Painting Pictures with Words – From Theory to System

Robert Coyne

Submitted in partial fulfillment of the
requirements for the degree
of Doctor of Philosophy
in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2017

ABSTRACT

Painting Pictures with Words – From Theory to System

Robert Coyne

A picture paints a thousand words, or so we are told. But how many words does it take to paint a picture? And how can words create pictures in the first place? In this thesis we examine a new theory of linguistic meaning – where the meaning of words and sentences is determined by the scenes they evoke. We describe how descriptive text is parsed and semantically interpreted and how the semantic interpretation is then depicted as a rendered 3D scene. In doing so, we describe WordsEye, our text-to-scene system, and touch upon many fascinating issues of lexical semantics, knowledge representation, and what we call "graphical semantics." We introduce the notion of *vignettes* as a way to bridge between function and form, between the semantics of language and the grounded semantics of 3D scenes. And we describe how VigNet, our lexical semantic and graphical knowledge base, mediates the whole process.

In the second part of this thesis, we describe four different ways WordsEye has been tested. We first discuss an evaluation of the system in an educational environment where WordsEye was shown to significantly improve literacy skills for sixth grade students versus a control group. We then compare WordsEye with Google Image Search on "realistic" and "imaginative" sentences in order to evaluate its performance on a sentence-by-sentence level and test its potential as a way to augment existing image search tools. Thirdly, we describe what we have learned in testing WordsEye as an online 3D authoring system where it has attracted 20,000 real-world users who have performed almost one million scene depictions. Finally, we describe tests of WordsEye as an elicitation tool for field linguists studying endangered languages. We then sum up by presenting a roadmap for enhancing the capabilities of the system and identifying key opportunities and issues to be addressed.

Table of Contents

List of Figures

List of Tables

Acknowledgments

I thank Julia Hirschberg, my faculty advisor, for all the support, guidance, and inspiration she has given over the years. It has been an exciting adventure! I especially thank Julia for introducing me to Richard Sproat with whom I collaborated on the initial versions of WordsEye. Most of the language processing in those versions (some of which still is being used) was developed by Richard, and I learned a tremendous amount working with him. I thank Owen Rambow and Daniel Bauer for the many stimulating conversations on meaning representation and grounded semantics and for the work we did together, much of which is reflected in the ideas presented in this thesis. I thank Morgan Ulinski for her help with the Mechanical Turk evaluations and for bringing a completely different dimension to text-to-scene technology in the area of computational field linguistics. I thank Larry Stead for suggesting that I go back to get a PhD and innumberable acts of friendship over the years. And I thank Fadi Biadsy for his friendship, enthusiasm, and the warm welcome he gave when I first arrived at Columbia.

Many people have contributed to the WordsEye system itself: In addition to Daniel, Larry, and Richard, I thank Gary Zamchick, Ben Kuykendahl, Phil Chu, Phil Zucco, and Michael Fleming – it has truly been a collaborative effort. I thank the following Columbia students for their help on different aspects of the project: Cecilia Schudel, Alex Klapheke, Mi Zhou, Mandy Korpusik, Sam Wiseman, and Anusha Balakrishnan. I thank Michael Bitz for his key role in our education evaluation and the many WordsEye users who have created scenes on the system, some of which appear in this thesis.

I would like to thank Steve Feiner for bringing to my attention several references to early text-to-graphics systems. Also, for the historical record, the term “text-to-scene” was coined by Richard Sproat, following the example of “text-to-speech”; and the compound name “WordsEye” was suggested by Julia Hirschberg as an improvement on the author’s earlier “Words Eye.”

Lastly, I thank my wife Linda for her support through all this and our two kids Theodore and Christopher who I hope find something as interesting to devote themselves to as I have found.

Frontispiece

S E M A N T I C L I G H T

The elephant is in the middle of the matisse field. A yellow light and a green light and a blue light are in front of it. The five lights are above the elephant. The five yellow lights are above the booth. The field is shiny. The booth is three feet left of the elephant. A clown fits inside the booth. A very enormous magenta glass cylinder is 4 feet behind the booth. The sun is coral.

Chapter 1

Introduction

1.1 Motivation and Approach

The research described in this thesis is motivated by a single application idea – how to automatically depict English textual descriptions.

Our premise is that pictures, like language itself, are an encoded form of meaning. When we see a person with their hand outstretched and a welcoming expression on their face, we interpret it as a *greeting*. When we see an apple on the top of the curved lower surface of a bowl, we interpret it as being *in* the bowl. So our task, in the most general terms, becomes one of translating one form of meaning into another.

There is something appealing about the notion that our mental models of textual meaning are grounded in visual representations. The intuition that language and pictures are intimately related takes different forms: from popular culture where in the Holodeck on Star Trek, voice commands can conjure up holograph experiences; to academic philosophy, where Wittgenstein's picture theory of language introduced the idea that all facts about the world can be reduced to pictures; to everyday life, where children's books, graphic novels, and internet memes rely on text and pictures being situated side by side to mutually reinforce one another.

In this thesis we examine the conditions under which text can be automatically converted to 3D scenes, where those scenes represent the meaning of the text. The first thing that becomes apparent is that the scene is usually more concrete than the text, since the objects and the spatial relations they have to each other are explicit. We note here however, and discuss later, that some *vagueness* and *ambiguity* does occur in pictures. This can occur because of object selection and placement, lighting effects, or camera placement. Text that

is concrete and apparently simple can result in an abstract design or be otherwise open to interpretation. Examples of this are shown in Figure 3.1.

Our goal is not only to extend prior work on text-to-scene generation but also to develop a new theory of meaning that encompasses lexical semantics and *graphical semantics* in a unified and extensible framework. In our theory, meaning is not only embodied in high-level functional and logical constraints but is ultimately reified as configurations of graphical objects. We represent all meaning (lexical, functional, graphical, world knowledge) by concepts linked together by semantic relations, and these relations are then graphically grounded by semantically-charged configurations of objects that we call *vignettes*. All levels of semantics are represented, translated, and decomposed in a uniform manner, where, for example, graphical knowledge can be used to interpret lexical semantics constructs. We show how our theory is computationally realized in a state-of-the-art text-to-scene system (WordsEye) for constructing scenes from linguistic descriptions.

The work in this thesis is presented as follows: We begin by examining the overall task of text-to-scene generation, asking what capabilities are needed at the linguistic, semantic, graphical, and representational level. We then establish a semantic theory and framework for representating the lexical and real-world knowledge that are needed to perform the translation from text to scenes. With the framework established, we describe in detail the computational mechanisms by which WordsEye, our text-to-scene system, is implemented to address both lexical and graphical semantic issues. Since text-to-scene generation, in the abstract, is a very open-ended and difficult task, we then examine some specific concrete real-world applications (authoring tool, educational tool, visual search-as-generation, and field linguistics) and evaluate WordsEye in those areas. And we conclude by summarizing with what we believe are the main contributions of the work and provide some key directions for the future.

In doing this work, we are motivated and guided by the following insights and principles:

- The meaning of sentences is largely compositional. Depicting them requires taking the meaning apart piece by piece in order to construct the full graphical equivalent.
- Graphical objects (and their real-world analogs) contain information required to understand language, especially spatial language. In particular, the functions and *affordances* of objects determine how they are used and configured in space. And this affects the words used to describe those configurations.
- There are many ways to say the same thing. So part of our task is representing and translating between overlapping meanings.

• In depicting actions, we are unconcerned with literally capturing temporal relations as animation. Instead, we view actions "comic book style" as frozen in time, where any given action is suggested by a configuration of objects and poses. The action in some sense *becomes* a concrete object, where the parts of that object signify roles in the original action. We call these configurations *vignettes*. Those configurations bridge between form and function and represent more than the sum of their parts. That added meaning is what the vignette captures.

• There is regularity to the visual structure of actions that we formalize with an inventory of reusable abstract vignettes (e.g. using a work surface).

• In modeling *graphical meaning*, we reduce the complexity of the real world with abstracted forms of objects and their properties and affordances.

• All scenes can be described with a set of graphical primitives.

• In representing knowledge about words and the world, we build bottom-up as well as top-down since the actual 3D objects used in the system are represented in the same ontology as general types and concepts. This allows a more grounded form of representing language and semantics.

• In translating meaning, we access an ontology and knowledge base using semantic pattern matching rules. This happens at both the purely lexical semantic level (*x is heavy* ⇒ weight(x, "high value")) and at the semantic decomposition level (being *on* a wall is different than being *on* a table).

• Our focus is not so much based on any particular technique but rather in the way components are fit together into a coherent whole. Representational, graphical and lexical semantic, and computational flow issues are the heart of our endeavor. We are less interested, at the current time, about the role machine learning might have. Our goal is framing the approach and defining the models rather than the fine-tuning of parameters.

1.1.1 Framework

In mapping language to pictures, we need to consider both *low-level* language that directly or indirectly expresses graphical constraints and *high-level* language that describes not what a scene looks like but what event or state-of-affairs is represented by the scene. To capture the relationship between these levels, we

collected a corpus of semantically grounded descriptive language where Mechanical Turkers annotated 100 pictures at both levels. The difference in language is shown is Figure 1.1.

Low: *A man is using the telephone. The man is wearing a yellow vest. The man has blonde hair. The man has white skin. A white rodent is inside a cage. The cage is on a table. The phone is on the table. The cage has a handle. A safe is in the background of the room.*
High 1: *The man is a scientist working with white rodents.*
High 2: *The man is talking to another scientist.*
High 3: *The man feels guilt at imprisoning a white rodent.*

Figure 1.1: High-low annotation done by Mechanical Turkers

In order to represent meaning, for both high and low level semantics, we define a representational framework where lexical items are associated with *concepts* to represent entities and *semantic relations* to represent relations between those entities. We also define *translation rules* that are used to translate from one semantic relation to another, ultimately to a set of graphically oriented primitives. On top of this framework, we introduce the concept of a *vignette*, which captures the various canonical ways of grounding a given relation in the context of its arguments. For example *cutting the lawn* involves pushing a lawnmower along the ground, while *cutting a carrot* involves holding an instrument (knife) and applying it to another object (carrot) on a *work-surface*. Vignettes act as the bridge between high-level and low-level (grounded) semantic relations.

One main theme of this thesis is that spatial and functional features of *objects* can be used to interpret the semantics and senses of *words*. Thus we focus on defining a representational system and building a knowledge base and to encode and utilize those differences. Our model of grounded semantics depends on physical and functional features of real-world objects for which we adopt the term *affordances*. (see Section 3.7). Affordances constrain how objects and human characters interact with one another in the world. For example, a *handle* on a suitcase is an affordance that allows us to hold it. We describe the graphical knowledge (spatial regions and other properties) underlying affordances as well as how affordances and other knowledge are used to decompose language into graphics. In addition, we use object features to analyze implicitly expressed relations (such as noun-noun compounds) and how these can be made explicit – a necessary step in order to depict them. For example, *stone table* can be interpreted as an implicit MATERIAL-OF relation and *horse head* as an implicit PART-OF relation.

To represent the knowledge needed to depict text as 3D scenes, we have developed a new unified lexical resource, ontology, and knowledge base, called *VigNet*. We have been populating this resource using crowdsourcing, text and resource mining, and custom software annotation tools.

1.1.2 Graphical Limitations

In attempting to create scenes from text, we must consider what type of language can feasibly be depicted. Some sentences immediately bring images to mind, while others would leave even the most visually oriented reader in the dark. Other sentences (such as "the devil is in the details") are not directly visual but can be "literalized" (e.g., a devil figure standing amidst the word "details") and depicted as such. In addition, some restrictions on the range of input text we handle are due to what is graphically feasible within the scope of this project. There are also some aspects of the system (poses and facial expressions and their associated linguistic correlates) which we have partially but not yet fully implemented. A partial list of graphical restrictions is given below:

- **Static scenes:** We will create static scenes, not animations. The input text will, hence, be limited to describing or implying a single location and point in time. Static scenes will not be able to capture the full meaning of actions and events. However, action, movement, and state change can often be suggested by putting a character in a characteristic pose or objects in a certain configuration. Furthermore, we only focus on a single time-slice of an action, leaving for future work issues involved with representing time as sequential panels, such as comics or other sequential art [McCloud, 1993].

- **No large aggregates:** We will not attempt to depict large groups of objects denoted by words such as *crowd*, *swarm*, and *clutter*. We will handle some simple smaller-scale aggregates such as *stack* and *row*.

- **Human-sized scale:** All objects, other than the background setting, will be constrained to be human scale. This will avoid issues with having to depict scenes with galaxy-sized objects next to microscopic ones.

- **No 3D deformations or topological changes of objects:** We will not attempt to modify the shape of objects (other than to apply affine transformations such as scaling). This a) prevents most changes to an object's parts (e.g. *the giraffe's neck is curved to the right.*) and b) restricts the ways that soft,

flexible objects, such as rope or cloth, can be manipulated and c) prevents object states from being changed (e.g. opening an umbrella or a book) and d) means that fluids, such as water or steam, can only be depicted as rigid 3D objects. Also, topological changes are not supported such as punching holes in objects or slicing or fracturing objects.

- **Limited poses and facial expressions:** While conceptually related to deformable shapes, we have experimented with poses and facial expressions. We present examples of some work in those areas but we have much yet to do, and the current system is oriented towards rigid shapes.

- **Limited support for object parts:**. It is very common for textual descriptions to reference object parts (e.g. *the spine of the book*). We attempt to handle part references (Section 5.6.2), but the support is limited in several ways. The primary problem is that the polygonal meshes used to represent objects do not necessarily segment the mesh into separate groups designating all the different parts of the object. Instead, parts are frequently only differentiated when they require different materials or other surface properties.

When there are graphical restrictions, we can sometimes attempt to interpret the meaning of the language, but the translation to graphics will not be done. When the system is being used interactively, the user is notified when a limitation of this type is detected.

Despite these and other limitations, there is a tremendous range of scenes that can be depicted. We have built the framework to accommodate new capabilities as they become available. As we describe the system and its underlying lexical, semantic, and graphical framework, we will point out the current limitations and plans for future work.

1.2 WordsEye Overview

The work in this thesis is centered around WordsEye, an open-domain text-to-scene generation system. The current version of WordsEye is a continuation of work described in [Coyne and Sproat, 2001] which was the first text-to-system to offer relatively flexible language input in combination with a large, wide-ranging library of (approximately 2,000) 3D objects. It supported language-based control of object selection, spatial relations, surface properties (such as textures and colors), poses, and a few simple facial expressions. It handled a relatively wide range of common spatial linguistic constructions as well as simple anaphor

resolution, allowing for a variety of ways of referring to objects. This version handled 200 verbs in an *ad hoc* manner with no systematic semantic modeling of semantic interpretation or decomposition to graphical primitives.

In the current version of WordsEye [Coyne *et al.*, 2010a], we have reformulated the entire system around a new lexically and semantically grounded knowledge resource that we call VigNet. The knowledge represented in VigNet provides the necessary foundation for mapping language into semantics, interpreting the semantics in terms of graphical constraints, and translating those constraints into an actual rendered 3D scene. The system runs in the cloud and supports an online website, web API, and a mobile app. It utilizes state-of-the arg GPUs for real-time raytraced rendering and has a significantly larger object library of 3,500 objects, including a few hundred "cutouts" (2D images with mask channels allowing them to act as stand-ins for real 3D objects when viewed from straight ahead). In this new system we have also focused on exploring the online social media aspects of the technology. Over a ten month period, 20,000 online users have used it to create over 25,000 rendered scenes. The website supports an online gallery with 6,700 user scenes posted and commenting, which fosters the use of textually generated pictures in a social manner as a form of "visual banter". This online system is described in Chapter 9. See Appendix C for sample pictures created by users of the system.

1.3 Background – Systems for Generating Graphics with Natural Language

Natural language text has been used as input to computer graphics in a number of research systems developed over the years. These systems vary in the form and content of the linguistic input they handle as well as the type of output they produce.

- *Time*: Some systems produce animation, while others create static scenes, and others segment time into separate static scenes in order to create picture panels or storyboards.

- *Scope and type of graphical content*: Some early systems only support simple 3D rectangular solids. Some systems exclusively use 2D imagery. At the other end of the spectrum are systems with several thousand relatively high-quality 3D objects.

- *Interactive versus state-based systems*: Some systems allow the user to issue commands to incrementally modify the current state of the system (for example, to control a virtual character in a pre-existing

virtual environment), while in others the input language is self-contained and does not rely on interaction with a user or the state of the system itself. And some systems are batch-oriented without requiring any user at all.

- *Domain-specificity* can greatly affect the form and content of both the input text and the output graphics. For example, some systems are tailored to specific domains like car accident simulations or visualizing scenes from a specific film.

- *Graphical capability and style* can vary from system to system, ranging from 2D images to stylized 2D to photorealistic 3D. Some systems support just rigid 3D objects while others include posable or animated 3D characters.

- *Degree of language processing*: Some systems support very structured pattern matching templates while others employ full parsing and semantic analysis tailored to the application needs. Some systems handle real-world text rather than text created by the user of the system. The size of the supported lexicons also vary wildly, often as result of the type and scope of the graphical content.

- *Inference, semantic decomposition, and constraints*: Some systems use knowledge and constraints to construct the graphics and handle conflicting information. Some systems decompose into a set of graphical primitives whether or not they have explicit semantic representations.

- *Technique-oriented*: Some systems are focused on testing a particular technique, such as using voxelized 3D representations to compute spatial regions or using online machine learning to acquire knowledge about environments.

In this section we review the capabilities of these various systems, the earliest of which were from the 1970s. We group them into the following categories: scene construction (Section 1.3.1), virtual worlds (Section 1.3.2), animation (Section 1.3.3), and 2D Images and video (Section 1.3.4). We then (Section 1.3.5) contrast WordsEye in detail with two of the aforementioned systems that share some common themes with WordsEye. As part of that, we give a brief overall summary characterizing the differences between WordsEye and the various other systems reviewed.

1.3.1 Scene Construction

A number of very early (pre-1985) systems use natural language input to control computer graphics; this was an era when computer graphics was in its infancy. These systems, built around micro-worlds, are very limited in both the graphical and linguistic processing they perform but are interesting for historical interest and some of the issues they raise.

SHRDLU [Winograd, 1972] is a very influential system that uses natural language to interact with a "robot" living in a closed virtual world. Given the small number of objects and actions, the vocabulary and language processing can use pattern matching in a relatively flexible way to interpret natural language commands. The system also performs extensive inferences in combination with a physics model to figure out how to move and stack blocks and other simple objects to achieve the expressed goal. SHRDLU maintains a model of user state so that objects just referenced could be referenced in shorthand, without having to specify again which particular sphere or cube was meant. SHRDLU also allows users to assign names to objects or arrangements of objects.

Envi [Boberg, 1972] is an extremely early system (1972) that generate 3D line drawings from graphical commands. It highlights issues of handling graphical spatial defaults with two kinds of objects (bricks and wedges). For example, if a command is given to the system to create a brick, the location, orientation, and size of the brick would be defaulted. The system is incremental and state-based. For example, the user could issue additional commands to make an object bigger, using its current size as the new starting point. The system also allows constraints to be asserted (for example, "dont-change-the-position-of ob-1") and detected conflicts such as when two objects were located in the same place. To satisfy certain constraints, the system runs a planning algorithm to change the unconstrained aspects of existing objects.

Two other very early systems were also based on microworlds – the **Clowns Microworld** [Simmons, 1975] which generates 2D line drawings of a clown in a very simple environment and **Ani** [Kahn, 1979] which creates created 3D animation from "high-level, vague, and incomplete descriptions of films." The Clowns Microworld system represents a scene as a set of objects connected by relations in a graph. It supports a small handful of objects that constitute a pre-fabricated world (a boat, a dock, a lighthouse, the ocean) and can render pictures for sentences such as *A clown on his head sails a boat from the dock to the lighthouse*. A small set of primitives such as *attach* and *rightOf* are used to create the graphics. The Ani system creates schematic animations displaying the positions of characters on a stage in the Cinderella story. The graphics produced is simply the motion of 2D icons representing the position of characters as

seen from above. There is no natural language input to Ani. Instead, the user provides a high-level, semi-textual description to Ani. The description specifies the personality and appearance of the characters (e.g. evil or shy), of the relationships and interactions between the characters (e.g. hate or love) as well as the type of film desired. Ani then produces a film description utilizing its knowledge of the Cinderella story to determine the timing of various "choice-points" and to set the speed of characters moving based on textual cues. The film description is fed to another piece of software to produce the graphics.

[Waltz, 1980] explores the goal of understanding language descriptions by simulating the construction of a scene using a system by [Boggess, 1979]. This paper seeks to map out what inferences must be made to reason about space, common-sense physics, and cause-effect relationships in the context of scenes. It focuses on issues related to making predictions about described scenes and how to judge if a described scene is physically possible or not. Some examples of proposed primitives are: *break-object-into-parts, mechanically-join-parts, hit, support, touch, translate, fall*. Complex actions such as *bite* would be decomposed into primitives (such as touch, using the mouth) and conditions in sequence. The paper argues for running simulations of events in order to judge meaning and plausibility. As an example, two sentences are considered: S1: *I saw the man on the hill with a telescope* and S2: *I cleaned the lens to get a better view of him*. It is argued that a text understanding system must be able to discover that *lens* is part of the previously referenced *telescope*. Then the various ambiguities of the sentences are discussed and how making inferences could narrow down the possible interpretations.

[Adorni *et al.*, 1984] is another early system using natural language to create static scenes. The language input is restricted to phrases consisting of: <subject> <preposition> <object> (<reference>). The system employs a set of rules to interpret and map spatial prepositions to graphical primitives. These rules can invoke object properties to differentiate between different cases and find the right graphical primitive. The system also has a naive physics model and inference mechanism to infer spatial relations.

Some other early systems create scenes but dispense with the natural language textual descriptions altogether.

While not a natural language system, the **[Friedell, 1984]** system allows simple 3D graphics to be synthesized from a knowledge base of objects. The knowledge representation includes an object type and associated attributes. The system applies a set of graphics primitives to modify color and shading and 3D transformation. It distinguishes between *intrinsic structural modifiers* of objects that change the existing structure and *extrinsic structural modifiers* that make ancillary additions to the object (for example adding

a flag to a mast of a ship). It has no natural language component and employs what appears to be an *adhoc* system of knowledge representation.

IBIS [Feiner *et al.*, 1993] is not a natural language system but is an example of a system that generates visualizations that are automatically composed to convey intent. Instead of purely geometric relations between objects or parts, the system is focused on presentational constraints, such as whether a object or region is highlighted in color, the focal point in the scene, and so on. To do this the system draws upon a knowledge base of presentation rules and strategies.

Comic Chat [Kurlander *et al.*, 1996] is a somewhat different type of system where chat streams are automatically converted to comic book panels with the participants' dialog assigned to different 2D graphical comic characters. The emphasis of this system is on the presentational style and graphical layout of the resulting cartoon panels. The characters' gestures and facial expressions are determined by a set of simple rules. For example, greeting words in the chat will cause the chat character to be put into a waving pose. And self-references (such as *I* or *I'll*) will cause the character to point to itself. The system can also mix different input triggers and detect conflicts. In addition, it employs strategies to avoid shuffling of positions of characters from panel to panel.

The following systems are more modern and more comparable to WordsEye

Put is a system [Clay and Wilhelms, 1996] that allows control of spatial arrangements of objects in a pre-constructed 3D environment. The language input is restricted to phrases of the form $Put(X, P, Y)$, where X and Y are objects and P is a spatial preposition. It is a command-driven and state-oriented system, changing the current state of the world in an incremental fashion.

The **[Seversky and Yin, 2006]** system is a text- and voice-activated system that positions objects in a 3D space. The system uses 3D objects from the Web of varying quality and automatically creates a voxel representation to subdivide space so that spatial relations can be applied on relevant regions. The system supports the following spatial relations in some form: *on, under, above, below, behind, in front of, and in*. These can be modified with *to the left of*, *to the right of*, *next to*, *towards*, and *center*. This system's focus is on its interactive voice-based user interface and using voxels to perform some spatial layout. It has limited linguistic and spatio-graphical capability and assumes that input descriptions contain spatial relationships. As such, it does not handle surface properties, sizes, distances, anaphora resolution, verbs, compund constraints, and compound objects.

[Tappan, 1996] describes text-to-scene system that supports different spatial frames of reference in a

simple manner. It includes 58 spatial relations for position (*in front*), distance (*near*), orientation (*facing*) as well as frames of reference: intrinsic (object-centered) for objects with a front, extrinsic (environment-centered), and deictic (viewer-centered) for objects without a front. Spatial relations are represented by circular two-dimensional fields of 100 rings and 32 sectors. Regions and intersections specify preferred and prohibited locations. This system, as well as others by **Yamada** [Yamada *et al.*, 1988] and **Olivier** [Olivier *et al.*, 1995], handles a limited set of graphical constraints but has minimal knowledge of objects and their spatial structure and affordances.

AVDT (Automatic Visualization of Descriptive Texts) [Spika *et al.*, 2011] is a system that automatically generates 3D scenes from natural language input. It parses natural language text using the Stanford dependency parser [De Marneffe *et al.*, 2006] and employs distance heuristics to achieve a natural spacing between objects. It also uses collision detection and a physics engine to enforce some spatial constraints. The system does not support textures or colors. The system uses models from Google 3D Warehouse [Warehouse,]. We examine the differences between WordsEye and AVDT in more depth in Section 1.3.5 below.

The text-to-system by **[Chang *et al.*, 2014]** is focused around learning spatial knowledge from and about scenes. Its aim is to determine what objects typically appear in room and what spatial relations apply between those objects. The system learns by finding what objects tend to support one another. For example, tables are more likely to support plates and forks rather than chairs. The system generates scenes from language input by inferring spatial constraints based on that knowledge. It also allows users to refine the scene, and the system uses that information to learn more and improve its spatial knowledge base.

The system works by parsing the input text into a set of constraints. The constraints are expanded automatically based on implicit constraints in the knowledge base that are not specified in the input text. It uses the resulting constraints to arrange objects in the scene. The system uses about 12,000 3D models from Google 3D Warehouse [Warehouse,]. These models were tagged with text keywords and are automatically scaled and oriented to give them approximate real-world sizes and default orientation.

PiGraphs [Savva *et al.*,] is a very recent text-to-scene system that poses a 3D stick figure character performing a variety of different actions. The system uses a probabilistic model to learn poses from real-world observations using 3D sensors of people performing everyday actions and interacting with objects.

1.3.2 Controlling Virtual Worlds

Several systems have targeted animation rather than scene construction.

