Multi-Modal Question-Answering: Questions without Keyboards

Gary Kacmarcik
Natural Language Processing Group
Microsoft Research
`garykac@microsoft.com`

Abstract

This paper describes our work to allow players in a virtual world to pose questions without relying on textual input. Our approach is to create enhanced virtual photographs by annotating them with semantic information from the 3D environment's scene graph. The player can then use these annotated photos to interact with inhabitants of the world through automatically generated queries that are guaranteed to be relevant, grammatical and unambiguous. While the range of queries is more limited than a text input system would permit, in the gaming environment that we are exploring these limitations are offset by the practical concerns that make text input inappropriate.

1 Introduction

The question-posing part of Question-Answering (QA) has long relied on the cooperative nature of the person posing the question. This assumption is not unreasonable because it generally behooves the querent to assist the QA system wherever possible.

However, even given this cooperative nature, QA systems that rely on text input still have to deal with input that is malformed, underspecified or problematic in some other way. This problem is further compounded when the system is open to users who may find it more entertaining to explore the boundaries, limitations and humorous errors of the text input and parsing system instead of using the system as intended. Thus, any system that is intended to be released to a wide audience needs to be designed to handle these problems in a robust manner.

Additionally, there are applications where QA technologies would be beneficial, but the reliance on text input renders them impractical. The focus of the present work, interactive virtual game worlds, is one such area where text input is not desirable — both because it interrupts the game flow and because many game systems do not have keyboards available.

In this paper, we explore one method of creating a non-text input mode for QA that relies on specially annotated virtual photographs.

Our approach is to create a virtual game world where all of the objects (and some non-objects) are annotated with semantic information that is constructed automatically by parsing natural language text descriptions. By interacting with world objects, a player is actually selecting portions of the semantic network that can in turn be used to enable a limited QA dialog with denizens of the game world. While this method is clearly not as flexible as full natural language input, it successfully avoids most of the serious natural language input problems in much the same way that Tennent et al. (1983) avoided the ambiguity, paraphrase and incorrect input problems in their NLMENU system. In addition, our system does so without the awkwardness of forcing players to build utterances word-by-word from a series of menus.

In our system, players interact with *non-player characters* (NPCs: characters in the world whose actions are controlled by the computer) by taking virtual photographs of objects in the world that they want to discuss and then showing the photos to the NPCs. When the photo is taken, all of the relevant semantic information from the world is attached to the photo so that it acts as a standalone object – even if the game world changes, the contents of the photo are still valid and consistent. By combining these photo annotations with information in the NPC's knowledgebase (KB), we can create the illusion that the NPC has a rudimentary understanding of the photo contents and create a novel interaction modality that gives the player a wide range of expression.

It is worth noting that while we discuss using these annotations in the context of a virtual photo, the annotations can also be applied in realtime interactive systems. In this work, we restrict ourselves to the use of virtual photos primarily because it allows us to interact with a static scene, thus eliminating the temporal difficulties (graphical and linguistic) that would be caused by interacting with a dynamic dataset.

2 Previous Work

A wide variety of work has been done on integrating graphics and/or virtual environments with natural language dating back to Winograd's (1972) classic "blockworld" simulation. More recently, researchers have been investigating how graphics and natural language can work together to create more compelling interfaces.

2.1 Multimodal Interfaces

A large body of work has been created on multimodal interfaces – combining multiple modes of interaction so that the advantages of one mode offset the limitations of another. In the specific case of combining natural language and graphics, there have been two main areas of study: interacting with graphical elements to resolve ambiguous references on the natural language side (Bolt, 1980; Kobsa et al., 1986); and generating coordinated text and graphic presentations using information from a knowledgebase (André and Rist (1994); Towns et al. (1998)).

In addition to these two main areas, early work by Tennant (1983) experimented with using a predictive left-corner parser to populate dynamic menus that the user would navigate to construct queries that were guaranteed to be correct and task-relevant.

