsson, A., Bichard, J., Brunnberg, L., Juhlin, O., Combetto, M.: Believable environments: Generating interactive storytelling in vast location-based pervasive games. In: Proceedings of the 2006 ACM International Conference on Advances in Computer Entertainment (2006)
2. Landauer, T., Dumais, S.: A solution to Plato’s problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review* 104, 211–240 (1997)
3. Lim, M.Y., Aylett, R.: Narrative construction in a mobile tour guide. In: Proceedings of the 4th International Conference on Virtual Storytelling (2007)
4. MacVean, A., Hajarnis, S., Headrick, B., Ferguson, A., Barve, C., Karnik, D., Riedl, M.O.: WeQuest: Scalable alternate reality games through end-user content authoring. In: Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology (2011)
5. Montola, M.: Exploring the edge of the magic circle: Defining pervasive games. In: Proceedings of the 2005 Digital Arts and Culture Conference (2005)
6. Reed, A., Samuel, B., Sullivan, A., Grant, R., Grow, A., Lazaro, J., Mahal, J., Kurniawan, S., Walker, M., Wardrip-Fruin, N.: A step towards the future of role-playing games: The SpyFeet mobile RPG project. In: Proceedings of the 7th Annual Conference on Artificial Intelligence and Interactive Digital Entertainment (2011)
7. Sahami, M., Heilman, T.: A web-based kernel function for measuring the similarity of short text snippets. In: Proceedings of the 15th International World Wide Web Conference (2006)