The **PAR System** is a graphical semantic framework, from the University of Pennsylvania's Center of Human Modeling and Simulation [Badler *et al.*, 1998] that uses language to control animated characters in a closed pre-constructed virtual environment. The system's PAR (Parameterized Action Representation) framework represents AGENT, SOURCE, GOAL, PATH, and other verb arguments. A PAR gives a complete represention of the action.

Each action also has applicability conditions, pre-conditions, and post-assertions. Applicability conditions must be satisfied for the action to be performed. Pre-conditions are pairs of conditions and actions to take to satisfy the condition. For example, a precondition for walking is to already be standing. Post-conditions are used to update the state of the world, for example, to update the current location of the agent after walking.

Input is parsed and used to instantiate a PAR. References are grounded in objects in the environment. PARs recursively execute and create new PARs that agents process in parallel in real time. The system supports a limited set of actions (walking, sitting down on a chair or bed, standing up, talking to others, climbing a ladder, opening a door, shaking hands, drinking).

PAR can also represent chains of actions for commands like "*go to bed*" that imply a sequence of sub-actions. PAR allows instructions such as "*if you agree to go for a walk with someone, then follow them*" to be given and then triggered in the future.

Ulysse [Godéreaux *et al.*, 1999] is an interactive spoken dialog system used to navigate in virtual worlds. It consists of a speech recogition system and natural language parser to control the movements of a virtual agent through a simple 3D virtual environment. The system also interprets deictic references based on the user's gestures, allowing the user to point at objects in the environment.

1.3.3 Animation

CarSim [Dupuy *et al.*, 2001] is a domain-specific system where short animations are created from natural language descriptions of actual accident reports. The system handles real-world, free-form text, using a corpus of 200 accident reports from Swedish newspapers. The length varies from a couple sentences to over a page. The system was evaluated; overall it was able to infer all relevant information in 23 of 50 texts. CarSim performs two main subtasks: a) a linguistic input module that performs partial parsing to extract the information from the text to fill templates representing semantic frames for accidents, representing the collisions and objects involved and b) a 3D simulation of the accident generated from the semantic

information.

In order to extract accident semantics from real-world text, the system does part-of-speech tagging, segment segmentation, and detection of noun groups and clause boundaries. It also performs domain-specific named entity recognition to handle the locations (e.g. cities or road names). And it performs domain-specific coreference and metonymy resolution of drivers and their vehicles (e.g "*a truck driver ran off the road*" versus "*a truck ran off the road*"). The system includes a special phrase structure grammar to handle time expressions. It sequences events with learned decision trees from TimeML [Pustejovsky *et al.*, 2003] and fills an ACCIDENT frame with slots for objects, road, collisions.

The system has a simple ontology that includes relevant objects such as ROADS (including subtypes such as crossroad, straight road, left_turn, and right_turn), TREES, TRAFFIC_LIGHTS, STOP_SIGNS, and CROSSINGS. The system handles temporal sequences – accidents themselves are comprised of an ordered list of COLLISIONS, where each collision is parameterized by the objects involved (the *actor* and *victim*) and the parts (front, rear, left, right, unknown) of those objects. These and other frame roles define the accident.

CONFUCIUS [Ma, 2006] [Ma and Mc Kevitt, 2004a] is a multi-modal animation system that takes as input a single sentence containing an action verb. The system blends animation channels to animate virtual human characters. CONFUCIUS leverages a graphically based verb ontology and what it calls *visual valences* in order to represent the required semantic roles that objects play in actual scenes (as opposed to just those that are required linguistically). The system utilizes lexical rules to convert from high-level verbs to basic actions and draws upon a graphics library of characters, props, and animations.

The taxonomy of verbs is based on visual valency. Often what is optional in syntax and semantics is obligatory for visual valency. For example, with *John is digging*, the syntactic valency is one (an intransitive verb that only requires a subject) but the visual valency is three, since the INSTRUMENT and PLACE are required to visualize the scene. The system also specifies *somatopic effectors* to classify actions by the participating body parts such as *facial expression*, *leg or arms*, and *torso*. And at the verb semantics level the system differentiates between levels-of-detail, for example: a pure event (*go*) versus manner (*walk*) versus troponym (a precise manner) (*limp*). We contrast some of the similarities and differences between WordsEye and CONFUCIUS in Section 1.3.5 below.

[Ye and Baldwin, 2008], from the University of Melbourne, uses machine learning to create animated storyboards on a pre-made virtual stage. This system works by assigning WordNet [Fellbaum, 1998] synsets to characters and objects on the virtual stage. The script, from the film *Toy Story*, was annotated, assigning

sequences of 21 base graphics commands to each eventive verb. For example, base commands for *The man grabs the mug on the table*: "Stand up; Move to mug; Extend hand to mug; End-of-sequence." The system uses various linguistic features, semantic role features, and graphical features to train a machine learning model on the annotated data. It then applies that model to a test set to produce graphics commands used to produce an animation.

[Glass and Bangay, 2008] is a text-to-animation system focused on solving constraints for trajectories of characters to produce an animation. The input is not natural language text but a templated XML-like annotation to the source fiction that specifies time-based parameters: avatars, objects, settings, spatial relations, and appearances of characters in the script. The system extracts graphical constraints from the annotations in order to compose the scene and create the animation.

1.3.4 2D Images and Video

The Story Picturing Engine [Joshi *et al.*, 2006] is a system for picturing stories by selecting images from image databases that best fit the text. It works by extracting keywords associated with WordNet synsets and matching them with an annotated image database to produce a ranking that is used to select images to represent the story.

[Schwarz *et al.*, 2010] is a text-to-video system that allows users to tell a story by semi-automatically retrieving sequences of pictures corresponding to text. The input text is segmented and tokenized in order to construct search queries using terms in the text. Different combinations of conjunctions and disjunctions can be used to construct queries of varying specificity. The system retrieves multiple images (up to a user-supplied threshold) that are returned to the user who then selects the best ones for each segment of the input text. The system assembles these into a storyboard or photomatic animation. The system relies on online photo sites with tagged images such as Flickr.

[Zitnick *et al.*, 2013] is a system that learns and creates scenes from 2D clip art. The system was trained and tested using a dataset of 10,000 images of children playing outside. To train the system they collected 60,000 sentences that described the scenes in different ways. They had Amazon Mechanical Turkers create scenes by positioning the clip art. There were 80 pieces of clip art representing 58 different objects for common objects such as people, plants, and animals. For text input, each sentence is represented by a tuple, where each tuple contains a primary object, a relation and optionally a secondary object. No parsing is performed. The system forms no explicit semantic model and uses conditional random fields to find the

closest scene to the input text.

1.3.5 Comparisons with WordsEye

We now compare, in more depth, WordsEye with two of the systems (AVDT and CONFUCIUS) described above. We choose AVDT since it is a recent text-to-scene system with a relatively well-defined set of capabilities and roughly the same goals as WordsEye. And we look at CONFUCIUS because it shares with WordsEye some of the same insights about lexical and graphical meaning. Finally we summarize the differences between WordsEye and the other systems we review.

1.3.5.1 Comparison with AVDT

The AVDT system, like WordsEye, is a tool for creating a scene by describing it in language. It uses collision detection and a physics engine to enforce some spatial constraints. It also employs heuristics to achieve a more natural spacing and orientation of objects in the resulting scene. Those are its main unique features. WordsEye, however, is based on a much richer representational system at all levels, from language to semantic representation to graphical knowledge and representation. The following is a partial list of significant differences:

- AVDT has no support for changing surface properties such as colors, textures, reflectivity, and transparency. WordsEye allows any of these properties to be changed by language. In addition, textures and colors are treated like normal entities by WordsEye that can have arbitrary properties. This allows texture sizes to be changed (*The plaid texture is on the bowl. The texture is 3 INCHES WIDE*) and color modifiers to be applied (*The PALE RED bowl*).
- The AVDT system has single interpretations for spatial prepositions between a FIGURE and GROUND. For example, *on* places the FIGURE object above the bounding box of the GROUND object, and *in* places the FIGURE object on the bottom of the GROUND object. This is insufficient given the varied ways that spatial prepositions can be interpreted and realized (see Figure 3.3). WordsEye, in contrast, applies its knowledge base and representational system (VigNet) to handle ambiguity and interpret a wide range of linguistic and semantic constructs.
- The AVDT spatial relations are enforced using bounding boxes for the objects in conjunction with a physics engine. This can work for simple objects but cannot account for objects with more structure and those with intrinsic object affordances (Section 3.7) which WordsEye handles explicitly by asso-

ciating with a geometric part. For example, in AVDT, the *in* relation puts the FIGURE object on the bottom of the GROUND object. However, no mention is given as to how to find the appropriate lateral position in which to find the lowest area. For a symmetrical bowl, this might not be a problem, but irregularly shaped objects would pose difficulties. In addition, the collision detection used by AVDT uses the bounding boxes of objects and hence cannot account for internal shape and structure, such as the floor of a birdcage, where there is no direct path from above.

- The AVDT system has a simple spatial layout model that assumes each object in the scene is dependent on a single other object, thus forming a tree of spatial dependencies. This model does not allow multiple constraints on the same object (e.g. *the dog is 3 feet to the left of and 2 feet behind the chair*). WordsEye can both represent and process multiple spatial constraints on the same object (Section 6.5). In addition, it is unclear how AVDT handles under-constrained objects. For example, [*The dog is on the table. The cat is on the table.*] does not specify where the dog and cat are placed relative to each other. It is possible that AVDT uses its collision detection to force them to be in separate locations. But if that is the case, there would then need to be a way to make that defeasible so that the objects could interpenetrate when the language specifies it. WordsEye has support for both adding implicit constraints and having them being overridden.
- The AVDT system has no support for explicitly specified sizes, orientations, and distances. It thus cannot handle sentences such as *the dog is 3 feet tall*, or *the dog is 3 feet behind the door*, or *the dog is facing the wall*. And as a result, it also has no support for more nuanced graphical-semantic issues, such as how to make a fence wide (by replication) versus making a sofa wide (by scaling) as described in Section 3.5.2. More generally, the AVDT system is limited by representing object properties with a flat list of slots per object – there is no capability to represent modifiers on the slots themselves or to have multiple slots of the same type.
- AVDT has no concept of collections of objects that can be given properties or otherwise referenced and positioned as a unit. It only allows a QUANTITY slot for each object without allowing an object to reference sub-objects. The same issue applies to parts of individual objects, where AVDT has no mechanism to differentiate between a part and a whole and the relation between the two.
- WordsEye's coreference module (Section 6.3) supports anaphora resolution (*THE CAT is on the table. IT is facing left*). It also handles merging of individuals into collections (*THE CATS are on the table. THE BIG CAT is facing right.*). It also takes into account definite versus indefinite reference and

adjective modifiers as they affect coreference merging. AVDT, in contrast, cannot handle anaphora and distinguishes cofererences of existing objects from new object references by heuristics related to spatial dependency, assuming that the same object cannot be multiply dependent, and hence a new reference as a dependent must imply a separate object. This assumption does not generally hold, since sentences can often impose multiple constraints on the same object.

- AVDT only references objects by their direct names (e.g. *couch* versus *furniture*). The authors mention the possibility of using WordNet to add more flexible naming. WordsEye, on the other hand, has a full ontology similar to WordNet but with richer semantics. For example, in WordsEye, the phrase *the dangerous animal* can limit scope to those animals that are tagged as being dangerous in the VigNet knowledge base. See Sections 2.3.3, 5.2, and 6.4.3.1.
- AVDT system has no specified user interface. WordsEye handles a number of subtle user interface issues, such as stability of object assignments and camera position as the scene is being incrementally built and objects come and go. See Section 6.8.

See Chapter 8 for an evaluation and error analysis of WordsEye's depictions of single sentences, including a set of captions from the PASCAL image dataset [Rashtchian *et al.*, 2010]. As part of our error analysis, we also estimate when the capabilities highlighted above are required to properly process the evaluation sentences but are missing in AVDT while being supported in WordsEye (Section 8.6.2).

1.3.5.2 Comparison with Confucius

The CONFUCIUS system is a text-to-animation system that produces an animation of a human from an input sentence. As such, it has somewhat different goals than WordsEye and is concerned with issues such as representing temporal intervals in order to control the timing of actions and sub-actions in an animations. And since CONFUCIUS is a text-to-animation system it does not handle the language or graphics needed to construct a scene. It is also limited to single sentence input. However, at a deeper level, it recognizes (like WordsEye) that linguistic semantic roles are quite different than the graphical roles needed to represent a scene. CONFUCIUS classifies motion verbs into an ontology based on what it calls *visual valences* in order to represent the semantic roles that objects play in those actions.

CONFUCIUS shares the insight established by the first version of WordsEye [Coyne and Sproat, 2001] that object-specific spatial semantics (affordances) can be used to perform graphical positioning. These affordances and associated graphical knowledge (such as how different types of objects are grasped) are

applied to posing and animating characters ([Ma and McKevitt, 2006]). The set of affordances used by CONFUCIUS is influenced by the "smart objects" described in [Kallmann and Thalmann, 2002]. CONFUCIUS calls these "space sites" and uses them to define locations where humans would grasp the given object for animation purposes. The current version of WordsEye, while not dealing with animated characters, defines a significantly richer set of spatial affordances (including those for both purely spatial positioning and for use in human poses) based on inspecting its entire library of 3D objects (Section 3.7). It encodes these within a broad semantic theory and supporting knowledge base (Section 2.4) centered around *vignettes*. This allows a separation of graphical meaning from lexical meaning and provides a knowledge-driven way to translate between the two. We describe in more detail in the rest of this thesis, especially in Chapters 2, 3, and 4.

To convert text to animation, CONFUCIUS utilizes lexical rules to convert from high-level verbs to basic actions and draws upon a graphics library of characters, props, and animations scripts to convert those actions in an actual 3D animation. One limitation of this approach is that there is no conceptual separation between lexical verb-based semantics and graphical meaning since semantic decomposition is directly tied to verbs themselves. In contrast, vignette semantics is based on the realization that different verb arguments (as translated into semantic arguments) can radically change the structure of the graphical meaning and representation. They are, therefore, essential in determining the actual set of graphical semantic roles and constraints. For example, at the graphical level, *washing hair* is completely different from *washing a car*. While conversely, *washing* an apple at a sink has much more in common with *filling* a bottle with water at a sink than it does with *washing* a car with a hose and a bucket in a driveway.

So while the idea of visual valences points in the same direction as vignettes, it is limited by being tightly tied to verb semantics. Vignette semantics, instead, bases graphical meaning on an independent hierarchy of graphical and real-world structures (vignettes). In our theory, the locus of graphical meaning (and hence much of meaning, more generally) is not the verb or lexical item but the vignette that instantiates it.

In Section 2.6 we further discuss the visual valence ideas embedded in CONFUCIUS along with other related semantic theories and how they compare with WordsEye.

1.3.5.3 Summary

In many of text-to-graphics systems, the referenced objects, attributes, and actions are relatively small in number or targeted to specific pre-existing domains. The semantic analysis (and often language processing)

is thereby similarly narrowed. The earlier systems tend to have an affinity to the SHRDLU system [Winograd, 1972] in their focus on constraints in a small, closed virtual world. A common theme in several of these systems is that of grounding to a set of graphical primitives defined for their domain. Some systems add semantics to 3D objects, an approach that is also relevant in computer game development [Tutenel *et al.*, 2009]. A survey of natural language visualization systems [Hassani and Lee, 2016] compares many of the above systems in addition to others.

WordsEye, is a much larger and broader text-to-scene system than the systems described above. The other systems are generally either domain-specific or tailored towards utilizing certain computational techniques. In addition, WordsEye is the only system built on an integrated and extensive semantic representational framework and knowledge base tied to a rich set of graphical and lexical semantics constructs. Also, unlike other systems, WordsEye has been used and tested with thousands of real-world users, both as a 3D authoring tool and in education. There is not as yet any comprehensive and agreed-upon way to compare text-to-scene systems.

.

1.4 Applications

Text-to-scene conversion has potential application in several areas. WordsEye has undergone extensive testing as an online graphical authoring tool and been evaluated in the classroom as a way to build literacy skills.

- **Education:** Seeing sentences spring to life makes using language fun and memorable. This suggests many uses in education including ESL, EFL, special needs learning, vocabulary building, basic numeracy skills, and creative storytelling. In Chapter 7 we describe tests of WordsEye in educational settings, including a formal experiment to help improve literacy with 6th-grade students in a summer enrichment program in Harlem.

- **Graphics Authoring and Online Communication:** Textually generated scenes can be created in a fraction of the time it takes to make a scene by hand. Traditional 3D graphics authoring is a difficult process, requiring users to master a series of complex menus, dialog boxes, and often tedious direct manipulation techniques. Natural language offers an interface that is intuitive, well-suited for voice interfaces, and immediately accessible to anyone, without any special skill or training.

This increase in speed and lower entry barrier enables a new form of social interaction and promotes "visual banter" where a user can respond to one scene by making one of their own. We have seen much of this in our online website (`http://www.wordseye.com`) and gallery. In Chapter 9 we describe some metrics from online testing that we have gathered from over 20,000 registered users who have created 25,000 final rendered scenes.

- **Image Search-as-Generation:** Image search is an area of much research and commercial activity. We believe WordsEye has potential to supplement traditional image search techniques with images generated on the fly from the search query. We describe an evaluation (Chapter 8) that tests imaginative and realistic sentences with WordsEye and Google Image Search.

- **Visual Summarization:** Online social media and the Web, in general, are filled with text, with few tools available to visualize its meaning. Word clouds [Wikipedia, 2016m] provide a shallow visual representation of meaning by distilling the input text to a set of standalone words that are displayed in different sizes corresponding to their saliency in the text. Text-to-scene generation has the potential to much more concretely depict the meaning of text.

- **3D Games, Storytelling and VR/AR:** 3D games are painstakingly designed by 3D artists – the malleability of the graphics in games is usually limited to external level design tools and rigid interfaces for character selection and modification. Some games, such as *Spore* [Spore,], allow interesting variability of graphical elements, and spoken language commands are supported by games such as *Tom Clancy's EndWar* [Wikipedia, 2016n]. We foresee this trend of language-based interfaces to games continuing. More recently, virtual reality and augmented reality technology have become more affordable and widespread. We believe WordsEye has great potential as a way to interact with virtual environment with voice commands to create or modify the virtual scene.

- **Field linguistics tool:** Linguists need tools to document endangered languages. In Chapter 10 we describe a pilot study using WordsEye for language elicitation in field linguistics.

1.5 Outline of Thesis

We begin in Chapter 2 by reviewing several knowledge representation frameworks and lexical resources. This leads to a discussion of vignette semantics as it is formalized and instantiated within the structure of

VigNet, our new lexical resource and ontology. VigNet is the backbone of WordsEye, providing all the lexical, semantic, and graphical knowledge used to depict scenes. We examine key issues in graphical semantics in Chapter 3 and lexical semantics in Chapter 4. In Chapter 5, we discuss the annotation tools and other techniques we used and plan to use to populate VigNet. We then pull it all together in Chapter 6 where we describe the computational flow of the system, including all stages of the linguistic, semantic, and graphics processing needed to convert text to a 3D scene.

The next chapters focus on evaluating and testing WordsEye in different ways. Chapter 7 describes testing as a literacy tool in education. In Chapter 8 we evaluate WordsEye at the sentence level by comparing scenes for "imaginative" and "realistic" sentences against output produced in a Google Image Search. Chapter 9 describes testing of WordsEye as an online 3D authoring tool. Chapter 10 describes a pilot study for using WordsEye as a field linguists tool. We close, in Chapter 11, with a summary of contributions of this work and a discussion of future directions.

Chapter 2

Vignettes – Meaning and Knowledge Representation

2.1 Introduction

In this chapter we develop our vignette-based semantic theory. This theory encompasses both the lexical manifestations of semantics and the grounding of semantics into graphically realizable properties and relations. We start by reviewing several existing semantic theories (Section 2.2.1) as well as relevant lexical resources (Section 2.2.2). We then describe VigNet (Section 2.3), the lexical and grounded semantics framework and resource (section 2.5) we have developed to support WordsEye's text-to-scene translation. Finally, in section 2.6 we compare vignette semantics with several of these other semantic theories.

2.2 Background

2.2.1 Semantic Theories and Frameworks

There is a long history in artificial intelligence and cognitive linguistics of decomposing meaning into semantic primitives. These efforts fall into two broad classes – those focusing on primitive features of objects used to distinguish one object from another (for example in prototype theory [Rosch, 1973]) and those focused on state changes, temporal relations, and causality [Miller and Johnson-Laird, 1976]. In addition, different frameworks focus to different degrees on linguistic and lexical issues versus more abstract seman-

tics that are aimed at representing facts about the world (even if mediated through language). For reference, we highlight several relevant semantic frameworks below.

Conceptual Dependency Theory (CD) [Schank and Abelson, 1977] is an early computational approach that specifies about a dozen functionally oriented primitives (such as change of location, transmission of information, grasping, and change of possession) into which all meaning is reduced. CD theory offers a representation system grounded in rather abstract concepts but has little to say about either lexical semantic matters or semantics grounded in 3D spatial relations. An offshoot of CD is Kodiac [Wilensky, 1986], which is influenced by frame-based knowledge representation paradigms such as KRL [Bobrow and Winograd, 1977].

Linguists have studied semantics from a non-computational point of view. **Natural Semantic Metalanguage** (NSM) [Goddard, 2008] is an attempt within linguistics to describe the *semantic primes* that occur through all human languages. NSM argues that creating an artificial meta-language to represent semantic concepts is unnecessary and ultimately futile since any artificial language still has to be interpreted by people using the representation. Instead, it is preferable to use a regularized version of an actual natural language and semantic elements which exist in all natural languages. Semantic primes are thus lexical units that denote a particular word sense rather than mere lexemes. It is a tenet of NSM (the "Strong Lexicalisation Hypothesis") that all semantic primes can be expressed in every language, whether through a word, morpheme, or fixed phrase. Semantic primes occur in several different categories, for example: SUBSTANTIVES (*I, you, people*), EVALUATORS (*good, bad*), MENTAL PREDICATES (*feel, know, think*), and SPACE (*where, far*).

Lexical Conceptual Structure (LCS) is a linguistically oriented approach that decomposes lexical relations into a set of primitives (similar to those in CD theory) [Jackendoff, 1985]. LCS posits certain conceptual categories such as THING, EVENT, STATE, ACTION, PLACE, PATH, PROPERTY, and AMOUNT. It then defines primitive functions that reference those categories and maps them into the same or different categories, thereby recursively elaborating them. Examples of functions for different types are EVENTS such as GO, STAY; STATES such as BE, CAUSE, ORIENT; PATHS such as TO, TOWARD, FROM, AWAY-FROM, and VIA. Lexical items then trigger these lexical conceptual structures. For example, a sentence such as *John ran into the room* would link a conceptual structure with EVENT=GO (for ran) that takes a first argument of type THING=JOHN and a second argument of a PATH, where PATH is filled recursively by other types and functions representing *into the room*.

To avoid an infinitely complicated categorical feature system, LCS theory assumes a graphical grounding

behind lexical concepts in which the lexical conceptual structure ultimately references a graphical 3D model. This notion is based on Marr's work in computer vision [Marr and Nishihara, 1978] in which 3D mental models represent objects in space. These 3D models are hierarchical in nature and involve various primitive shapes. Each shape representation uses an object-centered coordinate system that is composed of volumetric primitives of various sizes. There is a principal axis for the overall object hierarchy, and each subobject has its own axis, length, and 3D transformation. This is very much in the spirit of skeleton control rigs used in modern 3D animation systems and methods for automatically "rigging" skeletons to 3D polygonal meshes [Baran and Popović, 2007]. We note that while LCS posits the existence of such mental 3D models that are ultimately referenced by the conceptual structure, there is no discussion or proposed formalisms for how that interface would actually work.

In order to keep the lexical conceptual structure simple and avoid an infinite number of categorical features, LCS assumes that gradable attributes reference focal points on domains. So, for example *hot* will reference a point along a temperature domain.

We discuss LCS further, and compare it to vignette semantics, in Section 2.6.

Lexical Functions in Meaning Text Theory: [Melcuk, 2012] Meaning text theory (MTT) is an all-encompasing linguistic theory that formalizes the ways meaning can be translated from text to syntax to semantics. The semantic representation is a graph of connections where predicate nodes are linked via edges to argument nodes. Arguments can share multiple predicates. One particularly interesting aspect of this theory is the inventory of *lexical functions*. Lexical functions (LF) are formalized ways of mapping from one lexical unit to a related lexical unit or set of lexical units. Melcuk distinguishes between *paradigmantic* and *syntagmatic* LFs. Paradigmatic LFs map to individual words that stand alone independent of syntax, while syntagmatic LFs that take the same syntactic position as the argument to the function. For example, *Magn* is a syntagmatic LF that amplifies the strength of its argument. And *Conv* is a paradigmatic LF that returns a conversive for the argument. Some examples of four of the approximately 60 LFs in MTT:

Amplify: Magn(*patience*)= *infinite*, as in (*infinite patience*)

Conversive: Conv(*include*)=*belong*

Substantival: S0(analyze) = analysis

Standard Instrument: Sinstr⊃(shoot) = firearm

Embodied Construction Grammar [Bergen and Chang, 2005] [Bailey *et al.*, 1998] is a cognitive lin-

guistics framework for grounding construction grammars in sensorimotor percepts and mental simulations. It posits that pairings between lexical form and function (semantics) are learned and that these pairings and structures have meaning on their own that are imposed on the components. As a result, lexical items are linked to a detailed specification of actions in terms of elementary body poses and movements. This approach builds on the theory of *affordances* so that, for example, the meaning of a noun such as *cup* depends on the affordances it provides and hence how it is used. **Praxicon** [Pastra, 2008] is another grounded embodied conceptual resource that integrates motor-sensoric, visual, pragmatic and lexical knowledge (via WordNet). These frameworks are influenced by issues coming out of robotics.

Event Logic [Siskind, 1995] decomposes actions into intervals describing state changes and allows visual grounding by specifying truth conditions for a small set of spatial primitive relations (a similar formalism is used by the CONFUCIUS text-to-animation system [Ma and McKevitt, 2006]). Such primitives include SUPPORT, PICK-UP, ATTACHED, and TOUCHES. The theory aims to infer relations based on observable sequences of relations and a calculus for inferring possible resulting states.