Our work contains elements from all of these categories in that we use input gestures to resolve reference ambiguity and we make use of a KB to coordinate the linguistic and graphical information. We were also inspired by Tennant's work on restricting the player's input to avoid parsing problems. However, our work differs from previous efforts in that we:

- Do not use text input at runtime
- Use virtual cameras and input gestures for interaction
- Do not require that interactions be built one unit (word or graphical references) at a time
- Focus primarily on text generation

2.2 Virtual Photographs

The concept of a virtual photograph has existed as long as people have taken screenshots of their view into a 3D environment. Recently, however, there have been a few applications that have experimented with adding a limited amount of interactivity to these static images. Video games, notably POKÉMON SNAP (Nintendo, 1999), incorporate a limited form of interactive virtual photos. While there is no published information about the techniques used in these games, we can infer much by examining the level of interaction permitted.

In POKÉMON SNAP, the player zooms around each level on a rail car taking as many photographs of "wild" *pokémon* as possible. Scoring in the game is based not only on the number of unique subjects found (and successfully photographed), but also on the quality of the individual photographs. The judging criteria include:

- Is the subject centered in the photo?
- Is the face visible? (*for identifiability*)
- Does the subject occupy a large percentage of the image?
- Are there multiple *pokémon* (same type)?
- What is the subject doing? (*pose*)

In order to properly evaluate the photos, the game must perform some photo annotation when the photo is taken. However, since interaction with the photo is limited to scoring and display, these annotations are easily reduced to the set of values necessary to calculate the score. From the players' perspective, since there is no mechanism for interacting with the contents of the photo, all interaction is completed by the time the photo is taken - the photo merely serves as an additional game object.

2.3 Interactive Images

Recently, a lot of work has gone on in the field of making images (including electronic versions of real photographs) more interactive by manually or automatically annotating image contents or by making use of existing image metadata. The most commonly used example of this are the HTML image maps (Berners-Lee and Connolly, 1995) supported by most web browsers.

An example that is more relevant to our work is the ALFRESCO system (Stock, 1991), which uses graphical representations of Italian frescos and allows the user to query using a combination of natural language and pointing gestures. Beyond the obvious difference that our system doesn't permit direct natural language input, our work also differs in that we annotate the images with scene information beyond a simple object ID and we calculate the image regions automatically from the objects in the virtual world.

3 Interacting with Virtual Photos

As mentioned in the Introduction, virtual photos can become a useful metaphor for interaction with NPCs in games. Ideally, the player should be able to take a picture of anything in the virtual world and then show that photo to an NPC to engage in a dialog about the photo contents.

In our implementation, the player interacts with the NPC by clicking on an object in the photo to pull up a menu of context-dependent natural language queries. When the player selects an item from this menu, the query is sent to the NPC that the player is currently "talking to". This menu of context sensitive queries is crucial to the interaction because a pointing gesture without an accompanying description is ambiguous (Schmauks, 1987) and it is through this menu selection that the player expresses intent and restricts the scope of the dialog.

There are two obvious benefits to approaching the QA interaction in this way. First, even though the topic is limited by the objects in the photo, the player is given control over the direction of the dialog. This is an improvement over the traditional scripted NPC interaction where the player has little control over the dialog. The other benefit is that while the player is given control over the content, the player is not granted too much control since the photo metaphor limits the topic to things that are relevant to the game. This effectively avoids the out-of-domain, paraphrase and ambiguity problems that commonly plague natural language interfaces.

3.1 Annotations

The quality of player-NPC interaction is directly dependent on the kind of annotations that are used. For example, associating a literal text string with each object would result in a system where the NPCs would not exhibit individuality since they would all produce the exact same answer to a query. Alternately, using a global object identifier would also cause problems because in a dynamically changing world we would need to create a system to keep track of differences from object at the time of the photo and the object's current state.