Resource Description Framework (RDF) [Klyne and Carroll, 2006] is a formalism and format for encoding knowledge for the Semantic Web in graph structures [Berners-Lee *et al.*, 2001]. Nodes are connected by edges, where both nodes and edges are represented as Universal Resource Identifers (URIs)[Wikipedia, 2016o] or in some cases literals. RDF triples consist of an edge (PREDICATE) connecting two nodes (SUBJECT and OBJECT). The graph structure can be serialized in an XML syntax and other formats. **SPARQL** is an SQL-like semantic query language for RDF graphs consisting of database queries where patterns contain logical operators and logic variables to unify referents that match different parts of queries [Pérez *et al.*, 2006] [SPARQL,]. **OWL** (Web Ontology Language) [OWL,] is an extension to RDF, based on description logic [Brachman *et al.*, 1992] that supports the definition of ontologies.

Abstract Meaning Representation (AMR) [Banarescu *et al.*, 2012] is a semantic formalism for representing the meaning of sentences in order to annotate textual corpora. These annotated sentences can then be used to train semantic parsers. Sentences are represented by rooted, directed, acyclic graphs between nodes where edges are relations and nodes represent concepts. AMR uses PropBank frames [Palmer *et al.*, 2005] to represent relations between nodes. These same frames can be used in phrases lacking verbs. For example, the phrase *bond investor* could use the frame INVEST-01. AMR allows argument structures to refer to the same elements more than once to capture a) multiple sentences or b) sentences with explicit coreference like *JOHN said HE saw two flying saucers yesterday* where *JOHN* and *HE* co-refer to the same entity or c)

implicit coreference as in *the dog wanted to chase the squirrel* where *dog* is both an explicit argument of *want* and an implicit argument of *chase*. One goal of AMR is to abstract away from syntactic idiosyncrasies so that sentences with the same meaning are assigned the same AMR. The example is given of the following sentences [*he described her as a genius*], [*his description of her: genius*], and [*she was a genius, according to his description*] which would all be represented with the same AMR structure.

ISO-Space [Pustejovsky *et al.*, 2011] is a proposed mark-up language based on SpatialML [Mani *et al.*, 2008] to be used to annotate spatial relations, particularly between geographical entities and other entities situated on a geographical scale such as roads, bodies of water, and countries. A location can be modified by various topological, orientational, distance relations, and attributes. Two main types of elements can be distinguished: ENTITIES (location, spatial entity, motion, event, path) and SPATIAL RELATIONS (topological, orientational, and distance). Spatial entities are specified either relative to one another or by using GPS coordinates. Spatial relations can be decomposed into a set of five parameters: topological (*in*), orientational (*left of*), topometric (*near*), topological-orientational (*over*), and metric (*2 feet away*). Orientational relations specify a frame-of-reference. The annotation scheme is intended to be applied to a variety of corpora, particularly: written directions, standard newswire, location descriptions, interior descriptions, travel blogs.

The **Generative Lexicon** [Pustejovsky, 1995] is a semantic theory where lexical entries are assigned semantic structures, with entries for TYPE, ARGUMENT, EVENT, and QUALIA. For example [Johnston and Busa, 1999], for a *knife* the TYPE is an ARTIFACT_TOOL and the ARGUMENT structure defines the other various semantic roles (and their types) that are associated with the word *knife*. The QUALIA structure assigns relational properties to those argument roles to, in effect, define the overall lexical entry. The entity's QUALIA structure, itself, consists of sub-structures for the following: FORMAL (its type), CONSTITUTIVE (its parts or other consituents), TELIC (what it is used for), and AGENTIVE (how it is brought into existence). So, for example, in the case of a *knife*, the TELIC entry would specify that a knife is used by a human to *cut* a physical object. The various roles (for the human and object) are all specified in the ARGUMENT list.

For verb lexical entries, the EVENT structure defines the sub-events and the causal and temporal constraints implied by the given top-level event. As with nouns, the QUALIA structures for verbs are constructed to define relations between the semantic arguments. The verb *break*, for example, will specify a FORMAL value stating that the object argument is *broken* and an AGENTIVE value specifying that the act of *breaking* brings that state into existence.

Another innovation of the Generative Lexicon is how it accounts for regular polysemy. For example,

a *book* is both a physical object and an information source, while a *bank* can be either an institution or a building. The Generative Lexicon captures these regularities by defining lexical items as merged ("dotted") types that can encompass multiple related meanings. At a practical level, this helps reduce the proliferation of related word senses. And at a theoretic level, this provides a framework for a truly generative lexicon where word meanings can interact with each other based on what is allowed by their internal semantic structure.

We compare qualia structures to vignette semantics in Section 2.6.

2.2.2 Lexical Resources

In this section we review existing lexical resources and knowledge bases and highlight their limitations.

2.2.2.1 WordNet

The WordNet [Fellbaum *et al.*, 2007] ontology provides a taxonomy of words grouped into separate synsets, representing common meaning and word sense. For example, *car* and *automobile* are in the same synset, whereas *bank* (the institution) and *bank* (the incline) and *bank* (the building) are in separate synsets. Synsets are related to one another primarily by hypernym (IS-A) relations. This provides useful information such as the fact that a *chair* is a type of *furniture*. WordNet is a very useful resource for many tasks; however, the taxonomy is ultimately unsuitable for our purposes. In particular, it fails in several ways:

– Inconsistency and lack of important semantic relations between entities. For example, the WordNet synset for *princess* is a hyponym of *aristocrat*, but there is no encoding of the basic fact that a princess is also a female. Likewise, there is no semantic link to distinguish between polysemous terms and separate word senses.
– Conflation of lexical usage with functional semantic categories. For example, *spoon* is classified as a *container*, which is true in some sense; however, it does not match normal usage, since a spoon is unlikely to be called a container by typical speakers of English.
– Replete with obscure word senses, such as *spoon* being a type of golf club. WordNet orders words senses by frequency, but the multiplicity of obscure and seemingly overlapping senses make it unwieldy.
– Ontological confusion. Occupations such as *president* and *infielder* are classified under *activity*. They should not be activities but rather *roles* in activities.
– While WordNet does encode some part-whole and substance-whole relations, this information is very

spotty and many very common facts are missing, such as the fact that lamps have light bulbs and that snowballs are made of snow. Furthermore it is not evident what criteria is used to declare part-whole relations. In other words, whether the relation is necessarily true as a matter of definition, whether it is true empirically, or if it is sometimes or often true. Also, either hyponyms do not necessarily inherit relations from their hypernyms, or the ontology is inconsistent in that respect. For example, *loafer* is a hyponym of *shoe*, and shoe has *lace* as a part! Additionally, *shoe* has both *lace* and *shoelace* as parts even though *shoelace* is a hyponym of *lace*. This is also wrong since *bootlace* is hyponym of *lace* and part of *boot*, not *shoe*. WordNet also has no way to encode how many parts of a given type an object has.

– No encoding of functional properties of objects such as the fact that a *mop* is an INSTRUMENT used in cleaning floors or that a *teacher* is the active partipant (AGENT) in teaching. A broader set of semantic relations is crucial to the resolution and grounding of lexical references into 3D scenes.
– Very little in the way of lexical functions to map the structural links between lexical items. For example, that *investor* is the deep subject of the verb *invest*. WordNet is fundamentally lacking in semantic structure which is needed to express the relationships between words. This requires something more similar to FrameNet, which we look at next.

2.2.2.2 FrameNet

FrameNet is a digital lexical resource for English that groups related words together into semantic frames [Baker *et al.*, 1998]. FrameNet contains 10,000 lexical entries where each lexical unit is associated with one of nearly 800 hierarchically related semantic frames. Each frame represents the common meaning of the lexical units in that frame. Each lexical unit is also associated with a set of annotated sentences that map the sentences' constituent parts to their frame-based roles. A FrameNet frame consists of a set of slots, called *frame elements* (FEs), representing the key roles characterizing the meaning of lexical units in that frame. Frames can contain any number of lexical units. For example, the COMMERCE_SELL frame includes FEs for SELLER, GOODS, and BUYER and has lexical units for the verbs *retail*, *sell*, *vend* as well as nouns such as *vendor* and *sale*.

Valence patterns map grammatical roles to FEs for a given verb. Every verb typically has many valence patterns, representing the various ways that verb can be used in sentences. So, for the verb *give*, the sentence *John gave the book to Mary* has the valence pattern of: ((Donor Subject) (Theme Object) (Recipient Dep/to)). And *John gave Mary the book* has the valence pattern of ((Donor Subject) (Recipient Object)

(Theme Dep/NP)). FrameNet also supports *frame-to-frame relations* – these allow frames to be inherited, to perspectivize one another, or to map to a sequence of temporally ordered subframes.

FrameNet provides no semantic information to distinguish the meaning of words in the same frame. For example, the SELF_MOTION frame contains a large number of verbs related only by the fact that the SELF_MOVER moves under its own power in a directed fashion without a vehicle possibly from a SOURCE to a GOAL and along a PATH. As a result, this frame contains strongly related verbs such as *walk* and *stroll* but also verbs with very different manner of motion such as *swim*, and *swing*. Likewise there is no representation in FrameNet of synonymy, antinomy, or other lexical semantic relations.

FrameNet frames are related to one another by a fixed set of frame relations. These allow verbs in different frames (e.g. *buy* and *sell*) to be related to one another. The main frame relations are INHERITANCE, PERSPECTIVE_ON, INCHOATIVE_OF, CAUSATIVE_OF, and SUBFRAME.

2.2.2.3 VerbNet and Affiliates

VerbNet [Kipper *et al.*, 2000] primarily focuses on verb subcategorization patterns and alternations grouped by Levin verb classes [Levin, 1993]. Verb arguments are mapped to a universal set of *thematic roles* [Fillmore, 1968] (such as AGENT, BENEFICIARY, PATIENT, and CAUSE) available to that class of verb. For example, the BENEFICIARY role is used by verb classes for verbs such as *get* and *steal*. VerbNet puts the thematic roles into an inheritance hierarchy so that one role can be a subclass of another.

VerbNet frames can also incorporate semantic types to restrict the values of the arguments. Semantic restrictions include broad types such as animate or organization. VerbNet differs from FrameNet in that it groups just verbs (no nouns or adjectives) and those groupings are based on subcategorization and alternation patterns rather than common meaning [Baker and Ruppenhofer, 2002].

VerbNet also grounds verb semantics into a small number of causal primitives representing temporal constraints tied to causality and state changes such as CAUSE and CONTACT. It does not handle spatial relations or try to model typical scenarios.

OntoNotes [Hovy *et al.*, 2006] is a resource that links VerbNet to WordNet senses. SemLink [Palmer, 2009] is another resource that links VerbNet, WordNet, FrameNet, and PropBank [Kingsbury and Palmer, 2002]. Daniel Bauer has used VerbNet to populate FrameNet valence patterns to be imported into VigNet [Bauer and Rambow, 2011]. See also [Coyne and Rambow, 2009] for the automatic generation of conversives and other verb alternations from FrameNet valence patterns.

2.2.2.4 Other Resources

In the Open Mind Common Sense (OMCS) project [Singh *et al.*, 2002], online crowd-sourcing is used to collect a large set of common-sense assertions. These assertions are gathered from web users with open prompts (e.g. "what is one reason you would ride a bicycle") and with structured templates: (e.g. "___ can be used to ___"). The reliability of assertions can be boosted either explicitly or as a side-effect of many users making the same assertion. The goal of ConceptNet [Havasi *et al.*, 2007] is to extract semantic relations from these text assertions. In ConceptNet, OMCS sentences are normalized to a fixed set of 23 binary relations using regular expressions.

There are several problems with this resource for our purposes: First, the semantic relations are overly general. For example, the resource asserts both [*book* LOCATED-AT *desk*] and [*book* LOCATED-AT *library*]. These are conceptually quite different relations, a SUPPORTING-SURFACE versus a VENUE. Secondly, there is no word sense disambiguation for the arguments. For example, we find that [*coin* LOCATED-AT *bank*] but also [*bridge* LOCATED-AT *bank*]. Thirdly, all relations are binary and as a result there is no context for the relations. For example, cereal would be in a bowl when being eaten but in a box when being stored. OpenMind cannot easily encode these distinctions.

Other resources: LabelMe [Russell *et al.*, 2008] is a corpus of 2D images of locations labeled with lexically tagged polygonal regions. Corelex [Buitelaar, 1998] is an ontology extracted from WordNet that models systematic polysemy. Yago [Suchanek *et al.*, 2007] is an ontology based on WordNet that extracts additional synsets from Wikipedia. The Preposition Project [Litkowski and Hargraves, 2005] is an extension of FrameNet that annotates the sense of prepositions on top of normal FrameNet annotations. Omega [Philpot *et al.*, 2005] is an ontology that merges WordNet with Mikrokosmos [Mahesh *et al.*, 1996].

2.3 VigNet Ontology and Knowledge Base

In this section we describe the formal mechanisms (concepts, semantic relations, translation-rules) we use to represent and translate meaning from linguistic input to high-level semantics to grounded graphical scenes (vignettes, primitive graphical relations, and graphical objects). We then describe the lexical semantic resource and ontology (VigNet [Coyne *et al.*, 2011a]) we have built based on this framework. The formulation of the framework for VigNet grew out of extensive discussions with Daniel Bauer and Owen Rambow.

The semantic framework we present can represent world knowledge, the meaning of sentences, concep-

tual knowledge, and lexical conceptual knowledge (e.g. word definitions). Our semantic representations, in addition, must deal with the following issues: ambiguity, vagueness, polysemy, and synonmy (both lexically and semantically). For example, *the book is red* and *the color of the book is red* should be mappable to the same semantic representation.

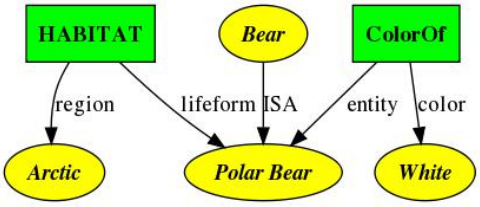


Figure 2.1: Definition of *polar bear*. Note: We show the IS-A relation as a link between the concepts for *polar bear* and *bear* for convenience. Conceptually it is just another relation.

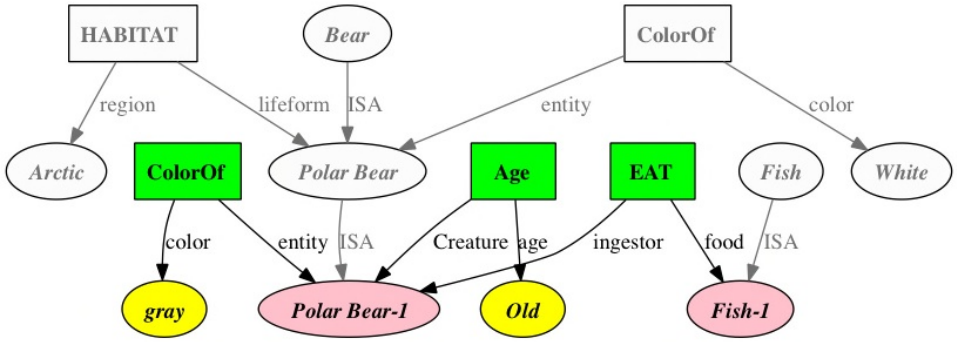


Figure 2.2: Sentence: *the old gray polar bear ate the fish*

2.3.1 Framework – Concepts, Semantic Relations, Lexical Items, Semantic Nodes

In order to represent lexical knowledge, world knowledge, and sentence meaning, we define the following:

2.3.1.1 Semantic Nodes

Semantic Nodes are any sort of item in the ontology or knowledge base. They can be concepts, semantic relations, or lexical items – these are all described below.

2.3.1.2 Concepts and Literal Values

Concepts represent entities of any type, from physical objects, to events, to 3D objects, to individuals, to abstract concepts, to psychological states – anything that can be referred to. Concepts are pure identities with no internal structure and no built-in semantics. The meaning they have is given to them by the semantic relations that reference them. Concepts often correspond to lexical items, in which case they have an associated lexical entry. Concepts can be *subtyped* by creating an arbitrary number of sub-nodes. The sub-nodes are related to their parents with IS-A relations.

Literal Values are numbers, vectors of numbers, colors, text strings, and numeric ranges (with a minimum and maximum value). Like concepts, they can fill argument values in semantic relations.

2.3.1.3 Semantic Relations

Semantic Relations are n-ary relations whose arguments can be a semantic node or literal value. Semantic relations are used to assert facts about their arguments. They can be used singly or in combination to represent word meaning to, in effect, define the meaning of a word (see Figure 2.1). They can also be used, as well, to represent the meaning imposed by a full sentence (see Figure 2.2). And the same inventory of semantic relations is also used to assert world knowledge. Semantic relations that are asserted on high-level concepts can be inherited or overridden by their children (subtypes). VigNet's *knowledge base* consists of a large inventory of semantic relations and assertions using those semantic relations. Semantic relations in some cases correspond to FrameNet frames, but are also subtyped to correspond to individual lexical units (verbs, prepositions, and adjectives). This allows knowledge to be asserted without creating a new language – we can just use semantic relations derived from English word senses.

In order to represent objects and their properties, it is necessary to handle features or properties (*the book is red*), properties with values (*the dog is 3 feet tall.*), and binary relations between objects (*the dog is on the table*) and n-ary relations between objects (*the dog is 3 feet right of the table*). These are all handled uniformly as applied semantic relations.

In addition to normal arguments, every semantic relation has a **SELF** argument that can be filled with a specialized concept that reifies the relation-as-object itself. So, for example, the semantic relation EAT takes arguments for INGESTOR and FOOD among others. An additional argument SELF represents the action of EATING itself. With vignettes (Section 2.4), the SELF argument represents the combined object containing all the elements. The notion of having a SELF argument for all semantic relations was jointly conceived of in a discussion with Daniel Bauer.

2.3.1.4 Lexical Items

Lexical items represent words and word senses. The lexical item consists of the word's inflections, part of speech, valence patterns (for verbs), and lexical properties (such as whether the word is a mass nouns and takes the singular form). Nouns are associated with concepts, while verbs, adjectives, adverbs, and prepositions are relational in nature and associated with semantic relations. Note that event nouns, such as deverbals, are concepts that play a specific role (often SELF) in a relation. For example, *decision* is the SELF in *decide*. Other functionally-oriented nouns are associated with different relational roles. For example, *invader* is a concept with an agent role in the semantic relation for *invade*.

2.3.1.5 Semantic Translation Rules

Semantic translation rules convert one meaning representation to another while preserving aspects of meaning. They can be used to decompose one level of meaning to another more concrete meaning and also to translate between specific semantic relations and more general parameterized ones. All translation rules take a set of semantic relations as inputs, and output a modified set of relations. In doing so they can check the semantic type of any argument for any relation. All rules can query the knowledge base while processing their inputs. This allows the same knowledge to be shared among many rules. Semantic translation rules are used as follows:

- To normalize semantics (e.g. to normalize gradable attributes)

  (blue :entity entity) ⇒ (color-of :entity entity :value <rgb-value-for-blue>)
- To fill in defaults for semantic relations. For example:

  (write :agent person :patient letter) ⇒

  (write :agent person :patient letter :instrument pencil)
- To decompose function-oriented semantic relations into vignette relations or directly to graphical

primitives based on based on argument patterns. (See Section 6.9)

(chase.r :agent ?agent :cotheme ?cotheme :path ?path) ⇒

(cotheme-on-path.vr :agent ?agent :cotheme ?cotheme :path ?path)

- To decompose vignette relations to graphical or ontological relations. (See Section 6.9)

  (cotheme-on-path.vr :agent ?agent :cotheme ?cotheme :path ?path) ⇒

  ((gfx.behind.r :figure ?agent :ground ?cotheme) ...)

- To convert syntactic dependencies to semantic relations. Note that while lexical items are not concepts or relations themselves, they are semantic nodes that are linked to concepts and relations. EG:

  ((photograph :dep of.prep) (of.prep :dep lily)) ⇒

  (of.prep :figure photograph :ground lily) ⇒

  (representing :representation photograph :referent lily)

See Section 6.4 for a description of the internal structure of these rules.

2.3.1.6 Naming Conventions

By convention, when concepts represent entities, they are usually denoted as {NODE-NAME}.N. For example TIGER.N represents the concept associated with the lexical item *tiger*. Likwise, concepts that represent relations are denoted as {NODE-NAME}.R. For example, GFX.ON-TOP-SURFACE.R represents a graphical relation that asserts that one of its arguments is on top of the other.

Note: Asserted relations (whether from the knowledge base or derived from input sentences) can be applied not only to full-fledged functional or lexicalized concepts, but also to specific instances or anonymous unlexicalized subtypes. We will denote such arguments with numeric indices. For example: BOOK.N.1 represents a particular instance or anonymous subtype of a book.

2.3.2 Fleshing Out the Ontology

We now define the following semantic relations and concept types. These have no special ontological status but are basic and used throughout.

Mereological Primitives. In order to capture notions such as parthood and group membership, we introduce a set of *mereological primitives*. These are relations that define their arguments as parts, wholes, groups, elements, and regions. They can be a) invoked by textual descriptions, b) generated by semantic

translations and decompositions, and c) employed in assertions in the knowledge base. In all cases, it is crucial to meaning representation to distinguish parts, subregions, or sub-elements from the whole. Note: we use $\sqsubseteq$ below to designate subsumption, that one concept is a subtype of another. The relation types are:

- WE.PART-OF.R (whole, part): This asserts that PART is part of WHOLE. The part can be a physical part (such as a *door* on a house), a temporal part (such as the *beginning* of a discussion), or a conceptual part (such as a *flaw* in an argument).
- WE.MEMBER-OF.R (group, element) where GROUP $\sqsubseteq$ GROUP.N: This asserts that ELEMENT is a member of GROUP. This relation differs from WE.PART-OF.R in that the membership is optional and can be *ad hoc* (for example, as specified by text).
- WE.GROUP.R (group, cardinality, element-type) where GROUP $\sqsubseteq$ GROUP.N: This asserts that a specific group is constrained to have a certain number of elements of a certain type. For example, the knowledge base asserts that musical quartets consist of 4 human members:

  (we.group.r :group musical-quartet.n :cardinality 4 :element-type human.n)

  We note that there can be more specific group types to represent different parameterized configuations of elements such as stacks, rows, and columns.
- WE.MADE-OF.R (entity, substance) where SUBSTANCE $\sqsubseteq$ SUBSTANCE.N: This asserts that an *entity* is made out of a *substance*.
- WE.REGION-OF.R (whole, region, area-arguments): This relation asserts that a REGION is defined from the WHOLE by the given AREA-ARGUMENTS. Regions are parts that are defined by a specified metric. There could potentially be temporally defined regions too.

Examples of how these mereological primitives are used are discussed throughout. In particular, they come into play in the discussion of entity selection in Section 6.4.3.

Graphical Primitives. Every scene can be represented by a set of primitive graphical relations applied to a set of objects. The objects are all concepts representing graphical objects. And the graphical primitives are responsible for positioning those objects in space and changing any other graphical properties. These relations are described in Chapter 3 and listed in Section 3.4.

Affordances and Functional Properties. Relations and concepts are used to associate functional and spatial properties with objects. Affordances are described in detail in Section 3.7. Functional properities are properties that specify aspects of how an object is used or where it is placed by default (e.g. whether it

is a ground element, a fixture, a hand-tool, or wall item). They can also, for example, specify the manner in which an object is scaled (e.g. uniformly along all axes, or along a particular length axis). Functional properties can be asserted as semantic relations applied directly to specific objects. But in other cases, it is more convenient that they be applied to higher-level concepts (that embody that property) that any other given object can inherit from. For example, the property of being a wall item can be applied to a high level WALL-ITEM.N concept. Specific sub-types such as WALL-CLOCK.N can then inherit the functional property by the IS-A hierarchy.

Meta Relations. Meta-relations are semantic relations whose arguments are other relations or combinations of relations and concepts. These are discussed in Section 4.3.

2.3.3 Asserted and Computed Knowledge

VigNet encodes knowledge both implicitly through its ontology of concepts and explicitly by a wide variety of assertions. Using its well-defined ontology of concepts and semantic relations it is able to make assertions in a simple, precise manner (by applying relations to concepts). It can make assertions both about general purpose concepts (e.g. elephants) as well as assertions about specific instances (e.g. a particular 3D model of an elephant). The knowledge base has approximately 200,000 asserted relations. Some examples:

- emblems (e.g. that a flag image is an emblem for a particular country)
- containment (e.g. that a beer bottle typically holds beer)
- representation (e.g. that a particular symbolic moon 3D object represents the moon).
- residence (e.g. that an Canadian is a resident of Canada)
- habitat (e.g. that parrots live in the tropics)
- selling (e.g. that a coffee stand sells coffee)
- style (e.g. that a stagecoach is old-fashioned)
- substance (e.g. that a rock is made of stone)
- age (e.g. that a child is under 12 years of age)
- groupings (e.g. that a herd is a group of land animals)
- parts (e.g. that a particular 3D object has a tail)

Assertions about concepts can be made to concepts at any level of the hierarchy. Every asserted relation has auxilliary arguments denoting whether the assertion:

a) is true by definition

b) is true with some level of certainty

c) has been induced upwards from subnodes or

d) has been deduced downwards to this concept from supernodes

When values are percolated up, the computed value on a higher-level concept will contain a range of values. If the values are concepts, those concepts will be merged into their common ancestor. For example, if (SUBSTANCE-OF :entity *penny* :substance *copper*) and (SUBSTANCE-OF :entity *dime* :substance *silver*), then the computed value for the substance of a *coin* will be *metal*. For numeric values, rather than having a single value, the parent concept will have a RANGE value.