It is for these reasons that we record for each object an abstract representation that we can manipulate and merge with data from other sources like the NPC's KB. Beyond providing a place to record information about the objects that are specific to a particular photo, this also allows us to individualize the NPC responses and create a more interesting QA interaction.

3.2 Example Interaction

As a simple example, imagine a photo taken by a player that shows a few houses in a town. Taking this photo to an NPC and clicking on one of the houses will bring up a menu of possible questions that is determined by the object and the contents of the NPC's KB. Selecting the default "What is this?" query for an NPC that has no special knowledge of the objects in this photo will result in the generic description (stored in the photo) being used for the NPC's response (e.g., "That is a blue house").

If, however, the NPC has some knowledge about the object, then the NPC will be able to provide information beyond that provided within the photo. Given the following information:

This is John's house.
My name for John is my father.

the NPC can piece it all together and generate "That is my father's house" as an answer.

4 Representing Knowledge

A key component of our system is the semantic representation that is used to encode not only the information that the NPC has about the surroundings, but also to encode the contents of the virtual photo. These KBs, which are created from text documents containing natural language descriptions, form the core document set on which the QA process operates.

4.1 Semantic Representation

While there are a variety of representations that can be used to encode semantic information, we opted to use a representation that is automatically extracted from natural language text. We chose this representation because we desired a notation that:

- Is easy to create
- Provides broad coverage over structures found in natural language
- Is easy to manipulate, and
- Is easy to convert into text for display

Because of these requirements, we use a predicate-argument style representation (Campbell and Suzuki, 2002) that is produced by our parser. These structures, called logical forms (LFs), are the forms that are stored in the KB.

This tree structure has many advantages. First, since it is based on our broad coverage grammar it provides a reasonable representation for all of the things that a player or NPC is likely to want to talk about in a game. We also are readily able to generate output text from this representation by making use of our generation component. In addition, the fact that this representation is created directly from natural language input means that game designers can create these KBs without any special training in knowledge representation.

Another advantage of this tree structure is that it is easy to manipulate by copying subtrees from one tree into another. Passing this manipulated tree to our generation component results in the text output that is presented to the user. The ease with which we can manipulate these structures allows us to dynamically create new trees and provide the NPC with the ability to talk about a wide array of subjects without having to author all of the interactions.

4.2 Anaphora

As mentioned, once these sentences for the KB have been authored, our parser automatically handles the work required to create the LFs from the text. However, we do not have a fully automatic solution for the issue of reference resolution or anaphora. For this, we currently rely on the person creating the KB to resolve references to objects within the text or KB (*endophora*) and in the virtual world (*exophora*).

5 Posing Questions

In our system questions are posed by first narrowing down the scope of the query by selecting an object in a virtual photo, and then choosing a query from a list that is automatically produced by the QA system. This architecture places a heavy burden on the query generation component since that is the component that determines the ultimate limitations of the system.

5.1 Query Generation

In a system where the only automatically generated queries are allowed, it is important to be able to create a set of interesting queries to avoid frustrating the user. Beyond the straightforward "Who/What/Where is this?"-style of questions, we also use a question generator (originally described by Schwartz et al. (2004) in the context of language learning) to produce a set of answerable questions about the selected object.

Once the player selects a query, the final step in query generation is to create the LF representation of the question. This is required so that we can more easily find matches in the KB. Fortunately, because the queries are either formulaic (e.g., the "Who/What/Where" queries), or extracted from the KB, the LF is trivially created with requiring a runtime parsing system.

5.2 Knowledgebase Matching

When the player poses a query to an NPC, we need to find an appropriate match in the KB. To do this, we perform subtree matches between the query's LF and the contents of the KB, after first modifying the original query so that question words (e.g., Who, What, ...) are replaced with special identifiers that permit wildcard matches. When a match is found, a complete, grammatical response is created by replacing the wildcard node with the matching subtree and then passing this structure to the text generation component.