2.4 Vignettes – Where Form Meets Function

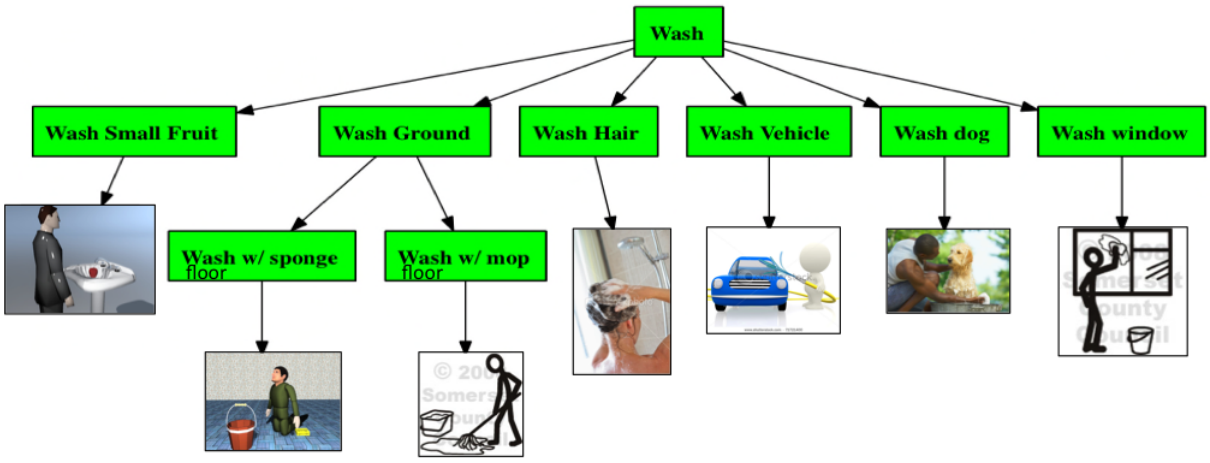


Figure 2.3: Variations of "wash" needed to be captured by vignettes

Vignettes ([Coyne *et al.*, 2010a]) are recognizable configurations of objects in the world. They can represent actions, locations, or compound obects. For example, a vignette invoked for *person washes apple* might involve the person holding the apple and standing in front of a sink, whereas *person washes car* would involve the person facing a car on a driveway and holding a hose. These decompositions are made

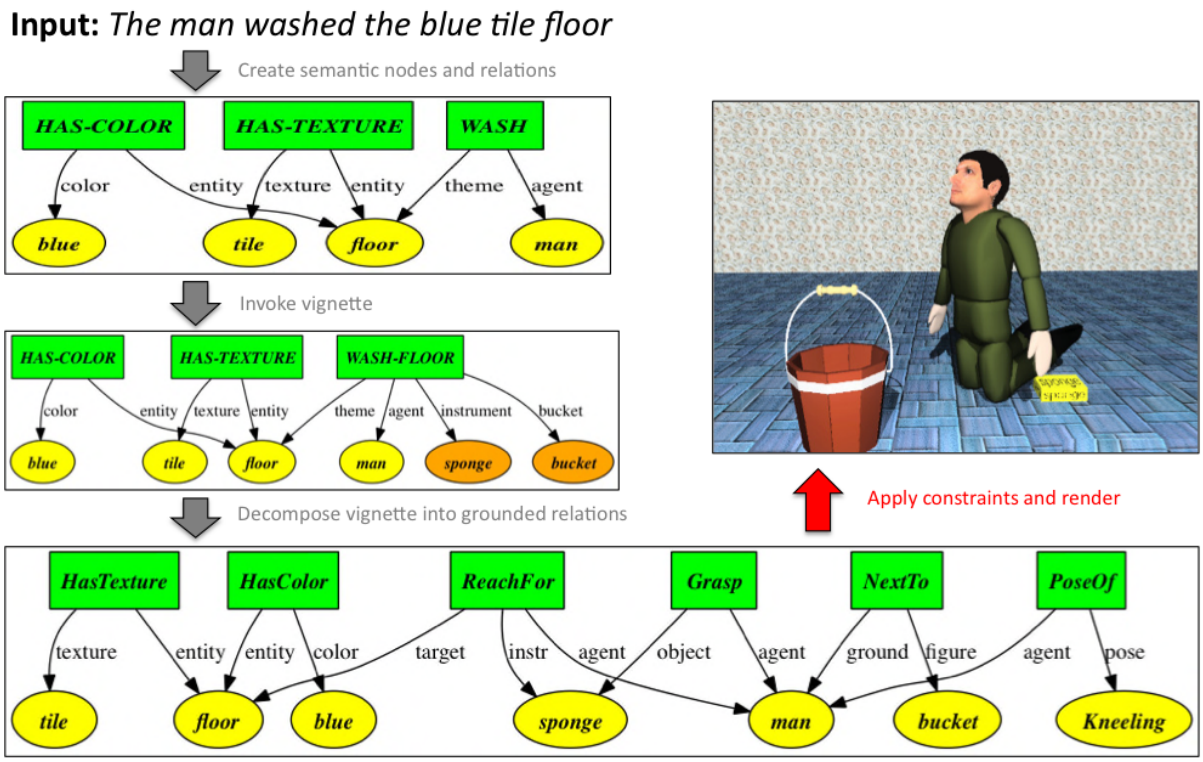


Figure 2.4: Decomposition of *wash the floor* into spatial relations via vignettes

with semantic translation rules. The semantic arguments of a vignette relation represent the objects involved in the visual scene associated with that vignette, and the resulting decomposition specifies how the objects are graphically realized.

More technically, a vignette consists of a vignette semantic relation and its associated SELF entity (concept). The vignette entity is the SELF argument associated with the vignette relation just as an event is the SELF of a high-level action semantic relation. The Vignette SELF entity is the actual concrete instantiation that spatially encompasses all the vignette arguments. Vignette relations are decomposed by semantic translation rules into to a set of primitive graphical relations on those same arguments.

This mechanism covers several different cases:

2.4.1 Actions

We conceptually freeze the action in time, much as in a comic book panel, and represent it in a vignette with a set of objects, spatial relations between those objects, and poses and other graphical manifestations relevant for that action. Action vignettes will typically be specialized so that the applicability of different vignettes with the same parent semantic relation will depend on the values of the semantic arguments of the parent. For example, Figure 2.3 shows some of the possible ways that vignettes for *wash* can vary based on argument type.

To translate a high-level semantic representation into a concrete scene, additional semantic arguments are often required to play auxilliary roles. These objects become semantic arguments of the vignette. For example, in washing the floor, the bucket would be a new role introduced by the vignette that is not found in *wash* itself. See Figure 2.4.

High-level relations are mapped into vignettes and vignettes in turn are decomposed into a set of specific, grounded relations that can be directly used to compose a scene. For example, to convert high-level semantics to vignettes, we can use patterns like the following to decompose *wash hair*:

```
INPUT: ((fn.removing.wash.r (:agent ?agent) (:theme ?theme))
        (we.part-of.r (:whole ?agent) (:part ?theme)))
TYPES: ((?theme (or human-head.n hair.n))
        (?location shower-stall.n))
OUTPUT: ((we.manipulate-patient-on-body.vg (:agent ?agent) (:patient ?theme))
          (we.in-3D-enclosure.r (:figure ?agent) (:ground ?location))))
```

See Section 6.9 for an example of a simple verb sentence being decomposed to semantics to graphical representation to scene.

2.4.2 Composite Object Vignettes

Even an ordinary physical object can be described by a vignette if it can be constructed out of separate elements and decomposed to spatial relations between those elements. For example, a *stop sign* can be defined as a two foot wide red hexagonal metal sheet displaying the word “STOP” positioned on the top of a 6 foot high post. Likewise rooms can be represented by walls, floor, and ceiling in a particular spatial relation.

In practice, we use composite object vignettes to represent "cutout characters" for a couple hundred celebrities and historical figures. In these composite object vignettes, the vignette consists of a 2D cutout head placed on a 3D cutout body using the graphical primitive GFX.CONNECT-ATTACHMENT-POINTS.R. See Figure 2.5. Individual cutout character vignettes (e.g. Thomas Edison) inherit from the general HUMAN-AS-HEAD-ON-BODY.VR. See Section 6.4.6 for translation rule. Note that a translation rule for each individual character is not needed, since they all inherit from the base vignette. But the option remains to define a specific translation rule for any individual vignette.

a) *John Travolta is next to Jon Stewart. The camel is 10 feet behind jon. The cactus is 3 feet left of the camel. The camel is facing southwest. A yellow light is above the cactus. A cyan light is above jon. Michael Jackson is in the camel.*

b) My small giraffe is in the maroon room. The Matisse elephant is 4 feet left of the room. The left wall is 3 feet tall. The room is 20 feet wide and 20 feet tall. It is 30 feet deep. The floor has a tile texture. The texture is 4 feet wide. The huge potted plant is above the left wall.

Figure 2.5: Composite object vignettes: (a) Characters: head, body. (b) Room: walls, floor, ceiling.

Composite object vignettes can be used for a wide variety of objects. For example, given the appropriate graphical primitives, a detective vignette might involve a character wearing a tweed cap and trench coat and holding a magnifying glass.

2.4.3 Locations

Location vignettes are a special case of composite object vignettes. The vignette relation expresses constraints between a set of objects characteristic for the given location, and the vignette object is the location itself. The semantic arguments of location vignettes include these constituent objects. A location vignette

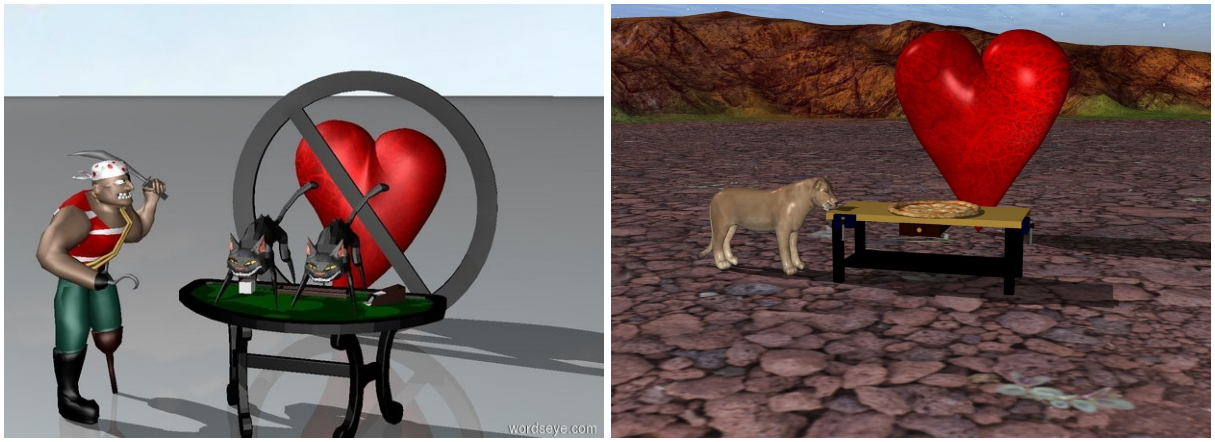
can represent rooms of different types, a garden, a shooting range, a dock area, and so on. The basic room vignette consists of a floor, a ceiling and walls arranged and oriented to create an open room space. See Figure 2.5. More specialized rooms can be built on top of that. For example, a simple living room vignette might consist of a room with a couch AGAINST the left wall, a coffee table IN-FRONT-OF the couch, and a fireplace EMBEDDED-IN the far wall.

2.4.4 Iconic Vignettes

Vigettes can be defined to translate figurative or metaphorical language. For example, negation can be depicted using a not sign. Some approaches for depicting abstract or figurative language that we have explored earlier are described in [Coyne and Sproat, 2001]. Section 3.3.1 also discusses this general issue.

As an example, the decomposition rule for *Steve hates cartoons* is listed below. In this case the decomposition is triggered directly from the semantic relation (for *hate*). It it is a matter of convenience whether to reify the decomposition as an actual vignette or to translate directly to primitives from the higher-level semantic relation. Instantiating a vignette allows easier re-use by other rules and has the benefit of a SELF entity that allows the vignette to be moved in space and referenced by other text. (Figure 2.6) shows examples of iconic vignettes.

```
(:input ((?rel :experiencer ?agent :content ?ob))
 :types ((?rel fn.experiencer-focus.hate.r)
         (?heart heart-shape.n)
         (?not-sign not-sign-symbol.n)
         (?black black-color.n)
         (?red red-color.n)
         (?pink pink-color.n)
         (?table table.n))
 :output ((gfx.left-of.r :figure ?agent :ground ?table)
          (gfx.3d-size-of.r :entity ?heart :real-world-size 6.5)
          (gfx.3d-size-of.r :entity ?not-sign :real-world-size 8)
          (gfx.color-of.r :entity ?not-sign :value ?black)
          (gfx.color-of.r :entity ?heart :value ?red)
          (gfx.fit-on-top-surface.r :theme ?ob :entity ?table)
          (gfx.behind.r :figure ?not-sign :ground ?table :value 5)
          (gfx.behind.r :figure ?heart :ground ?not-sign)
          (gfx.emotion-token.r :entity ?agent :value "happy")
          (gfx.orientation-toward.r :figure ?agent :ground ?table)))
```



a) *steve hates cartoons* *Jenny loves food.*

Figure 2.6: Iconic vignettes

2.4.5 Abstract Vignettes – Exploiting Graphical Regularity

Abstract vignettes are a set of vignette templates that leverage the fact that there is great regularity in visual scenes. For example, there is not much structural difference between using a knife and using a pencil – both involve the application of a small INSTRUMENT to a PATIENT on a WORK SURFACE. As such, they share the same *abstract vignette*, with different argument values (i.e. *pencil* versus *knife*). The actual details of how the instrument is held is captured by the usage pose for that instrument which in turn relies on the affordances of that object.

In Table 2.1 we list a set of 42 abstract vignettes that we are using (along with graphical primitives) to assign vignettes to a set of verb argument corpus. See Section 5.6.4 for further discussion.

2.4.6 Standin Object Vignettes

In order to compose a scene, we need to know salient parts of objects. Since actual 3D objects have arbitrarily complex 3D polygonal meshes, we create a stand-in object for each object in the library. These stand-in objects consist of several parts, including a bounding box representing the full size of the object and optionally, the base, the top surfaces, interior surfaces and so on. These parts, in fact, represent the various spatial affordances the object provides. They can be thought of as its external interface to the world.

Abstract Vignette	Description
GRASP-TOUCH-PATIENT-ON-SELF	buttoning shirt
GRASP-TOUCH-PATIENT-ON-WORK-SURFACE	example: cutting something on a table
POSITION-THEME-MANUALLY	putting a picture on a wall
POSITION-THEME-WITH-INSTR	roast the marshmallow
APPLY-INSTR-TO-FIXTURE	paint the ceiling
APPLY-INSTR-TO-HELD-PATIENT	writing on a notepad
APPLY-INSTR-USING-WORKSURFACE	cutting carrots
USE-UNDIRECTED-INSTR	play the violin. talk on the phone.
APPLY-MACHINE-TO-PATIENT	boil the potatoes
INTERACT-WITH-MACHINE	open door. buy candy from vending machine. type on computer.
LOOK-AT-TARGET	glance across the room
LOOK-AT-WITH-INSTR	Take a picture of friends
GESTURE-AT-TARGET	nod towards the door
HURL-PROJECTILE-AT-TARGET	throw the stone
AIM-INSTR-AT-TARGET	shoot the rifle
AIM-MACHINE-AT-TARGET	shoot the cannon
HUMAN-INTERACTION	
COMMUNICATION-SIMPLE	
COMMUNICATION-ABOUT	
COMMUNICATION-ABOUT-WITH-INSTR	
SELF-MOTION	swim across the lake
PUSH-OR-PULL-FIXTURE	push the wheelbarrow
USE-VEHICLE	ride the horse
OPERATE-VEHICLE	
OPERATE-MACHINE	
WEAR	wear A STRAW HAT
SURFACE-ATTRIBUTE	
LIGHT-PROPERTIES	
ENVIRONMENT	Changing environment and time of day
ENTITY-SIZE-SHAPE	
ENTITY-STATE	
INTERACT-WITH-HUMAN	
STANCE-AT-LOCATION	
THOUGHT-SPEECH-BUBBLE	think about home
HUMAN-CO-INTERACTION	
COTHEME	
COTHEME-ON-PATH	
COTHEME-ON-PATH-TO-GOAL	
COTHEME-ON-PATH-FROM-SOURCE	
SELF-MOTION-ON-PATH	
SELF-MOTION-ON-PATH-TO-GOAL	
SELF-MOTION-ON-PATH-FROM-SOURCE	

Table 2.1: Abstract vignettes

All other geometric data is omitted for the stand-in objects, and they can be considered to be abstracted forms for each object. It is these standin object vignettes that are used by the system to compose a scene. When rendering, the actual full polygonal 3D object is swapped in.

2.4.7 Pre-Fabricated Vignettes

Many real-world objects do not correspond to lexical items but are subtypes of the concepts associated with those lexical items. We call these *sublexical entities* – for example, there is no lexical item for a *Japanese-style tea cup*, but there can be a concept for it and even a 3D object for it.

Other sub-lexical entities and real-world objects are *compound entities* – combinations of more than one object. Both can be represented by *pre-fabricated vignettes*. For example, one such compound entity in our 3D library is a goat head mounted on a piece of wood. This object is represented by a vignette with two elements (GHEAD, GWOOD) that represent the goat head and the piece of wood. The vignette decomposes into:

(gfx.on-front-surface.r :figure ?ghead :ground ?gwood)
(isa.r :sub ?ghead :super head.n)
(isa.r :sub ?gwood :super piece.n)
(part-of.r :whole ?ghead :part goat.n)
(gfx.made-of.r :entity ?gwood :substance :wood.n)

2.5 VigNet Knowledge Base

VigNet is a resource that contains the lexical, semantic, and contextual information needed to ground semantics into 3D scenes. The knowledge represented in VigNet is formulated using concepts, semantic relations, lexicon, graphical object vignettes, assertions, and translation rules as described in above.

VigNet currently contains about 15,000 nouns representing the objects in our 3D library as well as related nouns. The resulting noun lexicon is defined with multiple inheritance to allow the same lexical item to be classified in several ways. The resource also includes about 10,000 verbs, adjectives, adverbs, nouns, and their associated concepts and relations seeded from FrameNet. In addition, the resource contains a knowledge base comprised of a large number of semantic relations applied to concepts. For example, VigNet encodes the fact that a snowball is made of snow with the assertion (SUBSTANCE-OF.R :ENTITY SNOWBALL.N :SUBSTANCE SNOW.N). Finally, the resource contains translation rules for vignette decomposition and other semantic translations.

VigNet contains a number of low-level properties assigned to objects. These include the relations SUBSTANCE-OF, PART-OF, COLOR-OF, SHAPE-OF and SIZE-OF. For convenience we have computed these for

all concepts by inducing upwards and deducing downwards through the hierarchy (as described in Section 2.3.3). These properties are of potential interest for automatically describing objects and attributes detected in computer vision [Farhadi *et al.*, 2009].

VigNet will be made publicly available at http://www.cs.columbia.edu/~coyne.

2.6 Comparison of Semantic Representations

In this section we compare vignette semantics to Pustejovsky's qualia structures. We then briefly extend the comparison to Jackendoff's Lexical Conceptual Structures and the visual valences of Ma's CONFUCIUS system.

As described in Section 2.2.1, the QUALIA structure includes a number of sub-structures that contain semantic information. These all access entities defined in the ARGUMENT and EVENT structures defined by the overall lexical entry. Within the QUALIA structure, the TELIC structure specifies what the given entity is used for. For example the qualia for *knife* would specify in its TELIC structure that it is used for cutting. The CONSTITUTIVE structure would specify the parts of the knife and the materials that it is made out of. In contrast to vignettes, these are just elements that can be referenced and do not include the graphical relations between the parts.

As such, vignettes can be thought of, in part, as an elaboration of the CONSTITUTIVE structure, where for any object a set of relations between the elements is asserted that specify how to physically represent that object. And crucially, vignettes also play this role for actions where they recast function into form. However, as noted, the argument roles for vignettes, unlike qualia structures, are visual roles related to how the action or object is physically structured as opposed to what it functionally or causally conveys.

So while both qualia structures and vignettes decompose meaning into a set of relations that can optionally reference explicit lexical arguments as well as implicit entities, they are quite different in their philosophy, in what they accomplish, and in their formalisms and mechanics. We summarize the differences as follows:

- VigNet spreads knowledge between the vignette itself, general knowledge about concepts that can be shared by any vignette, and translation rules that allow decomposition from semantic relations. As a result, vignettes for using a knife are not keyed on the knife, but instead on the actions of *cutting* or *stabbing* which happen to reference knives and other implements of similar types. This al-

lows relational decomposition to be concisely defined independently from object types and properties. Similarly, concepts in the ontology (affordances, for example) can be leveraged by decomposition to vignettes.

- The type of translation produced by qualia structures specifies what the word means at a functional level (e.g. that *giving* implies change of *possession*) but not how it is physically or graphically realized. Vignettes, in contrast, construe meaning graphically. A vignette for an action bridges between function and form by casting the action as a configuration of graphical objects that visually represents its structure. As a result, vignettes can decompose an event in the same manner as it decomposes a compound object (such as a living room) into relations between its constituent parts.

- Vignettes allow similar functional meaning with different combinations of arguments to take on very different graphical meaning. For example, *washing one's hair* is graphically much different than *washing a car* and hence is represented by a different vignette, even though the functional meaning of *wash* is the same for both. For vignettes, combinations of arguments determine how the high-level semantics should be decomposed to a scene and thus (according to our perspective) to a grounded meaning. Qualia structures, in contrast, decompose meaning for the lexical item itself, not combinations of arguments.

  One consequence of this is that just as there are many types of *chairs* with different part structures (e.g. with or without side arms, with or without cushion, etc.), there are also many variations of *wash* involving different implements, different poses, different default settings, and so on. The insight behind vignettes is that both actions and entities can have varied graphical instantiations and that more abstract types are encoded in lexical items that can reference those entities. Vignettes are those entities. In vignette semantics, the graphical structure is the ultimate arbiter of the meaning.

- Vignette semantics is based on a lexical resource (VigNet) that allows it to be implemented and tested. It has a well-defined set of graphical and logical primitives as well as specific concepts and semantic relations (inclulding vignettes) to apply to those concepts.

- Qualia structures, and hence semantic information, are applied to lexical items while in VigNet semantic relations can be applied to concepts that correspond to lexical items as well as to non-lexicalized concepts. For example, some notebooks have spiral binding while others have sewn binding. These

are conceptual differences that have no lexical counterpart. Likewise, any number of vignettes can be associated with a given verb, given different sets of arguments. In VigNet, that information can be properly applied to any concept, not just to lexical items.

The Generative Lexicon, through qualia structures, can explain how *enjoy the book* and *enjoy the movie* imply different actions (reading or watching) depending on argument (book versus movie). This is fine as far as it goes, but the goal of vignettes is to let those choices be driven at the level of specific objects, actions, or vignettes, rather than just for lexicalized types.

These same points apply in almost the same fashion to Jackendoff's Lexical Conceptual Structure where semantic decomposition is also functional in nature. Object geometry and spatial affordances are not represented in LCS and there is no sense in which a dozen varieties of *wash* would have separate meaning since their functional argument structures would all just specify the idea of removing dirt with a liquid. The *manner* of doing that (which includes radically different graphical realizations) is not encoded or considered to be part of the meaning.

Ma's visual valences (as discussed in Section 1.3.5) bear some resemblance to vignettes in that they both decompose high-level semantics into graphical constraints, and they both recognize differences between visual roles and syntactic or semantic ones. Visual valences, however, are defined on top of verb senses and their thematic roles rather than constituting independent standalone structures. For example, *writing a letter* is given as having three visual valences: the writer, the letter, and the pen (instrument) corresponding to standard thematic roles, where the instrument is made obligatory [Ma and Mc Kevitt, 2004b]. Vignettes, in contrast, have arbitrary sets of graphical roles and are structured in a separate hierarchy and can take advantage of visual regularities, which are unlike lexical or functional ones. A vignette for writing a letter might also involve a desk, a chair, and an ink well, for example. These do not correspond to verb-related thematic roles. As a result, with vignette semantics, the same verb sense can be tied to an arbitrary number of very different vignettes based on different levels of abstraction or realism or different supplied semantic arguments (e.g. *washing hair* versus *washing a car*). And conversely very different verbs can share the same vignettes when their graphical realizations are similar.

One other key difference is that visual valences only apply to actions (in order to produce animations), whereas vignettes can apply to compound objects or any concept. The SELF argument in vignettes can represent actions and compound objects, since vignettes serve to convert actions into configurations of objects. In addition, vignette semantics leverages a unified relational representational system, ontology, and

knowledge base (VigNet) where all meaning (lexical, functional, graphical, world knowledge) is represented as relations and transformed in a uniform manner. This constitutes a more expressive lexical and graphical theory that allows it to translate semantics generally (e.g. to fill in default arguments or to interpret gradable attributes); to decompose lexical semantic ambiguity based on graphical properties; and to decompose high-level semantics to graphical primitives and objects via vignettes.

2.7 Conclusion

In order to represent language graphically we need a way to represent a) the meaning of the input text, b) the relations between graphical objects and the properties of those objects in the output scene, c) world knowledge and conceptual knowledge d) knowledge about words and how they connect to one another and to concepts e) ways to specify that one semantic representation can be translated into one another f) representations of how semantic relations can be realized in a graphical or real-world manner. VigNet provides all this in a unified framework, allowing meaning to be processed at all levels and to ultimately be decomposed to grounded graphical representations. In Section 5 we describe the methods we have used and plan to use for populating VigNet.

Chapter 3

Graphics: Semantics and Primitives

3.1 Introduction

This chapter is centered around what we call *graphical semantics* – the knowledge and meaning implicit in real-world objects and scenes that allow us to make sense of them – and how that can be modeled in a text-to-scene system. We start by discussing some background work in spatial cognition. We then make some observations about how some of the choices available in representing actions and figurative language, as well as how other factors in the rendering process (such camera position), can affect the visual interpretation of the scene.

The bulk of this chapter is focused on describing the methods we use to actually translate semantics into graphical relations. Rather than treating objects as "dumb" sets of polygonal data, we give them semantics by representing their functional properties and the *affordances* (see Section 3.7) they provide to make themselves useful to the world. We discuss how affordances and other object properties are used to decompose vignettes and other semantic constructs to a set of primitive graphical relations. All the affordances, spatial and functional properties, and graphical primitives are expressed using the same representational framework of concepts and semantic relations provided by VigNet.

Our goal is not only to specify the properties used by WordsEye currently and how they are represented in VigNet, but also to highlight some important issues that a text-to-scene system should handle.

3.2 Background – Spatial Cognition

There is a long tradition in cognitive science and cognitive linguistics of spatial mental models – the notion that our thoughts and understanding of language are mapped to spatial representations. These models can be applied not only to spatially oriented language but also to metaphorical expressions such as "the temperature is rising" [Lakoff, 1987]. Mandler [Mandler, 2004] shows how infants understand spatial notions such as attachment and blocked paths at a very early, pre-linguistic stage of development.

Empirical evidence for spatial mental models in interpreting language has been demonstrated in controlled perception experiments by Mani [Mani and Johnson-Laird, 1982] and others. Feist [Feist and Gentner, 2003] and Andonova [Andonova *et al.*, 2010] have shown that object properties such as function, animacy, shape all affect spatial relation perception. Even non-concrete verbs can have a spatial connotation [Richardson *et al.*, 2001], and there is evidence that spatial models play a role not only in perception but in generation [Lockwood *et al.*, 2006]. Many cross-linguistic studies have been done comparing spatial language. Choi and Bowerman [Choi and Bowerman, 1991], for example, have compared the differences in spatial prepositions in English and Korean. Levinson [Levinson, 1996] has compared the linguistic aspects of spatial frames of reference among cultures. Many cultures conceptualize directions in terms of body parts. For example 47% of oceanic languages associate the term for *head* to express the concept of *up* [Heine, 1997] (page 43). These features vary from culture to culture.

Different approaches abound: A theoretical framework for the spatial interpretation of prepositions is given by Herskovits [Herskovits, 1986]. ISO-space is an standardized scheme to annotate spatial entities and relations in textual corpora [Pustejovsky *et al.*, 2011]. Region Connected Calculus (RCC) is a formal theory for qualitative spatial representation and reasoning by topologically dividing space based on disjoint, abutting and overlapping regions [Cohn and Hazarika, 2001].

Our work is in the spirit of some of the above but grounded in the requirements of a text-to-scene system and the 3D objects in our library. Our methodology is to first ask what properties of the object are relevant to participate in spatial and other graphical relations. From this we define a set of graphical primitives using those properties. We generalize some of the properties into a set of affordances and other properties on objects that are then endoded into VigNet.

3.3 Interpreting scenes

3.3.1 Interpreting Temporal and Figurative Language

As noted in Section 1.1.2, our goal is to create a static scene, not an animation. This raises questions of how to best depict temporal descriptions in a single frame. Movement and actions can be can often be suggested by putting a character in a characteristic pose or objects in a certain configuration as is often done in comic books and other illustrations.

Most actions have manner-related components and state change components. For example, *John broke the bottle* could show a bottle just below John's hands on its way to the floor. Or John in a throwing pose with bottle in his hand. Or it could show shards of glass on the floor with John looking down at it. There are many choices related to showing the preconditions, the poses suggesting an action itself, or the post-conditions (in this case the shards of glass).

Also, as we discussed in our earlier work [Coyne and Sproat, 2001], many sentences include abstractions or describe non-physical properties and relations, and consequently they cannot be directly depicted. So we described a variety of techniques that could be used to broaden the scope of what is depictable. Some of these are also supported in the current version of WordsEye. And see Section 2.4.4 for a description of how others are being implemented using iconic vignettes.

- Textualization: using actual 3D words in scenes to express idea
- Emblematization (e.g. using a flag to express the concept *patriotism*)
- Characterization (e.g. use a person wearing a baseball cap to depict *baseball player*)
- Conventional icons (e.g. using thought bubbles, motion trails, or not signs)
- Literalization (e.g. *the temperature is high* ⇒ thermometer in the air)
- Personification (e.g. *danger approached* ⇒ put "danger" or an emblem signifying danger on an approaching person. In some cases, we use a specific 3D object for personification. For example, a grim reaper 3D model can personify "Death.")
- Degeneralization (e.g. depicting *life* with an animal)

We show an example in Section 2.4.4 of how vignettes can be used in this manner.

3.3.2 Other Factors in Scene Interpretation

While our goal is to depict 3D scenes from natural language, the actual "end product" is a rendering of that scene. 3D graphics rendering can range from photo-realistic to stylized (e.g. cartoon outlines) to abstract designs, depending on the types of "shaders" [Upstill, 1989] used by renderer, the 3D camera viewpoint, and the lighting. Scenes can be produced with language that look more like 2D designs or abstract 3D sculpture than 3D scenes. In addition, the incorporation of 2D imagery as textures can add 3D depth to a scene and change the visual interpretation of a scene. Normally 2D imagery is perceived as a texture in 3D space, but it can also be perceived as a "pre-rendered" part of the 3D space itself, or part of an abstract space that contains both 2D and 3D elements. See Figure 3.1 for examples produced on the system.

While the current system does not support the control of these types of effects directly from language (other than directly specifying the locations of lights, objects, and textures), it would be desirable to have a way to interpret language visually in terms of style. Such questions are beyond the scope of this thesis.

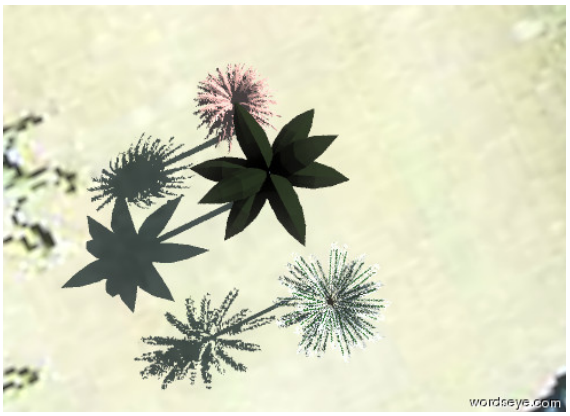
3.4 Graphical Primitives

In order to convert a semantic representation into a 3D scene, it is necessary to have a set of primitive relations that cover the desired range of graphical phenomena. What follows is a list of the basic categories of primitives we have identified. We include some primitives that are not currently handled by WordsEye due to limitations in its graphical capabilities. In particular, the system is currently unable to deform objects as needed to depict poses and facial expressions. We note, though, that earlier versions of the system had some amount of support for posable objects and facial deformations. We show examples produced at that time and plan, in the future, to support those again more fully.

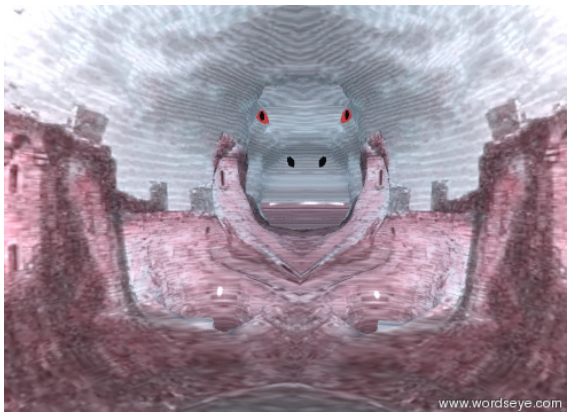
Materials and Surface Properties: RGB-VALUE-OF, HAS-TEXTURE, SATURATION, BRIGHTNESS, TRANSPARENCY, VISIBILITY, ROUGHNESS, REFLECTIVITY, SUBSTANCE-OF

Orientation: UPRIGHT, SUPINE, FACE-DOWN, FACE-UP, UPSIDE-DOWN, ORIENTATION-FROM, ORIENTATION-TOWARD, ORIENTATION-WITH, ORIENTATION-ALONG

Positions: BALLOON-ELEMENTS, EMBEDDED-IN, IN-CUP, STEM-IN-CUP, IN-3D-ENCLOSURE, IN-ENTANGLEMENT, UNDER-CANOPY, ON-SURFACE, ON-BOTTOM-SURFACE, ON-TOP-SURFACE, ON-FRONT-SURFACE, CONNECT-ATTACHMENT-POINTS, ON-ORIENTED-SURFACE, LEFT-OF, RIGHT-OF, IN-FRONT-OF,



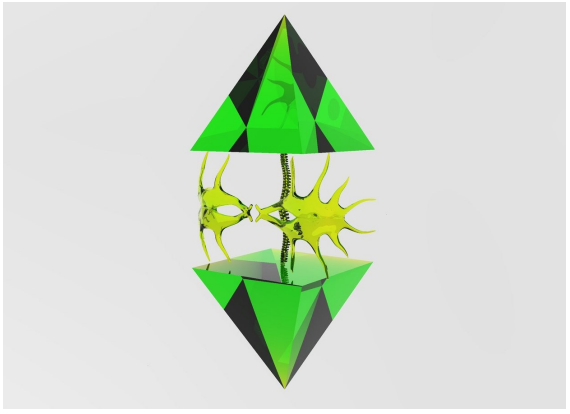
a) Looking from above: *Three palm trees are on the beach. The first palm tree is green....*



b) Looking from the inside (the hippo): *The pink castle hippo.*

c) Shadows and lack of depth: *The flat black man is in front of the huge foam wall. The small flat black tree is to the right of the man...*

d) Images adding depth: *The farmland wall is 10 feet wide and 6 feet tall. The ground has a straw texture. The rake is in front of the wall...*



e) Concrete to abstract: *A translucent chartreuse elk is 5.8 feet in the ground. A 5 feet tall translucent chartreuse pyramid is above the elk. A translucent chartreuse spinal cord is...*

f) Abstract to concrete: *A 1 feet tall 1st sphere is 8 feet above the ground. A 3 feet high and 1 feet wide cone is -0.8 feet in front of the 1st sphere. It leans 130 degrees to the front. ...*

Table 3.1: Effects of lighting, camera position, textures, and object parts on visual interpretation

BEHIND, NEAR, NEXT-TO, ABOVE, BELOW, LEVEL-WITH, POSITION-BETWEEN

Single axis positioning: ORTHOGONALLY-LEFT-OF, ORTHOGONALLY-RIGHT-OF, ORTHOGONALLY-IN-FRONT-OF, ORTHOGONALLY-BEHIND, ORTHOGONALLY-ABOVE, ORTHOGONALLY-BELOW, LATERALLY-CENTERED-X, LATERALLY-CENTERED-Y, LATERALLY-CENTERED-Z

Size Modification SIZE, 3D-SIZE-OF, HEIGHT-OF, DEPTH-OF, WIDTH-OF, FIT-DEPTH, FIT-WIDTH, FIT-HEIGHT, MATCH-DEPTH, MATCH-WIDTH, MATCH-HEIGHT, MATCH-TOP-SURFACE, MATCH-FRONT-SURFACE

Fit and position combos: FITTED-ON, FIT-ON-TOP-SURFACE, FIT-IN, FIT-BELOW

Poses and Expressions: FACIAL-EXPRESSION, GRASP-TOUCH-FIXTURE, HOLD, HOLD-NON-FIXTURE, IN-POSE, LOOK-AT, MOVE-JOINT-TO, POINT-AT, TOUCH-WITH-BODY-PART

Environmental Properties: TIME-OF-DAY, ATMOSPHERIC-PROPERTY

Type declarations: IS-CUTOUT, IS-PANORAMA-IMAGE, IS-OBJECT, IS-DECAL, IS-IMAGE

Geometric and Topological Deformations: HOLE, SLICE, CRUMPLE, BEND, INDENT, BLOAT, TWIST

In addition to these primitives, we also rely upon *mereological designators* to specify objects and their target regions or parts or affordances: ENTITY-SPECIFIER, PART-SPECIFIER, REGION-SPECIFIER, GROUPING-SPECIFIER. These designators interact with spatial affordances to resolve the actual region used by the graphical primitives.

3.5 Intrinsic Size, Orientation, Shape

It is necessary to know various intrinsic properties of objects in order for them to participate in spatial relations. For example, objects in scenes should be roughly the sizes one would expect to see in the real world. Similarly, if an object is said to face another object, we need to know where the front of the object is located.

3.5.1 Sizes

Objects in the world have intrinsic real-world size. These are represented in VigNet by a size relation applied to the object. Like other relations, the size relation can be assigned to any concept, whether a specific object

(e.g. a particular 3D model of an elephant) or a more general type (elephants in general). The default size of object, therefore, is derived by finding either its locally defined value or a value inherited from its parent nodes. Note, however, that an object's size can always be overridden by any specification supplied in the input text. Operations:

- Scale to exact size
- Scale by factor (from norm)
- Scale along axis: Axis can be XYZ (width | height | depth) OR relative axis (length)
- Scale to match another object's dimensions (in any axis)

In textual descriptions, the size of an object can be specified. The way that size is interpreted depends on features of the object in question. The relevant features (all represented as relations) are:

- SEGMENTED. (See section 3.5.2.)
- (for images) PORTRAYING versus INSTANTIATING. (See section 3.5.3.)

3.5.2 3D objects

We have identified three important ways that an object's properties can affect the resulting size and resulting aspect ratio of the object.

CONSTRAINED AXES: Consider the following two sentences:

- *The bench is 10 feet wide*
- *The pyramid is 100 feet wide*

In the first case, we would expect the bench to be roughly the same height and depth as usual, with just the width elongated. In the second case, we would expect the pyramid's height and depth to change along with its height. See figure 3.5.3 (a) and (b).

SEGMENTATION: This property of objects captures whether an object is stretched or duplicated when its size is changed. This latter class of *segmented* objects include fences, railroad trains, and sidewalks (tile-by-tile). See figure 3.5.3 (c) and (d).

LENGTH AXIS: The *length axis* is the axis along which the objects is scaled when it is specified as being *long* or having a certain length. Normally, the segmentation (if any) is along the length axis.

3.5.3 Sizes and Images

When images in a scene are used to represent other objects *pictorially* or figuratively, the actual size of the represented object is not maintained. For example, a picture on a billboard and a picture on a postage stamp may represent the same exact object, but the size will be very different. In these cases, the system will stretch the image to fill the area. A photo of a flower on a billboard or postage stamp will be very different sizes.

Images can also be used as *textures*, in which case their size should match the real-world size of the object they represent. For example an image of a patch of grass may be used to represent actual grass when mapped onto the ground plane. This can be described with sentences such as "the ground is grass" or "the ground has a grass texture" or "the grass texture is on the ground". In these cases, the size of the grass texture should be the same as it would appear in real world since it effectively becomes part of the object it is applied to.

To encode these distinctions, images come in several varieties. There are *pictorial images* (with no fixed size), *textures* (that map to specific real-world sizes), and *panorama and background* images (e.g. for the sky). Each of these image types have different properties that affect how they are scaled. A fourth type of image is a *cutout* image. This is usually a photo of a real-world object with an alpha (mask) channel around it. These are typically texture mapped onto flat rectangular 3D objects and used in the scene as though they were the actual object. Since they look flat from the side, they are typically used for background elements in 3D games. Like textures, these have a real-world size that is applied to the 3D object hosting the cutout. We use them for likenesses of real-world individuals. So, for example, "Abraham Lincoln is next to the table" will use the real world size of an Abraham Lincoln cutout as a stand-in 3D object of that same size. See section 2.4.2 for a description of cutout images used as vignettes.

WordsEye currently bases the default scaling of an image on the image type (texture, pictorial, panormama, or cutout). This, however, is not an inherent limitation since many objects (e.g. billboards and televisions) have an associated DISPLAY affordance (see section 3.7) that specifies how they typically display pictures. For example, one might consider an ambiguous case where a billboard with a grass texture could be intended not to be *made of* grass but to *portray* grass. In this case, the image could be scaled to fill the billboard rather than displayed in its real-world size. The scaling would be a function of both the image (whether it a texture or a pictorial representation) as well as the hosting object (whether it commonly displays an image).

Note also, that the system will change the size of image when explicitly specified in the text input. For example, [*The wall has a flower texture. The texture is 10 inches wide.*] will apply the texture to the wall but change of each tile of the texture to be 10 inches wide.

3.6 Orientation

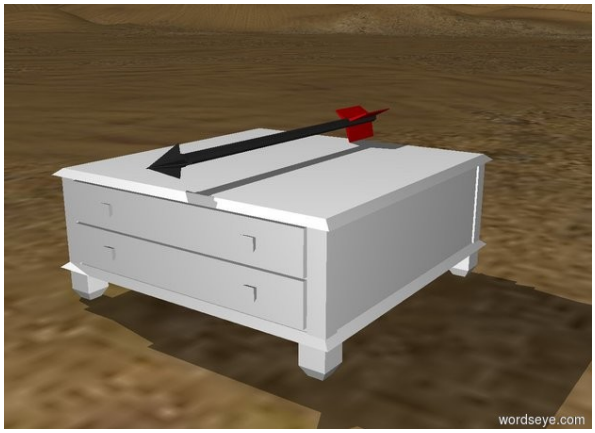
Objects can have an intrinsic orientation that depends on the overall object shape and symmetry. Their intrinsic orientation and their shape determine how they are displayed by default and arranged other objects. For example, objects with a well defined front should have that side facing forward. Other objects are rotated along other axes. For example, an envelope would be placed with its back on a table with its bottom edge facing forward (toward the person seated at the table).

3.7 Functional Parts and Affordances

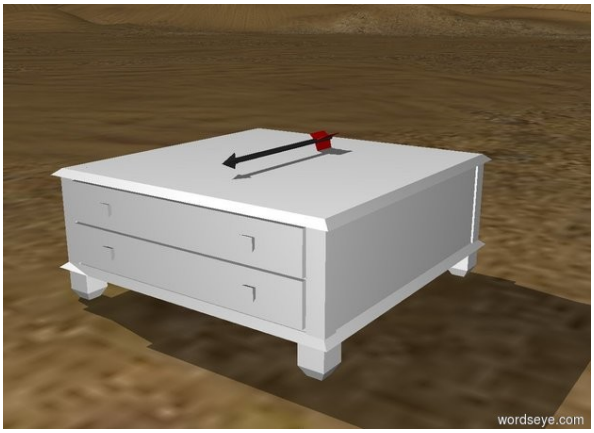
The concept of affordances [Gibson, 1977; Norman, 1988] has a long and influential history in the study of ergonomics and the psychological interpretation of the environment as well as in philosophy [Haugeland, 1995]. Affordances are traditionally considered to be the qualities of objects in the environment that allow people or animals to interact with them. For example, the door of a refrigerator provides an affordance that allows access to the interior of the refrigerator, and the handle on a door provides an affordance that allows the door to be grasped in order to open and close it. We take a slightly broader view and include as affordances any functional or physical property of an object that allows it to participate not only in actions with people but also in relations with other objects. For example, a chair usually has a *seat* area that is used for sitting but which also provides a location where an object could rest. Likewise, a cereal box provides an affordance as a container for the cereal, independently of how a person interacts with it. By inspecting all objects in our 3D library, we have identified and labeled (with Alex Klapheke) the following affordances:

Human Location: These represent different broad classes of locations where a human would stand.

- WALKING-AREA
- PATH
- WALKTHROUGH-OPENING



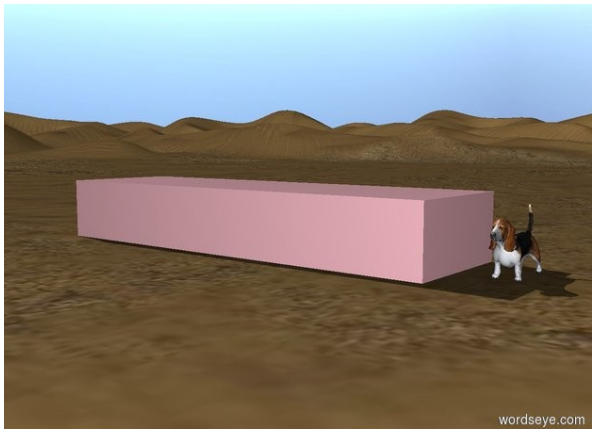
a) *A 4 foot long arrow is on the table.*



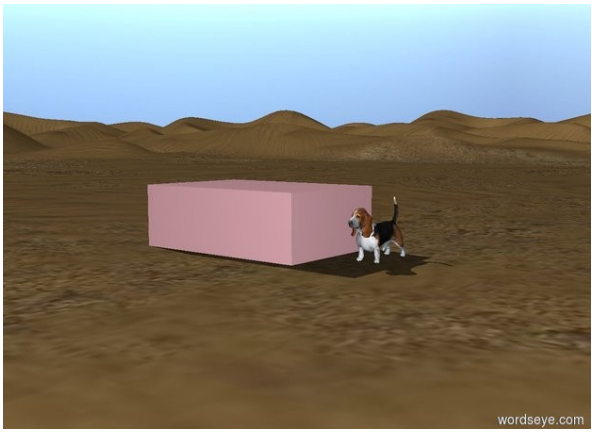
b) *A 2 foot long arrow is on the table.*

c) *A 30 foot long fence is in front of the house.*

d) *A 10 foot long fence is in front of the house.*



e) *A 14 foot foot slab is a foot left of the dog.*



f) *A 6 foot foot slab is a foot left of the dog.*

Figure 3.1: *Segmented* versus *scaled* versus *stretched* objects. In a) and b) the arrow is uniformly scaled while setting its size along its length axis to the specified value. In c) and d) the fence is replicated along its length axis to achieve the desired size. No scaling or replication occurs on other axes. In e) and f) the object has no well defined aspect ratio and hence is scaled (stretched) along its default length axis.

- DOOR-GATE
- VIEWING-WINDOW
- VIEWING-OPENING-AFFORDANCE

Hotspot: This represents the area of contact or the source for a particular instrument or device.

- INSTRUMENT-HOT-SPOT
- MACHINE-PATIENT-AREA
- **Source:** SOUND-SOURCE, LIGHT-SOURCE
- **Container-and-Surface**: WORK-SURFACE, MACHINE-PATIENT-AREA, STORAGE-AREA, CONTAINER-BASIN, CAP-COVER, RECEPTICLE
- **Self-Support**: SEAT-WITH-BACK, SEAT, SEAT-STRADDLING, HANDHOLD, FOOTHOLD, HANDHOLD-FOOTHOLD, LYING-SUPPORT, ARM-REST, LEG-REST, HEAD-REST
- **Touch-Grip**: INSTRUMENT-GRIP, CARRYING-GRIP, OPEN-CLOSE-GRIP, PUSH-PULL-GRIP, TOUCH-CONTROL, GRIP-CONTROL, PEDAL, EYEPIECE, EARPIECE, NOSEPIECE, MOUTHPIECE, INSERTION-PIECE
- **Functional-Device**: OUTPUT-AREA, INPUT-AREA, DISPLAY, WRITING-AREA

In order to use an affordance at a graphical level, we need to know what region of the object is associated with the given affordance. We also need to know other properties associated with the affordance – for example, that a path affordance (e.g. a sidewalk) has an implicit direction that affects how one moves on it. We do this by asserting semantic relations between the object and the affordance.

Spatial relations require knowledge about the objects they take as arguments in order to correctly resolve the exact locations on the objects [Coyne *et al.*, 2010b; Tutenel *et al.*, 2009]. In older versions of WordsEye the 3D objects had separate spatial regions known as "spatial tags" that were associated with them. We have adapted our procedure to better conform with the concept of vignettes and affordances, so that rather than composing the scene out of the actual 3D objects with associated spatial tag objects, we now start by constructing a stand-in object for every object in the scene. It is those stand-in objects that are positioned to compose the scene. This allows us to fully represent the scene independent of the actual graphics internals. The most basic stand-in object is a simple 3D box representing size and position. However most stand-in objects will also have additional 3D geometric parts (each stand-in part being a 3D box itself) representing

in abstracted form, the various affordances of the object. So, for example, an object might have 3D parts functioning as one or more top surfaces, and another part functioning as a cupped area. These parts are similar to the spatial tags that marked the original object except that they are built directly into the stand-in object.

We note that this conforms to vignettes where the elements of the vignette are all parts of the vignette itself and can act as affordances. The stand-in object is itself a vignette representing the real 3D object.

Object affordances and other spatial and functional properties of objects are used to resolve the meaning of spatial relations. Examples of spatial regions and their use as affordances are shown in figure 3.2. For example, the *top surface* region marked on the seat of the chair is used in sentences ike *The pink mouse is on the small chair* to position the *figure* (*mouse*) on the *ground* (*chair*). See Figure 3.3 for the depiction of this sentence and several others that illustrate the effect of spatial tags and other object features. Affordances also allow action vignettes to be decomposed. This is mediated with vignettes (see Section 2.3).

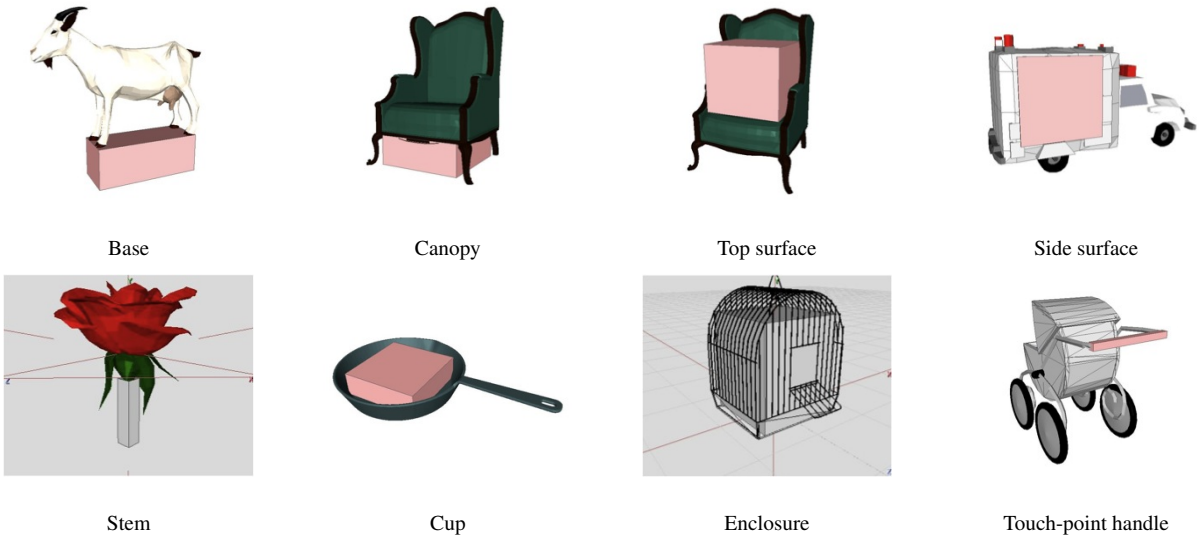


Figure 3.2: Spatial tags, represented by the pink boxes, designate target regions used in spatial relations [Coyne and Sproat, 2001]. These tags are used to construct a new "smart" stand-in object that is used in all scene layout. It is the stand-in objects that are positioned in the scene. The original higher-resolution objects are swapped in to perform the render.

3.8 Spatial Relations

In order to resolve a spatial relation, we find the spatial affordances and other features of the FIGURE and GROUND objects that are applicable for the given spatial relation. For example, if the relation is *under*, a CANOPY affordance for the GROUND object is relevant, but not a TOP-SURFACE. Various other factors, such as size, must also be considered. With ENCLOSED-IN, the FIGURE must fully fit in the GROUND. For EMBEDDED-IN, only part need fit. For other relations NEXT-TO, the objects can be any size, but the FIGURE location might vary. Table 3.2 shows some cases and conditions involved in mapping from prepositions to spatial relations.

In addition to these object-based features, linguistically referenced *subregions* must also be considered. Spatial descriptions often express regions relative to an object (e.g., *left side of*, in *The chair is on the left side of the room*). The same subregion designation can yield different interpretations, depending on the features of the objects.

External-Vertical-Surface: *shutters on the left side of the house*

Interior-Vertical-Surface: *picture on the left side of the room*

Region-of-Horiz-Surface: *vase on the left side of the room*

Neighboring-Area: *car on the left side of the house*

These regions (when present) are combined with the other constraints on spatial relations to form the final interpretation of a scene.