5.3 Deixis

In order to make the NPC's responses believable, the final step is to incorporate deictic references into the utterance. These are references that depend on the extralinguistic context, such as the identity, time or location of the speaker or listener. Because the semantic structures are easy to manipulate, we can easily replace these references with the appropriate reference. An example of this was given earlier when the subtree corresponding to "my father" was used to refer to the owner of the house.

This capability gives us a convenient way to support having separate KBs for shared knowledge and individual knowledge. General information can be placed in the shared KB, while knowledge that is specific to an individual (like the fact that John is "my father") is stored in a separate KB that is specific to that individual. This allows us to avoid having to re-author the knowledge for each NPC while still allowing individualized responses.

6 Creating Annotated Photographs

Our virtual photos consist of three major parts: the image, the object locator map and the object descriptors. In addition, we define some simple metadata. We use the term "annotations" to refer to the combination of the object locator map, the descriptors and the metadata.

While the photo image is trivially created by recording the camera view when the photo is taken, the other parts require special techniques and are described in the following sections.

6.1 The Object Locator Map (OLM)

The object locator map (OLM) is an image-space map that corresponds 1-to-1 with the pixels in the virtual photograph image. For each image pixel, the corresponding OLM "pixel" contains information about the object that corresponds to that image-space location. We create the OLM using the back buffer technique attributed originally to Weghorst et al. (1984).

6.2 The Object Descriptors

The object descriptors contain the semantic description of the objects plus some metadata that helps determine how the player and NPC can interact with the objects in the photo.

In our system, we use the semantic annotations associated with each object as a generic description that contains information that would be readily apparent to someone looking at the object. Thus, these descriptions focus on the physical characteristics (derived from the object description) or current actions (derived from the current animation state) of the object.

7 3D Modeling

The modeling of 3D scenes and objects has typically been done in isolation, where only graphical (display and performance) concerns were considered. In this section, we discuss some of the changes that are required on the modeling side to better support our interaction.

7.1 Enhancements

Beyond the enhancement of attaching abstract semantic descriptions (rather than simple text labels as in Feiner et al. (1992)) to each object in the virtual world's scene graph, we introduce a few other features to enhance the interactivity of the virtual photos.

Semantic Anchors

A limitation of attaching the semantic descriptions to objects in the 3D world is that this only covers concrete objects that have a physical representation in the world. Semi-abstract objects (called "negative parts" by Landau and Jackendoff (1993)) like a cave or a hole do not have a direct representation in the world and thus do not have objects onto which semantic descriptions can be attached. However, it is certainly possible that the player might wish to refer to these objects in the course of a game.

We provide support for these referable, non-physical objects through the use of *semantic anchors*, which are invisible objects in the world that provide anchor points onto which we can attach information. For example, abstract objects like a hole or a cave can be filled with a semantic anchor so that when a photo is taken of a region that includes the cave, the player can click on that region and get a meaningful result.

Since these objects are not displayed, there is no requirement that they be closed 3D forms. This gives us the flexibility to create *view-dependent* semantic anchors by tagging regions of space based on the current view. For example, a cave entrance could be labeled simply as a "cave" for viewpoints outside the cave while this same portal can be termed an "exit" (or left unlabeled) from vantage points inside the cave. By orienting these open forms correctly, we can rely on the graphic engine's backface culling[1] to automatically remove the anchors that are inap-

[1] *Backface culling* is an optimization technique that removes back-facing surfaces (i.e., surfaces on the side of the object away from the viewer) from the graphic engine pipeline so that resources are not wasted processing them. This technique relies on the assumption that all the objects in the virtual world are closed 3D forms so that drawing only the front-facing surfaces doesn't change the resulting image.

propriate for the current view.

Action Descriptions

In addition to attaching semantic descriptions to objects, we also allow semantic descriptions be added to animation sequences in the game. This provides a convenient mechanism for identifying what a person is doing in a photo so that questions relating to action can be proposed.