Orientation: The orientation of an object can be expressed in different ways, and will receive different interpretations based on the object shape and the object's intrinsic orientation. The orientation can be *direct* – with the object specified as facing in a particular direction (e.g. left or northwest) or facing towards or away from another object.

When objects are put on horizontal surfaces they normally keep their default orientation. But if an object is put on a vertical surface such as a wall, then the object will normally be placed with its base oriented to be flush with the wall. For example a postcard would be placed flat, with its back toward the surface, for either either a wall or table. Likewise, a fly or insect would be placed with feet oriented toward the surface. A broom, on the other hand, if attached to a wall would probably not be attached with its base toward the wall but instead would remain upright and placed back to the wall. The shape of the FIGURE affects if and how it is re-oriented when placed on differently oriented surfaces.

Spatial Relation	**Example**	**Partial Conditions**
on-top-surface	*vase on table*	*Ground* is *upward-surface*
on-vertical-surface	*postcard on fridge*	*Ground* is *vertical-surface*, *Figure* is *Flat*
on-downward-surface	*fan on ceiling*	*Ground* is *downward-surface*
on-outward-surface	*pimple on nose*	*Ground* is *surface*
pattern/coating-on	*plaid pattern on shirt*	*Figure* is *texture* or *layer*
fit-on-custom	*train on track*	*special base pairing*
ring-on-pole	*bracelet on wrist*	*Figure=ring-shape*, *Ground=pole-shape*
on-region	*on the left side of...*	*ground=region-designator*
hang-on	*towel on rod*	*figure* is *hangable*
embedded-in	*pole in ground*	*Ground* is *mass*
embedded-in	*boat in water*	*Figure* is *embeddable*
buried-in	*treasure in ground*	*Ground* is *terrain*
enclosed-in-volume	*bird in cage*	*Ground* has *enclosure*
enclosed-in-area	*tree in yard*	*Ground* is *area*
in-2D-representation	*man in the photo*	*Ground* is *2D representation*
in-cup	*cherries in bowl*	*ground* has *cup*
in-horiz-opening	*in doorway*	*ground* has *opening*
stem-in-cup	*flower in vase*	*figure* has *stem*, *ground* has *cup*
wrapped-in	*chicken in the foil*	*ground* is *flexible/sheet*
group-alignment	*plate in a stack*	*ground* is *arrangement*
in-mixture	*dust in air*	*Figure/Ground=substance*
in-entanglement	*bird in tree*	*Ground* has *entanglement*
fitted-in	*hand in glove*	*Figure/Ground=fit*
in-grip	*pencil in hand*	*Ground=gripper*

Table 3.2: Spatial relations for *in* and *on* (approximately half are currently implemented). Similar mappings exist for other prepositions such as *under*, *along*. Vignettes resolve the spatial relation given the object features.

Input text: *A large magenta flower is in a small vase. The vase is under an umbrella. The umbrella is on the right side of a table. A picture of a woman is on the left side of a 16 foot long wall. A brick texture is on the wall. The wall is 2 feet behind the table. A small brown horse is in the ground. It is a foot to the left of the table. A red chicken is in a birdcage. The cage is to the right of the table. A huge apple is on the wall. It is to the left of the picture. A large rug is under the table. A small blue chicken is in a large flower cereal bowl. A pink mouse is on a small chair. The chair is 5 inches to the left of the bowl. The bowl is in front of the table. The red chicken is facing the blue chicken...*

Figure 3.3: Spatial relations and features: *enclosed-in* (*chicken in cage*); *embedded-in* (*horse in ground*); *in-cup* (*chicken in bowl*); *on-top-surface* (*apple on wall*); *on-vertical-surface* (*picture on wall*); *pattern-on* (*brick texture on wall*); *under-canopy* (*vase under umbrella*); *under-base* (*rug under table*); *stem-in-cup* (*flower in vase*); *laterally-related* (*wall behind table*); *length-axis* (*wall*); *default size/orientation* (all objects); *subregion* (*right side of*); *distance* (*2 feet behind*); *size* (*small* and *16 foot long*); *orientation* (*facing*).

It should also be possible through language to specify an object's orientation *indirectly* as a positional spatial relation that implies the overall orientation. For example *the left side of the dog is on the floor* should re-orient the dog. This is not currently supported by the system.

3.8.1 Spatial Reference Frames

Linguistically expressed reference frames can be absolute, intrinsic, or viewer-centric [Levinson, 1996]. We assume an absolute reference frame except for two cases a) when language is used to invoke an object's intrinsic reference frame (e.g. *the box is at the chair's left*) and b) when an object is supported by another object – in that case, the supporting object will supply the reference frame. For example, in "*The vase is on the table. The book is to the left of the vase. The table is facing right.*" the book's location relative to the vase depends on the table's orientation. Intrinsic reference frames can also be supplied by what we call a *guide* — a large directional object like a wall or path that serves as an implicit reference frame. For example, if we say [*The bookcase is left of the dresser. They are against the left wall*] then the left wall functions as guide and establishes a frame of reference for the spatial relation.

Reliably inferring the intended reference frame from text is generally not possible. The sentence *the dog is in front of the car* could easily be interpreted either using the car's intrinsic reference frame, in which case

the dog might get hit if the car is moving forward. Or it could be interpreted in a viewer-relative reference frame where the car could be facing sideways and dog is visually in front of the car. Additionally, since the camera can move, a relative reference frame would not be stable or derivable purely from the text but instead depend on how the viewer is looking at the scene at the moment. As a result, WordsEye uses a *stage-centric* absolute reference frame where an imaginary stage defines all directions aside from the intrinsic reference frame exceptions noted above. We note also that we support text using the cardinal directions in sentences like *the cow is facing north*, where north is into the screen. This is analogous to looking at a map on a table where north is away from and south is towards the viewer.

3.9 Facial Expressions

Scenes involving human characters carry important semantic content in the *facial expressions* and *facial features* of those characters. Facial expressions (such as smiling) convey aspects of emotion, communicative intent, personality, mental states, and actions. Facial features convey age, gender, race, attractiveness, and other intrinsic properties of the person.

To depict facial expressions, we must be able to control the actual facial shape of 3D human characters. Ekman [Ekman, 1992] describes six universal (core) emotions. These are anger, disgust, fear, happiness, sadness, and surprise. We are not looking to convey just emotions or just core emotions but a much wider set of emotions and facial expressions. To do this, we used FaceGen [FaceGen,], a commercially available software package that allows facial expressions and features on 3D characters to be controlled by a relatively high-level set of parameters. We note that the current running version of WordsEye no longer has access to the FaceGen graphics software, so the graphical output displayed in Figure 3.4 was created with earlier versions of WordsEye.

Since we wanted to convey as large a set of facial expressions as possible, we extracted a large set of emotion terms from digital lexical resources (WordNet and Whissell's Dictionary of Affect [Whissell, 1989]) and mapped those to graphical parameters provided by FaceGen software. For example:

```
expressions: ANGER, DISGUST, FEAR, SAD, SMILECLOSED, SMILEOPEN, SURPRISE
modifiers: BROWDOWN-LEFT, BROWDOWN-RIGHT, BROWUP-LEFT, LOOKDOWN
phonemes: BIG-AAH, CH-J-SH, EH

Example Mappings:
"gleeful" => ((:expression-smileclosed 0.45) (:expression-smileopen 0.63))
"sad" => ((:modifier-lookdown 0.26) (:expression-sad 0.66) (:phoneme-eh 0.15))
```

The system has mappings for approximately 250 emotion words. See appendix D. Similarly, we extracted and mapped a wide-ranging set of descriptive terms for other (non-emotion-related) facial features. This work was done by Cecilia Schudel. These facial feature mappings when fully operational will allow WordsEye to control from its textual input a large set of overall facial characteristics such as race, gender, and age as well as more specific facial features such as size of chin, shape of lips, and width of forehead.

Figure 3.4: *Obama is in front of the Republican Elephant emblem. He is furious. The "election" is above him.*

3.10 Poses

Posed characters are used to convey actions in visual scenes. Poses can require the full body or only particular parts. Arm and leg poses are achieved either by stored rotations on a character or by dynamically using inverse kinematics [Tolani *et al.*, 2000] and targeting the effector to an obect affordance in the environment. In order to determine the target points for dynamically constructed poses, object affordances can

be used. We note that the newest version of WordsEye does not fully support poses at the graphical level yet. Previous versions have supported it, and we describe that here.

Poses, themselves, are parameterized by target positions for the hands and feet, extension of limbs, tilt of the torso and neck, bending of the arms and legs, spread of the arms and legs. We distinguish between three broad categories of poses [Coyne and Sproat, 2001]. Poses can be combined, depending on what body parts are being changed, allowing a character to be running and holding an object at the same time, for example. Likewise, arms and lets can be moved by inverse kinematics, allowing modifications to stored poses.

Standalone poses are poses, such as *running*, where the character is not holding an object.

Grips convey how an object is normally held. There can be more than one such pose. Grips will be achieved by choosing the appropriate basic hand poses for the object involved. For example a medium sized hand-held object such as a drinking glass would be held between the thumb and the fingers. The pose will be dynamically adjusted for the size of the object.

Usage poses convey how a human uses a given object. For example, using a bicycle involves sitting on the seat, extending the feet to the pedals and the hands to grasp the handlebars.

3.11 Surface Properties

Surface properties include color, texture (pattern), decals, transparency, reflectivity, and other render-related properties. Surface properties such as decals and textures can be layered with levels of transparency. Surface properties can be applied to both 3D objects and to images themselves.

Surface properties, by default, apply to the *dominant parts* of an object. It looks unnatural when an entire object is a uniform color – normally, the accents and small parts should remain unchanged. Of course, the user can override this and say in language [*the ENTIRE cow is brown*], and then even the whites of the eyes and the horns will be brown. As a default, dominant parts are computed by a heuristic of finding the largest sized parts up to some threshold. But, in practice, this is does not usually work well since it is often more of an aesthetic judgement as to what constitutes the most visually salient set of parts. As a result, dominant parts are explicitly specified in the VigNet knowledge-base for objects where the automatic calculation is insufficient. We note that in some cases, a color term could more reasonably designate a part when the color is related to the function of the apart of the object. For example, a *red pen* may refer to a pen with red ink.

We thank Richard Sproat for pointing out this distinction.

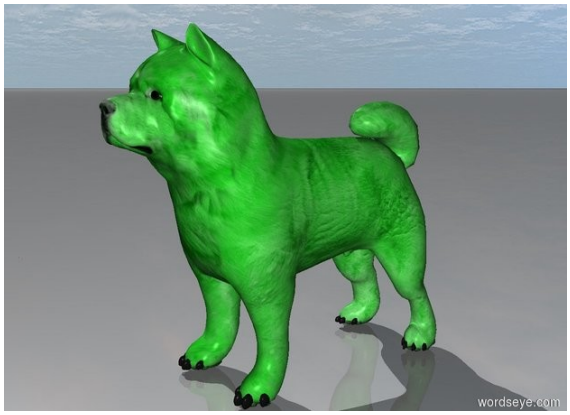
Not all objects have dominant parts. For some objects, the entire object is a single geometric part and what apear as parts are really just different regions of the 2D texture map image. For this type of object, it is impossible to change the color in the same way. Also, replacing the texture with a solid color does not look natural. As a result, in these cases we modify the texture rather than replacing the texture with a solid color. The rules for modifying the texture are as follows. When setting the texture to a color, we modify the hue and saturation of the texture image as a preprocessing step before rendering. The intensity (brightness) component of the texture is left unchanged. However, for very dark objects, the color will not be visible unless the brightness of the overall texture in increased. So for such objects we set the hue and saturation components and boost the intensity (brightness) component. See figure 3.5 for examples. And see 4.4.6 for a description of how the language and semantics is processed.

3.12 Conclusion

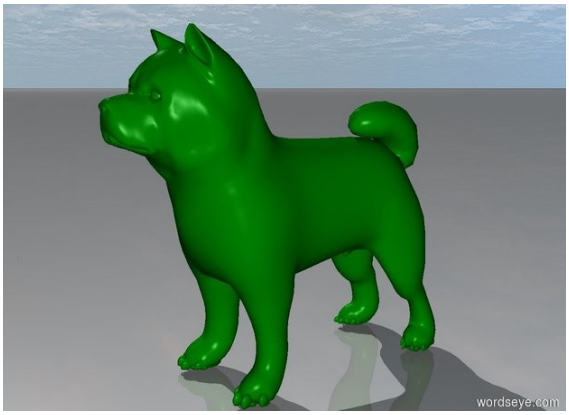
This section has described some of the areas where graphical objects exhibit semantic properties and primitives beyond the pure linguistic types they are associated with. This involves how they establish reference frames, the spatial affordances they provide, and a variety of functional properties that control how they scale and orient themselves. These properties can apply to 3D objects or 2D images. We have also discussed how scenes are interpreted, including both how render and camera effects can change the semantics as well as how figurative language an be suggested from simple objects in the appropriate configurations.

a) *A green cow.*

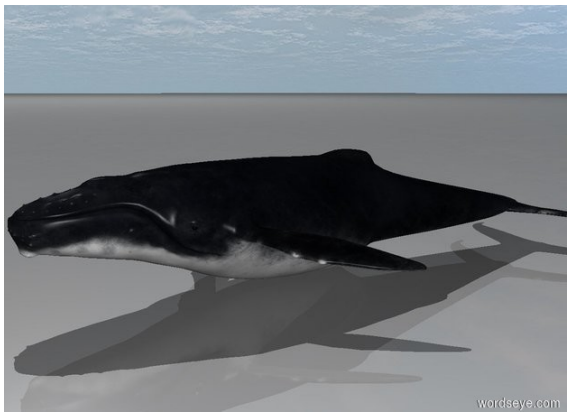
b) *An entirely green cow.*



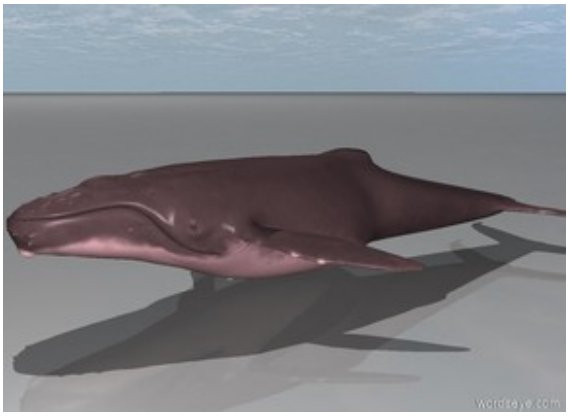
a) *The green dog.*



b) *The solid green dog.*



d) *A humpback whale.*



e) *A pink humpback whale.*

Figure 3.5: Handling colors: Examples a) and b) illustrate the effect of dominant parts. Specifying *ENTIRELY* will override the dominant part and set the entire object to the color. Examples c) and d) illustrate how colors on objects with textures are changed. In c) the texture is tinted (leaving the intensity as-is), which looks relatively natural and retains the textured feel, whereas in d) the texture is set to a solid color. In e) and f) the original object has a texture but is very dark. So in this case the brightness is increased in addition to the hue and saturation being set.

Chapter 4

Lexical Semantics

4.1 Introduction

Text-to-scene generation is concerned with the meanings of words, both individually and in combination. In this chapter we discuss how information in VigNet is used to interpret a variety of lexical semantic phenomena. Note that there is a certain amount of arbitrary division, with some lexical semantics issues covered in the graphical semantics discussion (Chapter 3) in cases when the graphical aspects dominated. Likewise the computational pipeline discussion (Chapter 6) also touches on some lexical and graphical features when it was expedient to describe them in the context of the overall computational flow.

4.2 Lexical Entities

Lexical entities are linked to concepts and semantic relations in the ontology. In fact, most concepts have lexical accomplices – nouns are associated with concepts; and verbs, prepositions, adjectives, and adverbs are linked to semantic relations. Concepts in the ontology (and hence lexical items attached to them) are given meaning by virtue of the semantic relations that reference them. The most basic relation is IS-A which defines type inheritance between concepts.

4.3 Lexical Functions as Meta-Relations

In addition to semantic relations between concepts, VigNet also defines a smaller number of semantic relations between relations themselves or between relations and concepts. These meta-relations cover many of the lexical relations described in Cruse [Cruse, 1997] as well as some of those described by Melcuk. See Section 2.2.1.

In the following example, a meta-relation is asserted between a directional target (FRONT) and a spatial relation that it invokes (IN-FRONT-OF). This allows sentences of the form *the dog is at the right of the cat* to be processed. In this sentence, *right of the cat* specifies a region via a partitive relation that supplies a head noun of *right*. The dog is by the right of that region. The meta-relation converts the one to the other.

```
(we.directional-rel-to-direction.r
       :direction-relation gfx.in-front-of.r
       :direction front.n)
```

One might argue that in this case *right* is really shorthand for *right side*, in which case some more structure would be needed to make the conversion. That may be true, but nonetheless, a lexical-function-like transformation via meta-relations is needed in either case.

Assertions of this form are used by semantic translation rules to convert one semantic representation to another in a data-driven manner. The knowledge required to make the translation is encoded separately from the translation rule which performs the translation. In the rest of this section we give additional examples of lexical function behavior in the realm of meta-relations.

4.3.1 Gradable attributes

Adjectives are associated in VigNet with corresponding unary semantic relations. Translation rules are used to map those relations to values in an appropropriate domain (for example that *hot* means having a high *temperature*). This mechanism handles various lexical relations such as gradable attributes and antonyms. For every gradable attribute relation there is an assertion in the knowledge-base mapping it to a scale on another relation. For example:

```
(we.domain-attribute-value.r
       :relation we.size.big.r
```

```
:domain gfx.size.r
:scale-value 2.0)
```

This allows a single semantic translation rule to map all gradable attributes to their domains. Similar transformations utilizing lexical functions can happen with other, non-gradable attributes. For example, *the wooden vase* implies *the vase is made of wood*. This transformation can be made using the asserted meta-relation between substance attribute (*wooden*) and substance entity (*wood*).

4.3.2 Size Domains

Size can be specified directly (*3 feet tall*) or as a gradable attribute (*big*). For the latter, our approach is to map those to a scale factor. It could be argued that this approach is too simple-minded, and that a term like *big* should only change size relative to the norms for the class of object at hand. Approaches have been tried to to interpret size modifiers based on other factors [Mitchell *et al.*, 2011]. But it is not clear how well any given approach will work in all use-cases. The simple direct scaling system system is fairly predictable and quickly grasped by users, whereas a more complex system might be better in the aggregate but less useful in some contexts. We discuss this general issue of the tradeoffs between predictabity versus accuracy in Section 11.1.

4.3.3 Generics and Substantives

Some nouns denote concrete entities. Other nouns (e.g., *location*, *size*, *brightness*) denote functional roles (arguments) within a relation. These nouns are handled by inheriting from the concepts that designate those roles. Lexical relations of this type will allow us to interpret sentences such as *The enormous size of the lion scared everyone*, where *size* denotes a semantic role rather than a concrete object. Likewise nouns such as *invention* can denote both a result and an event. These, also, are handled by lexically oriented semantic meta-relations relating them to relational argument roles.

4.4 Interpretation and Implict Relations

In this section we examine several instances of lexical semantic interpretation. We note that Chapter 3 overlaps in topic, though not detail, with some of what follows.

Much language usage involves implicit semantic relations and arguments. As such, it is underspecified and takes on more specific meaning in context. By *context* we do not mean only (or even primarily) the context supplied by discourse or pragmatics. Instead we are concerned with the lexical semantic context – for example, how a preposition's meaning depends on its arguments, or how a verb will suggest defaults to missing roles (for example, the instrument of *punched* is a fist).

4.4.1 Prepositions

Preposition sense disambiguation for *over* and *with* has been studied computationally in [Boonthum *et al.*, 2005] and [Alam, 2004]. In addition to their role in verbal adjuncts, prepositions are especially important in text-to-scene generation, since they directly denote spatial relations in a variety of subtle ways [Herskovits, 1986]. Prepositions can also carry non-spatial senses. So, for example, [*John is on the sidewalk*] and [*John is on the bus*] map to a WALKING-STANDING-AREA or SEAT affordance. Likewise, [*The flower is in the vase*], [*The spider is in the web*], and [*The goldfish is in the aquarium*] represent different spatial interpretations of *in*.

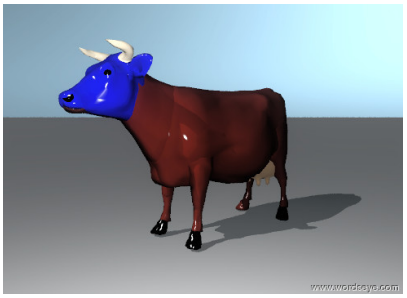
Text (?n0 on ?n1)	**input match patterns**	**Resulting Frame Relation**
john on sidewalk	isa(?n1, phys-ob), has-top-surface(?n1)	on-top-surface(n0, n1)
john on wall	isa(?n1, phys-ob), has-front-surface(?n1)	on-front-surface(n0, n1)
john on phone	isa(?n1, comm-device), isa(?n0, human)	use-instrument(n0, n1)
john on tv	isa(?n1, display-device), isa(?n0, human)	display(?n1, ?n0)
john on bus	isa(?n1, public-vehicle), isa(?n0, human)	ride-vehicle(?n0, ?n1)

Table 4.1: VigNet: Semantic mappings for *on*

To address these issues, VigNet extends FrameNet's notion of valence pattern to directly include semantic and contextual constraints, drawing upon the semantic types of words and their semantic and contextual relations to other words as defined in VigNet [Coyne *et al.*, 2011a]. This allows the appropriate semantic relation to be assigned to parsed input text. Consider, for example, a few of the semantic interpretations for the preposition *of* and how they are handled by the system in Table 4.2 and Figure 4.1.


Made-of: *horse of stone*


Grouping-of: *stack of cats*


Part-of: *Head of cow*


Dimension: *height of horse*


Represents: *picture of girl*


container-of:*bowl of cats*

Figure 4.1: Depictions of *of*

Text (A of B)	**Valence Patterns for** *of*	**Resulting Frame Relation**
bowl of cherries	A=container, B=plurality-or-mass	container-of(bowl, cherries)
slab of concrete	A=entity, B=substance	made-of(slab, concrete)
picture of girl	A=representing-entity, B=entity	represents(picture, girl)
arm of the chair	A=part-of (B), B=entity	part-of (chair, arm)
height of the tree	A=size-property, B=physical-entity	dimension-of(height, tree)
stack of plates	A=arrangement, B=plurality	grouping-of (stack, plates)

Table 4.2: VigNet: Semantic mappings for *of*

4.4.2 Noun Compounds

Noun-noun compounds contain an implicit semantic relation between the two nouns. For example, *olive oil* could be interpreted as (source-of.r :source *olive* :product *oil*). Levi [Levi, 1978] posits that there are nine possible target relations. Downing [Downing, 1977] argues against Levi and posits that there is no fixed set, and that general predicates such as *for* (which Levi assigns to both *headache pills* and *fertility pills*) are so

vague as to be meaningless. A variety of computational approaches have been tried to map to elements to a given set [Barker and Szpakowicz, 1998], [Rosario *et al.*, 2002], [Vanderwende, 1994], [Girju *et al.*, 2005].

Figure 4.2: [Sproat and Liberman, 1987] study speech stress patterns in English nominals and point out the common misconception that left-stressed examples (such as *dog annihilator*) need not be members of the phrasal lexicon. We find a similar productivity and lack of regularity in visual analogs as well.

Some noun-noun compounds can be felicitously depicted using textures. For substances, we can put a substance of that type as a texture on the object. Sometimes a noun-noun compound can be used to select one subtype of object over another. For example, *A New York skyscraper* or *a Chinese house*. In other cases, an emblem for the leading noun can be used as a texture map, such as a flag if a country is named. In fact, any associated image can be used to depict idiosyncratic interpretations. See Figure 4.3 for depictions of these examples.

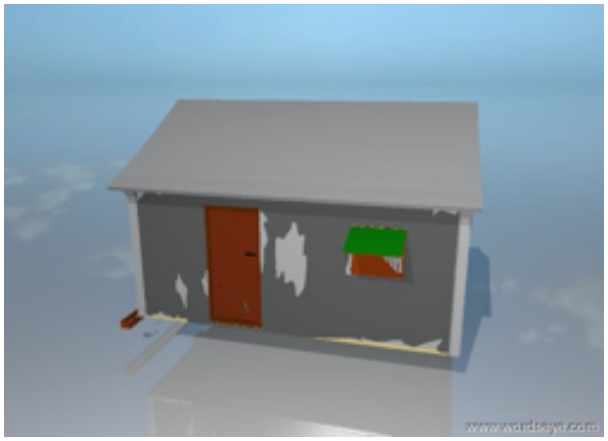
Our approach here is similar to our approach with prepositions – we use semantic properties of the individual nouns to help resolve the implied semantic relation. The actual translation from the noun compound to the semantic relation is made with semantic translation rules. Relevant features for N-N compounds are show in Table 4.3.

Text (?N0 ?N1)	**Valence Patterns for *of***	**Resulting Frame Relation**
American Horse	emblem-for(?Emb,?N0)	Has-texture(?n1,?emb)
Stone Horse	isa(?n1,substance)	made-of(?n1,?substance)
China House	style(?sel,?n0), isa(?sel,n1)	?sel
Van Gogh Horse	creator-of(?n0,?image)	Has-texture(?n1,?image)
Goat head	part-of(?n1,n2)	?n1
Flower picture	representing(?image,?n0)	has-texture(?n1,?image)

Table 4.3: VigNet: Semantic mappings for noun-noun compounds

Emblematic: *American horse*

Substance: *stone horse*



Selection: *Chinese house*

Authorial: *Van Gogh Horse*

Figure 4.3: Semantic mapping of noun-noun compounds

4.4.3 Metonymy

Metonymy is "a figure of speech in which a thing or concept is called not by its own name, rather by the name of something associated in meaning with that thing or concept" [Wikipedia, 2016h]. It is an important linguistic phenomenon that must be handled in text-to-scene conversion. Langacker [Langacker, 2009] (pp. 40–59) gives an example illustrating the extensive use of metonymy in language. In the sentence *the cigarette in her mouth was unlit*, the cigarette is really between the lips (not in the mouth) and most of the cigarette is actually outside the mouth and lips. Whether or not one accepts this as an instance of metonymy or instead as a simple use of parts or affordances, this example illustrates how flexible language can be in terms of what part, whole, or functional aspect is referenced.

The question becomes what parts of objects can stand in for the whole. This is largely determined by the affordances (see Section 3.7) provided by the given object. The tip of the cigarette provides a MOUTHPIECE affordance and the lips can be considered either as an INPUT-AREA-AFFORDANCE or as an EFFECTOR, similar to more active body parts like hands or feet.

Many other sorts of metonymy are of direct interest to WordsEye. For example, *in the sun* is metonymic for *in the sunlight*. Like noun-noun compounds, metonymy invokes an implicit relation. If we consider such relations to be binary, there are three elements of interest, the two arguments and the relation itself. For the case of noun-noun compounds the arguments are both explicit but the relation is implicit. In the case of metonymy, only a single argument is explicit, while the target argument and the relation that connects them are both implicit. In the example above, only *sun* is explicit, while the SOURCE-OF relation and the true referent (sunlight) are implicit.

Container-contents substitution is another important type of metonymy that shows up in text-to-scene conversion. The visual import of *The beer is on the table* is that a beer bottle (albeit with beer in it) is on the table.

Also, as we discussed when discussing noun-noun compounds, emblems for objects affixed as textures can be used to generatively instantiate a property as a visual metonymic substitute. Likewise, the emblem can stand for the object mentioned on its own, as a visual analog of a generative lexicon [Pustejovsky, 1995].

4.4.4 Regular Polysemy

Regular polysemy is a related phenomenon where a limited set of implicit relations connect related word senses such as *chicken* (the animal) and *chicken* (the meat). This is handled in VigNet by asserting with

an explicit relation between the concepts for the related terms. For example, the two senses of *chicken* are related by a SOURCE-OF relation. A list of a dozen common polysemous relations is given in [Peters and Peters, 2000].

Another common type of regular polysemy that shows up in WordsEye is with plants and the fruits or other parts of those plants. For example, a *flower* can refer to the entire plant or the blossom. Our 3D object library has both plants with flowers on them and isolated flower blossoms. So, for example, the rose plant is assigned to the concept ROSE-PLANT.N while the blossom (either standalone or the 3D part of the entire plant) would be assigned ROSE-BLOSSOM.N. We note that in general, there will not be a specific concept for each blossom type. In those cases, we can still posit a new entity on the fly (e.g. ?blossom) and assert that (isa.r :super blossom.n :sub ?blossom) and (we.part-of.r :whole rose-plant.n :part ?blossom).

4.4.5 Discussion

The area of implicit relations gets at the heart of lexical semantics and hence much of language understanding. It certainly is important in the visual analogs central to text-to-scene conversion. VigNet provides a representational framework to encode the necessary information to process these and already encodes a wealth of such knowledge. WordsEye currently utilizes some of that knowledge and will do more in the future.

4.4.6 Modifiers on attributes

Attributes themselves can be modified. A typical example of this is with intensifiers such as *VERY big* and color modifiers such as *BRIGHT blue*.

When modifying colors, we need to be able to handle variants such as: *the dog is entirely green* versus *the entire dog is green* versus *the entirely green dog*. We also handle *the dog is solid green* versus *the solid green dog*. See Section 3.11 for a discussion of color and other surface properties and how they can be modified in this way. See also section 6.4.3.3 for a short discussion of how attribute modifiers are processed.

4.5 Conclusion

Lexical semantics issues are the heart of text-to-scene conversion. In converting language to visual scenes, we can better understand how language works.

Chapter 5

Populating VigNet

5.1 Introduction

We haved explored a combination of methods to populate VigNet. These include a) hand annotating text files b) crowdsourcing with Amazon Mechanical Turk and the WordsEye system itself, c) mining existing resources such as FrameNet and WordNet, d) automatically extracting information from corpora, and e) writing software tools to facilitate manual annotation.

We discuss the different methods of populating the resource and the knowledge we have acquired and intend to continue acquiring in the sections below.

5.2 Acquiring the Noun Lexicon

WordsEye's noun lexicon, consisting of approximately 15,000 nouns, was created manually as follows. Initially, concepts (with their corresponding nouns) were assigned to every object in our 3D library. Those concepts were manually grouped under *supernodes* (using multiple inheritance) in similar fashion to the WordNet hyperym hierarchy and other ontologies.

Additional related nouns were added for missing objects, often using WordNet as a reference. For example, although our original 3D object library had objects for tiger, cheetah, and lion, it had none for jaguar. In cases like these, we created concepts for the missing words. These additional words typically occurred in large clusters and could be imported in large groups from WordNet, with appropriate filtering to eliminate obscure senses (e.g. "Spoon" as as type of golf club).

The nouns incorporated into concepts in this way include almost all *concrete nouns* in common usage. Other sets of nouns, related to parts of objects and distances and other graphically related terms, were added in a bottom-up fashion to support the language used in depicting spatial relations. Another large set of relationally oriented nouns was imported from FrameNet. In addition, named entities and sublexical items were assigned to concepts. Our target was a basic vocabulary, focused on descriptive language that would be readily used and understood by roughly a 10 year old child. Obscure words and word senses were excluded based on common sense judgments as to whether such words and senses would would actually be commonly used to describe scenes.

In the process of defining and modifying the ontology, in the context of the full text-to-scene system, functionally oriented concepts are sometimes created to facilitate decomposition or other semantic translation rules. For example, a concept containing "dangerous items" was created to be used for vignettes signifying danger. Two overriding goals are to clearly recognize conceptual and functional versus lexical notions in inheritance (as noted in Section 2.2.2.1) and to represent functional and relational structure in VigNet's assertions and relations rather than in the lexicon itself.

Also, as we add new 3D objects, especially those representing celebrities and historical figures who are unlikely to already be in our lexicon (Section 2.4.2), we must add new concepts to define those objects or individuals. WordsEye currently has over 250 individuals as 3D composite objects. These are inserted into the ontology so that THOMAS-EDISON.N, for example, is a subnode of MAN.N and INVENTOR.N. We could also add assertions into the knowledge base such as his birthdate, nationality, and so on as desired. Once the entity is assigned a concept, it becomes available as a referent and any number of facts can be asserted and others can be inferred.

5.3 3D Objects and Images

VigNet incorporates meta-information referencing a library of 3,500 polygonal 3D objects from various sources. This includes several hundred custom-made objects. A concept for each object has been inserted into the IS-A hierarchy. So, for example, the concept for a 3D model of a dump truck would be inserted as a subnode of the concept for *dump truck*. In addition, the 3D model is assigned a real world size, a default orientation, the style of object (when appicable), and the substance it is made of. These are all represented by semantic relations. All named parts of objects are also assigned to concepts. So, if the dump truck has a

3D part for wheels, those will be assigned to the appropriate concept. Finally, objects are tagged with spatial regions corresponding to affordances (Section 3.7).

VigNet references a library of 5,500 public domain 2D images. Like 3D objects, every image is assigned a unique concept as a child of its genre (illustration, photograph, texture). Semantic relations can also be asserted about the image's author, content, and style. Elements depicted by images or representational 3D objects (such as statues or iconography) are also given concepts and in some cases semantic relations are asserted between those elements to annotate what the image or 3D object represents.

5.4 Relational Lexicon – Prepositions, Verbs, Adjectives, Adverbs,

FrameNet frames, as well as verbs in those frames, were automatically converted into semantic relations, where the frame becomes a relation and the core frame elements become the arguments to the relation. Many lexical items that invoke relations (such as verbs, adjectives, adverbs and deverbal nouns) were directly imported from FrameNet. When a necessary word was missing from FrameNet and an an appropriate frame existed, it was added manually.

Prepositions were entered manually and assigned one-to-one with a corresponding semantic relation. Different spatially oriented sub-senses of prepositions were assigned to the appropriate parent semantic relation. Knowledge-based translation rules were manually defined to map adjectives to domain values (Section 4.3). For various lexical items, additional features have been assigned manually, such as whether they take measure arguments.

5.5 Vignettes

5.5.1 Crowdsourcing

We have worked on two crowdsourcing efforts. Data has been collected but not yet integrated into the system.

5.5.1.1 Amazon Mechanical Turk

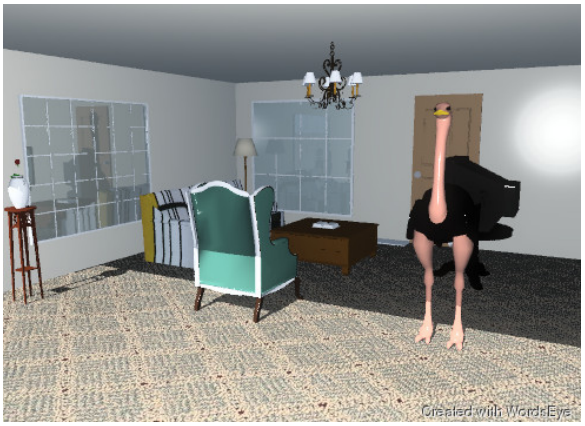
The success of crowdsourcing in NLP [Snow *et al.*, 2008] has encouraged us to explore using Amazon's Mechanical Turk [Wikipedia, 2016b] to acquire knowledge about the constituent objects and their arrange-

ment for action vignettes and location vignettes. We have performed several preliminary experiments to collect object properties (parts, typical location) as well as typical objects in rooms. We are now scaling up these efforts to acquire a larger amount of data.

We used Mechanical Turk to acquire location vignettes, in particular for rooms. The approaches involve showing pictures of different types of rooms to Turkers and having them list the main structural objects in that room picture (for example, a particular kitchen might have a stove and refrigerator and sink). We then ask them to specify the spatial relations between the objects, particularly which objects are against walls, and what surface is supporting them. We also ask about room size and surface textures. In another variant we had Turkers describe the room spatially, and we then parsed their descriptions into spatial relations using WordsEye's parser. These experiments, performed by Masoud Rouhizadeh, are described in [Rouhizadeh *et al.*, 2011a] and [Rouhizadeh *et al.*, 2011b].

5.5.1.2 WordsEye System

We have collected instances of about 40 different room types (e.g. kitchen or living room) using the WordsEye system to position objects. See Figure 5.1. The construction of these rooms was done by Richard Sproat. The rooms were built using the base room vignette that consists of a left and right and back wall, plus a ceiling and floor. The front wall is omitted to facilitate camera viewpoints. The same technique is intended to be crowsourced more widely to let users create their own location and composite object vignettes. These constructed room types have not yet been integrated into the system as vignettes.



a) living room (ostrich not included)

b) kitchen

Figure 5.1: Rooms created online with WordsEye, a technique suitable for crowdsourcing.

5.5.2 Composite Objects Vignettes

We currently have two classes of composite object vignettes in the system.

The first is a room vignette. This vignette creates a room by arranging walls, ceiling and floor. In order to make camera viewing easy, we omit the front wall. The room is constructed with flexible constraints, allowing the size of element to change and have the rest adjust. See section 6.6.1. In the future this vignette can be used as a basis for creating specific room types.

"Cutout characters" are a second type of composite object vignette. See Figure 2.5. These are created automatically automatically from the annotated heads of the characters.

5.5.3 Stand-In Object Vignettes

Stand-in Object Vignettes were semi-automatically created from the 3D object library, combining computed object bounding box size information with spatial affordance regions that were annotated by hand using custom 3D tools. The resulting information was merged to create vignette definitions providing the necessary element information for all 3D objects.

5.5.4 Location Vignettes and Iconic vignettes

The room vignette and a few iconic vignettes were created by hand. About 40 preliminary room (location) vignettes crowdsourced on the WordsEye system are waiting to be integrated into the system.

5.6 User Interfaces for Manual Knowledge Acquisition

Smart interfaces can improve speed and accuracy of annotation. For example, Kilgarriff describes a lexicography tool that presents salient collocated words automatically grouped by grammatical relation to the lexicographer. [Kilgarriff and Tugwell, 2001]

5.6.1 VigNet Ontology Browser

The web-based VigNet ontology browser (Figure 5.2) is a very useful tool that supports supports searching and browsing for all concepts and their subnodes and supernodes, as well as all asserted relations that reference the given concept. The browser includes not only lexically oriented concepts but also displays thumbnail images for 3D objects and 2D images.

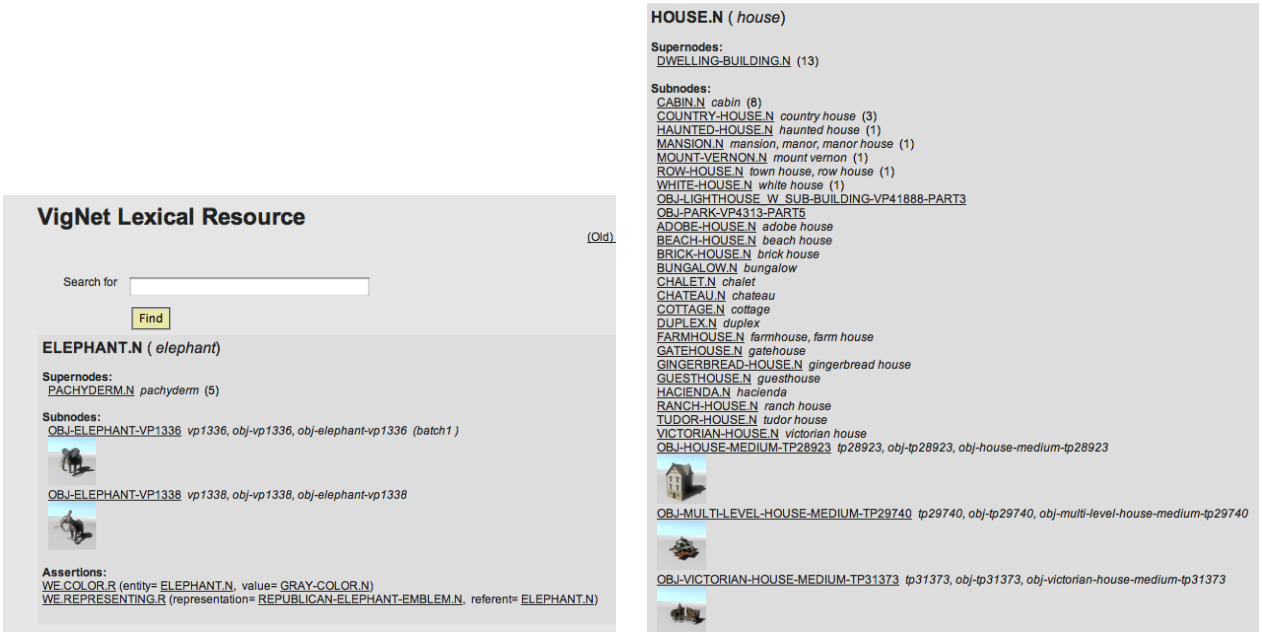


Figure 5.2: VigNet Browser: *elephant* and *house*

5.6.2 Part Annotation

We have built software tools and a user interface to facilitate the renaming and the assigment of affordances to 3D object parts. The objects in our 3D object library have parts assigned to them by the artists who created the objects. The names of those parts are normally not regular English words. For example the left tusk of an elephant is called *ltusk*. In order to reference these parts from language, it is necessary that they be mapped to actual words. This mapping was done by Alex Klapheke using a custom annotation interface (Figure 5.3) to assign lexical items to internal 3D object parts. In a second pass, affordances (Section 3.7) were assigned to all appropriate parts on the object.

After assigning an English name to parts, the annotated terms were all fed through the WordsEye lexicon to find the appropriate set of possible concepts. And finally, hand-corrections and heuristics were applied to disambiguate in cases where the lexical form mapped to more than one concept.

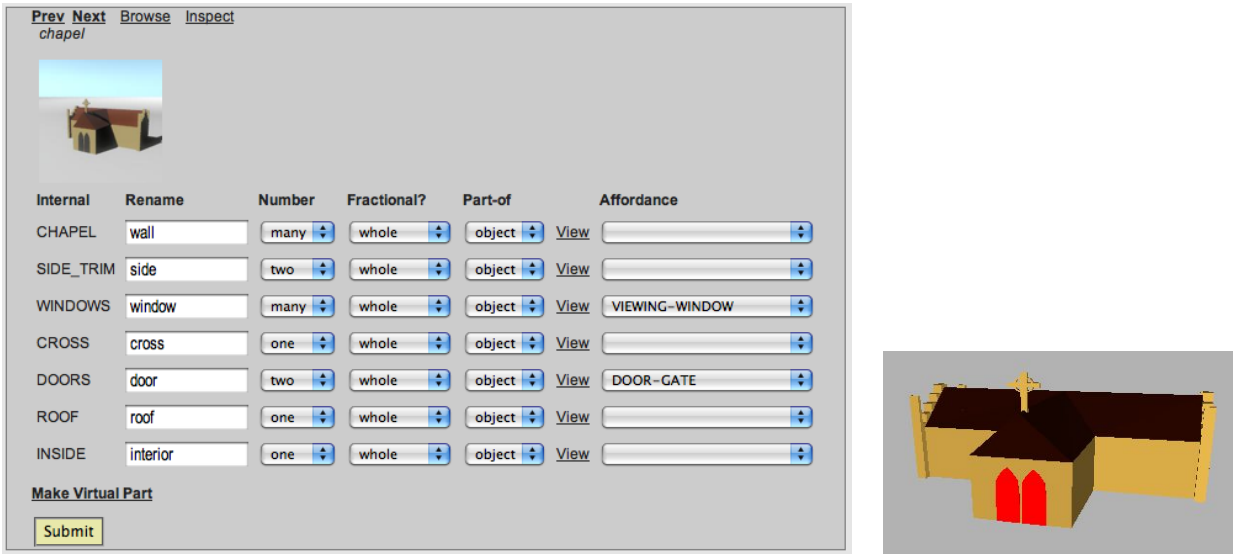


Figure 5.3: User interface for renaming parts

5.6.3 Location (Room) Vignette Editor

And see Figure 5.4 for a screenshot of a simple software tool for defining default properties of room location vignettes.

5.6.4 Action Vignette Annotation

An ongoing and unfinished aspect of our work is to annotate many examples of verbs and argument with action vignettes. In order to acquire action vignette decompositions, we first identified a set of "core verbs" (see Appendix E). This list was created by merging a list of common action verbs acquired manually over the years of working with WordsEye with verbs in FrameNet. Verbs of interest were rated on a scale from 0 to 5, where less commonly used and less concrete and depictable verbs were given lower ratings or excluded. We plan to initially focus on 500 high-priority core verbs (ranked 0-3) out of 1000 core verbs total.

Our next task was to find common verb-argument pairings, so we parsed the British National Corpus (BNC) [Wikipedia, 2016d] with the Mica dependency parser [Bangalore *et al.*, 2009]. A summer student (Alex Klapheke) wrote a script to extract the verb and head words for the verbs arguments from the parsed output. These were then filtered to only include verbs from our set of core verbs.

We have built a custom user interface that allows vignette decompositions to be assigned to verb patterns

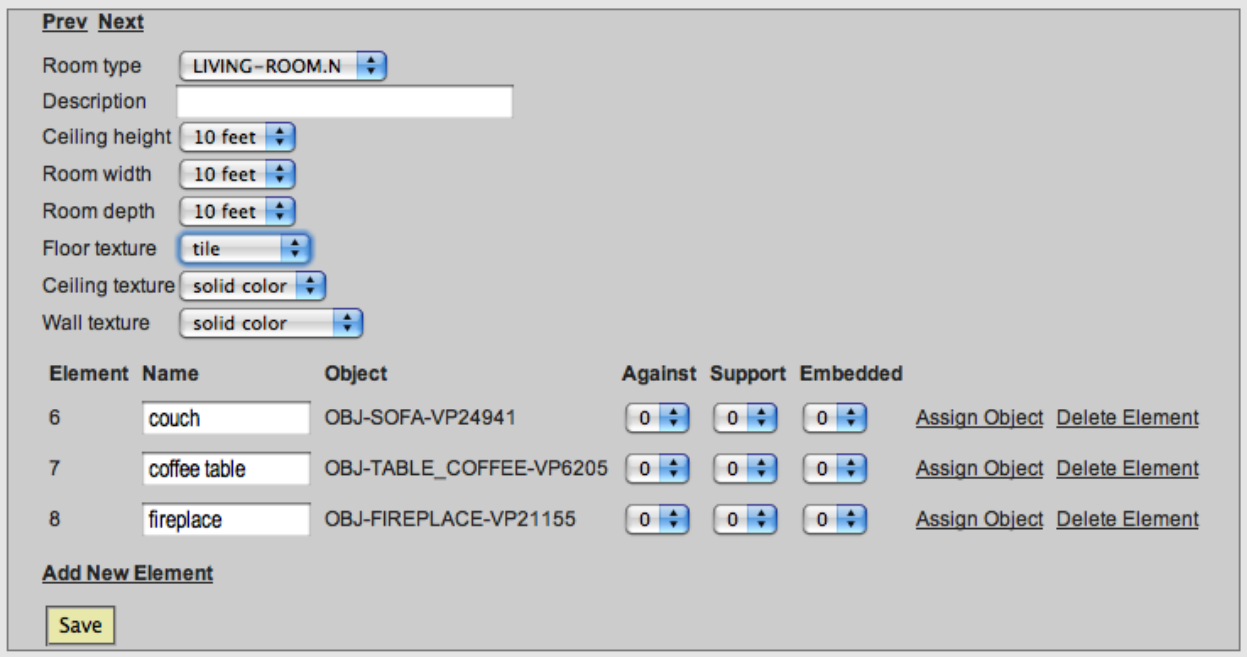


Figure 5.4: Room Layout Editor

(Figure 5.5). The interface lets the annotator specify how verb patterns can be decomposed into sets of primitives or abstract vignettes (Section 2.4.5). The interface gives choices to decompose the verb pattern into a few relevant graphical primitives and a set of 35 abstract vignettes. About 2900 action vignettes have been annotated in this manner so far, covering 90 verbs (with varied argument patterns). The interface allows verb patterns to be grouped together if they invoke the same vignette decomposition.

The next steps are for an action vignette to be created for each verb pattern group. These action vignettes will be integrated into the VigNet itself and the full WordsEye system once the posing primitives are in place on the graphical end.

5.6.5 Adding New Graphical Content

When new 3D objects are added to the system, they must be integrated into VigNet in order to be used. This involves adding meta-information about the object.

Unique ID: Every object is given a unique concept and associated identifier.

IS-A: Every object must be inserted into the ontology. Most objects will fit under an existing type, but

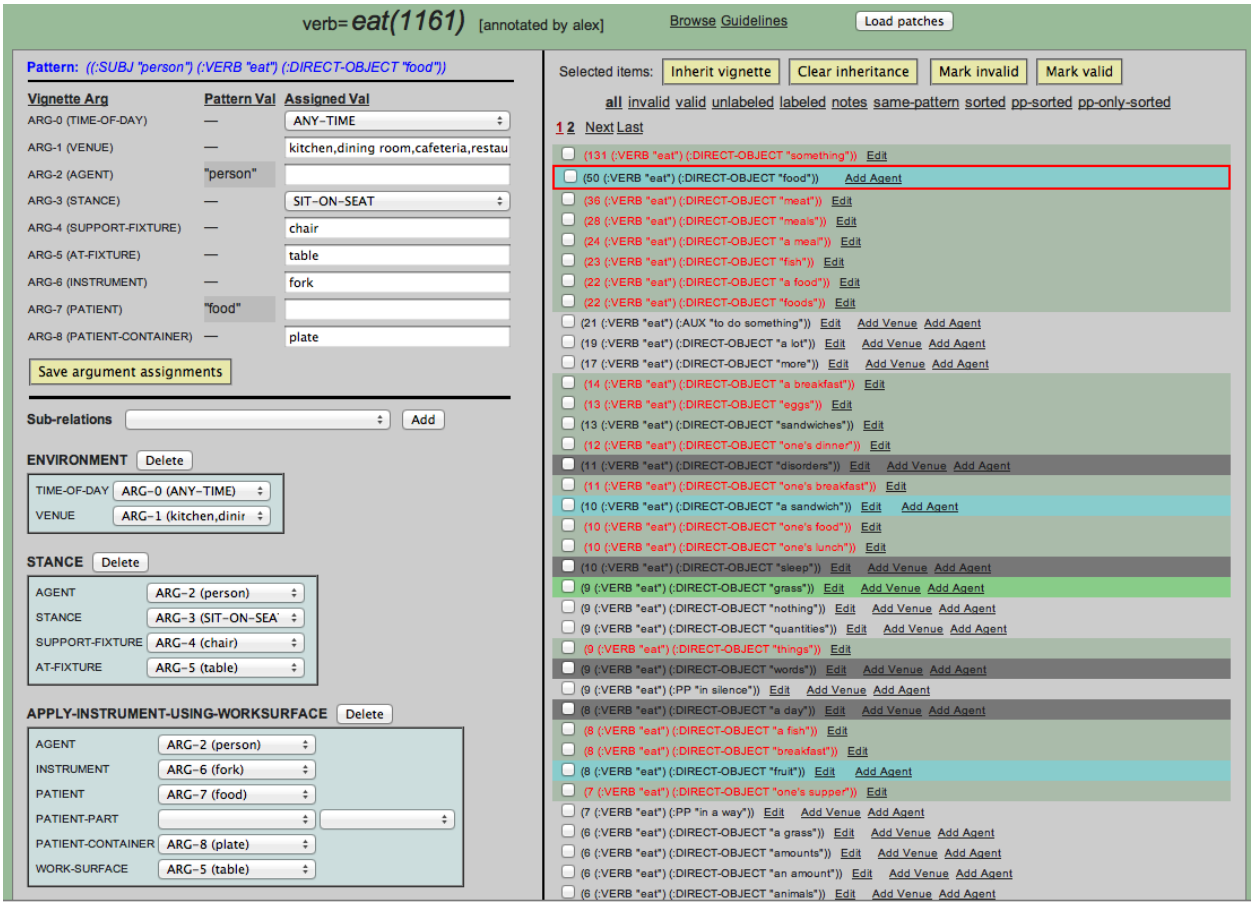


Figure 5.5: Action Vignette Editor

sometimes we get an object not in the ontology. For example, in a new 3D library of plants, many of the plants were specific sub-varieties and hence not in the ontology. So the new concepts were added and the new objects added under those.

Parts. When adding a new object, its parts must be known. This requires a mapping from the internal 3D object part name (which can be any arbitrary, often meaningless, identifier) to a lexicalized concept. If there is no good name, then it will be assigned to an *anonymous* concept type. The object is then processed to determine the size and bounding box for every part.

Affordances. Most objects have spatial affordances that must be defined. The most common is a top surface or a cupped interior region. These are defined with the 3D Surface Annotator Tool (Section 5.6.6). Affordance plus part information is used to make the objects'stand-in vignette.