Key Features

We also permit key features to be defined (as was apparently done for POKÉMON SNAP) so that we can approximate object identifiability. In our implementation, we require that (at least a portion of) all key features are visible to satisfy this requirement.

The advantage of this approach is that it is easy to implement (since there's no need to determine if the entire key feature is visible), but it requires that the key features be chosen carefully in order to produce reasonable results.

7.2 Limitations

Even with the proposed enhancements, there are clear limitations to the annotated 3D model approach that will require further investigation.

First, there is an unfortunate disconnect between the modeled structures and semantic structures. When a designer creates a 3D model, the only consideration is the graphical presentation of the model and so joints like a wrist or elbow are likely to be modeled as a single point. This contrasts with a more semantic representation, which would have the wrist extend slightly into the hand and forearm.

Another problem is the creation of relationships between the objects in the photo. This is difficult because many relationships (like "next to" or "behind") can mean different things in world-space (as they are in the virtual world) and image-space (as they appear in the photo).

And finally, there is the standard "picking from an object hierarchy" problem where, when a node in the hierarchy is selected, the user's intent is ambiguous since the intended item could be the node or any of its parent nodes.

8 Conclusions

In this paper we have described our approach to allowing users to specify queries by interacting with virtual photographs that have been annotated with semantic information. While this approach is clearly more limited than allowing full text input, it is useful for applications like games that do not always have a keyboard available.

References

E. André and T. Rist. 1994. "Referring to World Objects with Text and Pictures". In *Proceedings of COLING 1994*. 530-534.

T. Berners-Lee and D. Connolly. 1995. "HyperText Markup Language Spec. - 2.0". W3C RFC1866.

R.A. Bolt. 1980. "'Put-that-there': Voice and Gesture at the Graphics Interface". *SIGGRAPH 80*. ACM Press. 262-270.

R. Campbell and H. Suzuki. 2002. "Language Neutral Representation of Syntactic Structure". In *Proceedings of SCANALU 2002*.

S. Feiner. B. MacIntyre and D. Seligmann. 1992. "Annotating the real world with knowledge-based graphics on a see-through head-mounted display". *Graphics Interface 92*. 78-85.

A. Kobsa, J. Allgayer, C. Reddig, N. Reithinger, D. Schmauks, K. Harbusch and W. Wahlster. 1986. "Combining Deictic Gestures and Natural Language for Referent Identification". In *Proceedings of COLING 1986*. 356-361.

B. Landau and R. Jackendoff. 1993. "'What' and 'Where' in spatial language and spatial cognition". *Behavioral and Brain Sciences*, 16, 217-265.

D. Schmauks. 1987. "Natural and Simulated Pointing". In *Proceedings of the 3rd European ACL Conference*, Copenhagen. 179-185.

L. Schwartz, T. Aikawa, and M. Pahud, "Dynamic Language Learning Tools", in *InSTIL/ICALL Symp. 2004: NLP and Speech Technologies in Adv. Lang. Learning*, Venice, Italy, June 2004.

O. Stock. 1991. "AlFresco: Enjoying the Combination of NLP and Hypermedia for Information Exploration", In *AAAI Workshop on Intelligent Multimedia Interfaces* 1991. 197-224.

H. Tennant, K. Ross, R. Saenz, C. Thompson and J. Miller. 1983. "Menu-based natural language understanding". In *Proc. of ACL '83*. 151-158.

S. Towns, C. Callaway and J. Lester. 1998. "Generating Coordinated Natural Language and 3D Animations for Complex Spatial Explanations". In *Proc. of the 15th National Conf. on AI*. 112-119.

T. Winograd. 1972. *Understanding Natural Language*. Academic Press. 1972.

H. Weghorst, G. Greenberg and D. Greenberg. 1984. "Improved Computational Methods for Ray Tracing". *ACM Transactions on Graphics* 3, 1, 52-69.