Other properties. All properties associated with the object are made as assertions into the knowledge-base. All objects have a real-world size that should be specified (though the object can default a size through the ontology from its supernodes). These can vary by object. For example, a cartoon dog character may declare that it IS-A *cartoon* and that it REPRESENTS a *dog*. Some objects might have a length axis or segmentation property that needs to be defined. See Chapter 3 for different graphical properties.

5.6.6 3D Surface Annotation Tool

This tool allows sample points on salient surfaces of 3D objects to be marked, for example the top surface or an interior surface. This interface is built on top of the WordsEye website (Section 9.5). To use it, the operator just needs to type in the text to display the object and then click on the object's surface. The surface sampling works by utilizing WordsEye's internal "hit buffer" that records the 3D XYZ coordinate for every pixel whenever a scene is rendered. This is normally used to aim the camera at any point the user clicks on, but for surface annotation purposes, the coordinate is recorded and associated with an annotator-designated surface type. See Figure 5.6. The resulting sample points are then used to form regions that become part of the stand-in object vignette (Section 2.4.6) associated with the given object.

5.7 Conclusion and Future Work

VigNet is a large and growing resource tailored to WordsEye's needs in text-to-scene conversion. Populating the different aspects of VigNet is an ongoing process and is driven by the needs of the full WordsEye system. As new graphical content is added, it is necessary to insert those pieces of content (as concepts) into the VigNet ontology as well as adding other associated meta information and spatial affordances. Sometimes new lexical items are needed and added in the process. Also, as WordsEye's graphics capabilities improve, more vignettes will be added and integrated into the system.

We also plan to allow users to upload their own 3D models and automatically create stand-in object vignettes for those objects. This would require letting them add them into the ontology on the fly. The Surface Annotation Tool (Section 5.6) could then be used by those users to annotate the surfaces of their objects, and thereby define the necessary spatial affordances.

One method for defining composite object vignettes would be to let users of the site do it with language

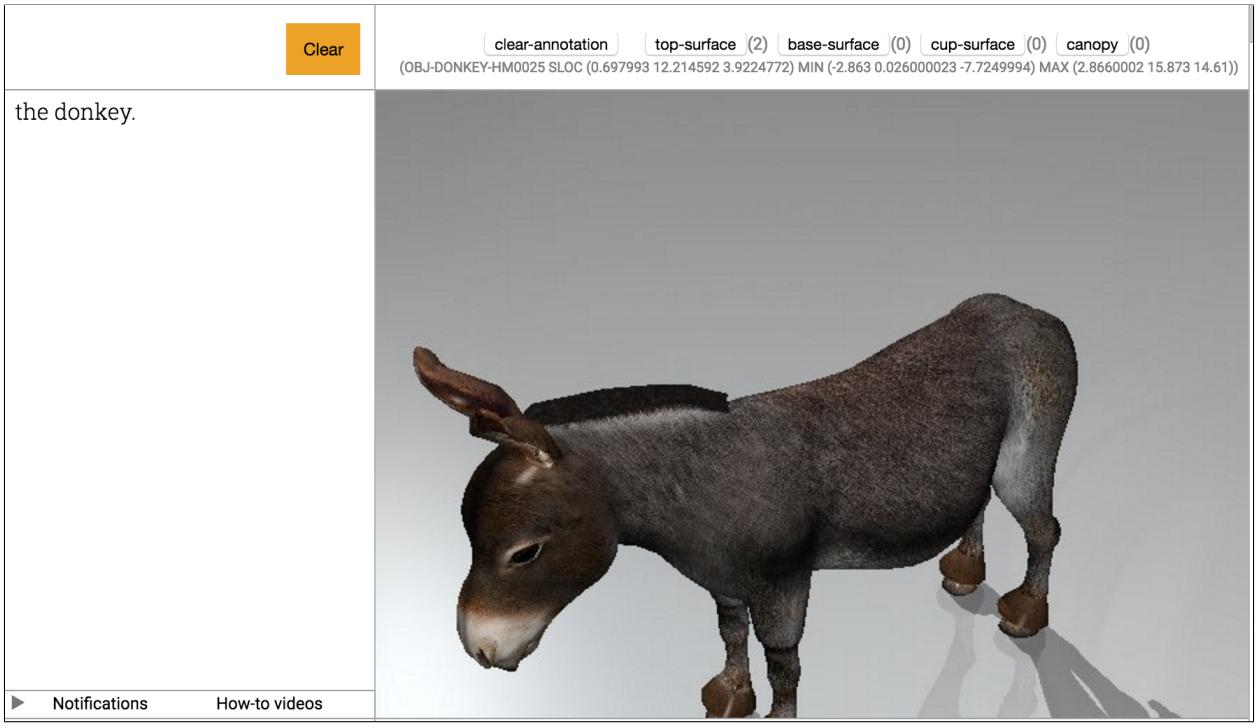


Figure 5.6: Surface Annotation Tool

itself, similar to our own creation of rooms (Section 5.5.1.2) . Several users of the online system have requested that they be able to re-use configurations of objects that they have created. For an example, see characters such as Figure 3.1 (f). The same technique could be used to define other types of vignettes.

Chapter 6

Computational Model and Pipeline

6.1 Introduction and Architecture

At a high level our task can be looked at as a series of meaning preserving translations.

We start with text strings and produce a syntactic structure and then a few stages of semantic representations that roughly correspond to the surface meaning of that text. That surface semantics will usually be vague or ambiguous in some way and must itself then be decomposed to a specific, unambiguous semantic representation. The resolved semantics is translated to graphical primitives and then to a full composed 3D scene graph and ultimately to rendered pixels on the screen.

text ⇒ syntax ⇒ deep syntax ⇒ surface semantics ⇒ resolved semantics ⇒ graphical semantics (vignettes) ⇒ graphical primitives ⇒ 3D scene graph ⇒ rendered scene

The WordsEye system architecture (see Figure 6.1) is structured as a pipeline and contains the following major components:

Linguistic and semantic processing: The input text is converted into a sequence of lemmatized lexical tokens with associated parts of speech and other properties. This is done by looking up each potential token in the VigNet dictionary, which contains all the relevant information. These tokens are then parsed into a phrase structure tree that is converted to a syntactically labeled dependency structure (Section 6.2 and 6.2.6). Coreference resolution is performed to merge references across sentences (Section 6.3). The resulting text

is then fed into a semantic analysis module that maps the lexicalized semantics into semantic relations, does some semantic and contextual coreference resolution, and then decomposes that semantics into specific objects, vignettes, and graphical primitives.

Spatial and Graphical processing: The graphical primitives output by the linguistic and semantic processing modules are analyzed and processed to infer additional constraints, handle spatial reference frame ambiguity, and perform other lower-level graphically-oriented reasoning. A final set of graphical constraints is then produced that is used to construct the scene. This involves loading objects and applying the graphical constraints (such as size, location, orientation, color, and surface texture). The final 3D scene is then rendered to produce a final image.

Databases: The various processing steps rely on two databases. The graphics library contains approximately 3,500 3D objects and 5,500 2D images. These are used when actually constructing and rendering a scene. The VigNet knowledge base contains knowledge used by all other processing steps. In addition VigNet contains knowledge about the objects in the graphical database (such as the size of the objects, their type, relevant spatial parts and the affordances on those objects).

External interfaces: In addition to the core system, the system supports a set of external interfaces that can be used to invoke the processing pipeline and display its results. These include a command line interface, an extensive web API providing access to all aspects of the running system including the gallery of user-generated scenes, a forum, and social media functions such as comments, likes, and hashtags associated with scenes. The API also allows users to upload and name their own photos to use as texture maps. The API is used by both the browser-based web interface and the mobile app.

6.2 Lexical Analysis and Parsing

In this section we discuss the first stages of language processing: lexical analysis and parsing.

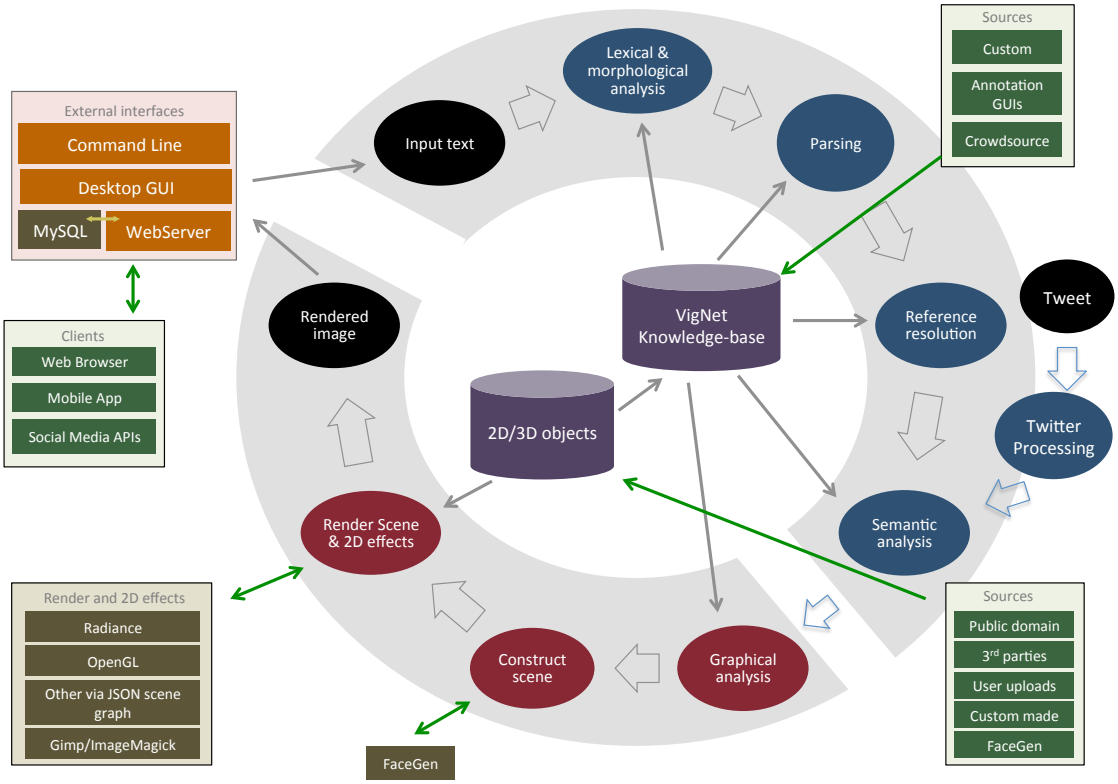


Figure 6.1: WordsEye architecture: Computational flow (the circular path), knowledge base and graphical libraries (center), external interfaces along the periphery

6.2.1 Lexical Analysis

Linguistic processing starts by segmenting the text input into separate sentences and then tokenizing the input text. The lexical tokens are all found in our dictionary. Special logic is used to pre-parse contractions and abbreviations such as *a.m.* or *p.m.* that contain periods. In addition, care is taken to handle quoted regions as a unit which can have embedded punctuation.

6.2.2 Parser

WordsEye uses a custom CYK parser [Younger, 1967] that is augmented with feature unification. This allows agreement between verbs and arguments and determiners with their arguments. The WordsEye grammar consists of about 700 production rules. The hand-built grammar provides several advantages for our

application over an off-the-shelf parser. The parser also allows each rule to have arbitrary number of right-hand-side nodes of any type (both terminals and non-terminals).

6.2.3 Grammaticality

Checking for correct grammar is important for educational applications where students are using the text-to-scene system to improve language skills. Enforcing grammaticality can also significantly reduce the number of ambiguous parses. WordsEye's parser handles this by having explicit agreement tests in the grammar. It does this by propagating lexical features during every production rule. For example, if a particular noun is plural, that will be tested for any agreement with determiners. And the resulting noun phrase will inherit those properties. The noun phrase can then be tested for agreement with the verb.

For example, this rule specifies the NP and VP must agree to form the sentence and that the NP must be in nominative case.

```
;; NP must be in nominative case. (don't want "me" etc as subjects)
(s ((np :subject t :match (:case (any :unspec :nominative)))
    (vp :head t :match (:conj :inflected) :promote (:sentence-type :voice)))
   (:test subject-verb-agree-p))
```

6.2.4 Measure Terms

In describing scenes, it is crucial to understand distances, sizes, and other measure terms. And it is likewise important to correctly interpret the use of those terms. The problem is that, while terms such as *three feet* or *2 inches* are noun phrases, they often function as adverbs to modify prepositions or verbs. Parsers can easily get this wrong.

6.2.4.1 Stanford Parser Examples

To show the problems handling measure terms, we look at a set of parses produced by the Stanford Online Parser [Manning *et al.*, 2014]. In all cases, the phrase *3 feet* is interpreted as a direct object rather than an adverbial modifier.

a) **Syntactically Ambiguous:** *The boy ran 3 feet under the table* or *The boy ran 3 feet forward*. These cases are syntactically ambiguous since *run* can be transitive (as in *John ran the company into the ground* or *The cowboy ran the cattle across the open range*). However, it would be semantically nonsensical for the verb *run* to be used transitively with the direct object of *3 feet*.

```
(ROOT
  (S
    (NP (DT the) (NN boy))
    (VP (VBD ran)
      (NP (CD 3) (NNS feet))
      (PP (IN under)
        (NP (DT the) (NN table))))))

(ROOT
  (S
    (NP (DT the) (NN boy))
    (VP (VBD ran)
      (NP (CD 40) (NNS feet))
      (ADVP (RB forward)))))
```

b) **Syntactically and semantically ambigous:** *The boy saw 3 feet under the table*. This case is truly ambiguous, and the direct object interpretation is acceptable.

```
(ROOT
  (S
    (NP (DT the) (NN boy))
    (VP (VBD saw)
      (NP (CD 3) (NNS feet))
      (PP (IN under)
        (NP (DT the) (NN table))))))
```

c) **Unambiguous:** *The boy slept 3 feet under the table*. In this case, the phrase is unambiguously an adverbial modifer but is still classified as a direct object by the Stanford Online Parser.

```
(ROOT
  (S
    (NP (DT the) (NN boy))
    (VP (VBD slept)
      (NP (CD 3) (NNS feet))
      (PP (IN under)
        (NP (DT the) (NN table))))))
```

6.2.4.2 WordsEye on Measure Terms

WordsEye handles this issue by introducing lexical features for MEASURABLE-TERM? (for nouns) and MEASURABLE? (for relational terms such as verbs and prepositions). These feature allow different prepositions and verbs to accept measure terms as modifiers in the grammar. This, in turn, helps produce the desired parse and dependency structure. For example, *above* is MEASURABLE because an object can be any distance above another (*The balloon is 3 feet above the chair*). On the other hand, *during* is not measurable (* *the intermission is 3 feet during the concert*). Hence the parser will accept *three feet above* but not *three feet during*.

Measurability and other lexical features percolate upward via production rules from prepositions to prepositional phrases. This allows measurability to be tested between verb and its prepositional phrase arguments. For example, a sentence such as *the man stepped on three feet during the soccer game* parses *three feet* as a direct object while *the bird flew three feet above the tree* parses it as an adverbial modifier on the preposition *above*.

The grammar incorporates this knowledge via feature constraints on its production rules. For example, this rule forms a modified preposition (IN2) by merging a measure term with a raw preposition.

```
IN2 => (NP-MEASURE) (IN :head t :match (measurable (any t :both)))
```

The rule above (and others like it) are driven by lexical information in VigNet. Features are found on lexical tokens. For example, the part of speech token for *in front of* is shown below:

```
(lookup-pos "in front of") ⇒
((NLP::IN
  :WORD "in front of"
  :NOT-OF T
  :MEASURABLE-P T
  :LEMMA "in front of"
  :NODE (#<PREPOSITION: WE.IN-FRONT-OF.PREP
            ("in front of" "in the front of" "to the front of"
              "at the front of" "at the front side of")>)))
```

6.2.5 Grammar Rules

Grammar rules have the following elements:

- Arbitrary number of right-hand-side nodes (both terminal tokens and non-terminals)
- Optional items (denoted by "ANY" and "?"). These are pre-processed into all possible combinations.
- Features to match (via MATCH)

- Logical operators: AND, OR, NOT to apply to feature matching
- Features to percolate from the right hand side (via PROMOTE)
- HEAD designation with optional features to assert (via DECLARE)
- Additional tests to check for agreement or if valence patterns match those in knowledge base

The following rule, for example, is used to form a verb phrase from a simpler verb phrase and a prepositional phrase and an optional adverbial following the prepositional phrase. The rule will match a more basic verb phrase (VP1) that contains adjuncts such as adverbs, modals, and auxilliaries but no true arguments such as prepositional phrases or direct objects. In testing what VP1s can match, various properties of the inflected verb are examined, with features tested using logical operators.

```
(vp ((any (vp1 :head t ,@v-all
               :match (:intransitive-ok t
                       :voice (any :unspec :active)
                       :verb-type (not :linking)))
          (vp1 :head t ,@v-all
               :match (:voice :passive :verb-type (not :linking))))
     (prepp* :declare (:v+prep t))
     (? (advp :match (:sentence-adv t))))
   (:test valence-patterns-agree-p)))
```

6.2.6 Using the Knowledge Base

Like all other aspects of the system, the WordsEye parser draws upon the VigNet knowledge base. It does this primarily through the lexical and conceptual knowledge encoded in VigNet. Every lexical item is annotated with part of speech tags as well as other lexical features of interest to the parser. These include whether a term is measurable, whether a verb is intransitive, transititive, or ditransitive. All lexical items can also be tied to graphical entities, either directly or indirectly. Any information in the knowledge base can in-principle be utilized as a feature in the parser. In practice, we currently use a limited set of lexical features.

As an example example, this production rule turns a possible measure noun-phrase (eg *3 feet*) plus a raw preposition into a prepositional phrase:

```
(in2 ⇒
  (np-measure (in :head t :promote (:not-of) :match (:measurable-p (any t :both)))
```

Note that each rule designates the head. This allows the phrase structure tree to be automatially and

reliably converted to a syntactic dependency structure (which is later converted via semantic translation rules into a semantic relational structure).

6.3 Reference and Coreference

6.3.1 Referring Expressions

When text refers to objects, we must determine if two references are to the same object or not. If they are, then any references to them (including attributes) are merged. In this section we describe the approach we take to identify and merge coreference. Coreference resolution and the interpretation of referring expressions is a complex and fascinating problem [van Deemter, 2016] requiring a mixture of semantics, pragmatics, and context and logical inference. Our procedural approach is fairly simple and intuitive (based on Gricean maxims [Dale and Reiter, 1995]) and covers the basic cases needed reasonably well. Computational systems for coreference often work in very different domains and use machine learning methods to take a grab-bag of features [Ng and Cardie, 2002].

6.3.2 Accuracy Versus Predictability

For a text-to-scene authoring system, the user is most concerned with being able to achieve the effects they desire. In order to position objects relative to each other and change their colors and other properties, it is necessary to refer to the same objects multiple times. And since there will often be more than one object of the same type in a given scene (e.g. two different tables in a room), the system must figure out if the same or a different object is being referred to. In doing so, we have two criteria: first, the system should try to *accurately* mimic what a user would expect their text to mean; secondly, the system's behavior should be *predictable* (Section 11.1). This second criteria allows the user to formulate their input in a way that the system can easily and consistently process it. We see, therefore, that accuracy alone is not the only criteria. A system that is more accurate on text in the aggregate may be less accurate, or certainly less efficacious, on text from a particular user who has learned to use the given system. Thus, the easier the user can predict the system's algorithm the easier it will be for them to use the system with minimal effort. Or to put it another way, it can be easier to learn the mistakes that system makes if they are done in a regular and predictable manner. We perform an evaluation of the accuracy of the scenes produced with WordsEye versus Google Image Search in Chapter 8.

6.3.3 Lexical Features and Processing

Our reference resolution model, originally created by Richard Sproat in older versions of WordsEye [Coyne and Sproat, 2001] and slightly extended in this work, takes into account several features: ANIMACY, GENDER, HUMAN, NUMBER, DEFINITENESS, POSITION-IN-SENTENCE (subject or object), and SENTENCE-NUMBER. In addition, the system chains results, so once a reference is resolved from sentence 3 to sentence 2, it can then be resolved back to sentence 1.

The system uses the DEFINITENESS feature in determining if a reference should be merged, based on a given-new distinction. An indefinite article is assumed to introduce a new entity reference while a definite article could be either given or new.

In identifying whether two references are the same, the system also considers attributive adjectives. For example, [*the red dog is on the table. the green dog is big.*] would imply two dogs since *red* and *green* do not match. The system does not currently merge entities based on predicative adjectives (e.g. *The dog is red. The red dog is on the table*). This could be done in some cases, but is less predictable.

The system treats indexicals (e.g. *the first dog*) purely as attributive identifiers and hence will merge multiple instances. So, [*the first dog is small. the second dog is blue. the first dog is on the table.*] will put the small dog on the table. It does this without interpreting what *first* actually means. This is useful behavior, but only a partial solution. For example, [*The red dog is big. the blue dog is small. the first dog is on the table.*] should put the red dog on the table, even though it was never identified explicitly as being the *first*.

In addition to one-to-one merging of references, the system handles the merging of singular references into collections. For example, in [*The CAT AND THE DOG are on the table. The CAT is green.*], the entity *cat* will merge into the collection *cat+dog*. Likewise, for [*The CAT is on the table. The DOG is green. The ANIMALS are asleep.*], both *cat* and *dog* will merge into the collection for *animals*.

In order to merge singular entities into collections, which may or may not be single lexical terms, the system first (in the earlier lexical processing) identifies and forms collection entities either from individual terms or from conjunctions. Collections are then "grown" based on the reference resolution.

There is currently no support for handling reference resolution with quantification. It would be desirable for the system to be able to process text like [*10 chairs are in the room. Some of them are blue*].

WordsEye currently processes anaphora and other coreference across sentences but not within a sentence. Since intra-sentence coreference is commonly used to reference parts (e.g. “The knife was standing

on its handle") and our support for parts is limited by the information on the 3D objects, this has not been a serious impediment. Nonetheless, it is an important capability that the current system should be able to handle. It is also an area where the grounded knowledge in VigNet (of parts when they exist) can be used to resolve references.

6.3.4 Using the VigNet Ontology in Reference Resolution

In addition to supplying lexical features, VigNet is used in reference resolution to help decide whether to merge or not based on the types in the ontology. For example, [*The dog and cat are sleeping. The animals are on the chair.*] would merge because of knowledge in VigNet that dogs and cats are animals.

World knowledge can also be used, in the future, to merge part references. For example, for *The couch rested on its legs*, the reference of *its* is the *couch* since a couch has legs. A similar type of contextual reference resolution is currently supported in the semantic processing stage for vignettes. For example, [*The dog is in the room. The left wall is red*] links *wall* with the *room*. In this case, the two entities are not merged, but the one is designated as part of the other.

6.3.5 Existential *There* and *It*

Some pronouns can be used existentially, with no definite reference. For example, sentences like [*It is raining*] or [*it is 10 a.m.*] assert propositions about the global state rather than any particular earlier referenced item. We have special-case logic to avoid merging references for such assertions.

6.3.6 Test Cases

See Table 6.3.6 for a set of test cases produced to serve as guideposts to both define the desired behavior and to test the system.

6.3.7 Other Uses of Reference Resolution

In addition to using reference resolution to interpret the text for depiction purposes, WordsEye also incorporates it into the graphical user interface presented to the user. The template interface (also described in Section 9.3 and shown in figure 9.6) utilizes reference merging to present to the user a single entry per entity to modify. This can reduce the work needed to change one object for another and help reduce error in that

the blue cats are strong. it is on the table.	Don't MERGE-INTO it/cats
The cats are on the table. it is red.	merge it/table)
The cat is on the table. it is red.	merge it/cat
"the BLUE CATS are on the table. the RED CAT is big."	no merge
the CAT AND THE DOG are on the table. the CAT is green.	MERGE-INTO cat/dog-cat-collection
The gray cat is strong. the cats are tall.	merge-into GRAY CAT/CATS
there is a dog sitting on the top of the chair. she is 3 feet tall.	merge SHE into DOG
the dog is on the desk. dog is blue.	merge (probably want)
the beautiful cat is strong. the red cat is on the table.	don't merge
the fadfa cat is big. the abdfadbf cat is strong.	don't merge
the red cat is big. the blue cat is big. the cats are on the table.	red cat and blue cat MERGED INTO cats
the cats are red. the cats are big.	merge
the animals are red. the cats are big.	merge?? probably not
the red cats are on the table. the blue cat is big.	no merge
the 3 foot tall dog is blue. the 4 foot tall dog is strong.	**** don't merge
the GROUND is blue. the GROUND of the restaurant is pink.	**** don't merge
the dog is on the table. it is on the rug. it is blue.	merge both ITs with DOG
the cat is big. the table is big. the cat is on the table. it is red.	cat is red
the red cat is big. the table is big. the red cat is on the table. it is purple.	merge cats for consistency...?
the cat is big. the big cat is tall.	****?? merge??
the red cat is tall. the large cat is small.	don't merge
the dogs are large. the animals are red. the greyhound is nice.	**** do merge. what with what? greyhound into both dogs/animals?
the dog is on the TABLE. the cat is on the TABLE. IT is red.	it=cat, cat is red
the TABLE is behind the chair. the book is on IT. IT is blue.	*** it=table, table is blue
the cats are on the cube. the blue cat is big. the red cat is big.	merge red, blue cats into collection
the CATS are on the table. IT is blue.	merge it/table
the BOY AND THE GIRL are tall. HE is blue.	merge he/boy
The red cat is behind the cat. the red cat is to left of the cat.	merge cat/cat, red-cat/red-cat
the barn is behind the cow. the cube is left of the barn. it is purple	cube is purple
the blue cat. the cats. the purple cat	merge both into cats.
the cats. the purple cats. the green cat.	merge green cat into cats but not purple cats.
the purple cat. the cats. the purple cat.	one collection with one – purple cat.
the cats. the purple cat. the purple cat.	one cat in one collection
the cats. the purple cat. the purple cats.	two collections. same cat in both.
the purple cat. the cats. the green cat. the cats.	****one collection with two cats
the box with a marble in it.	****do internal merge
the box has a marble in it.	****do internal merge
the girl with a box on her.	****do internal merge
the cones are 2 feet tall. they are 2 inches wide	merge cones/they
the cats are blue. they are strong.	merge cats/they
The cat is next to the dog. they are red.	merge
the cat is red. the dog is blue. they are strong.	merge dog, cat into they
the cat and the dog are on the cube. they are blue	merge
the cube is on the cats. they are blue.	cats are blue? probably
the dog and the cat are on the table. they are pink.	they=dog+cat, not table
the three dice are on the table. the dice are big.	merge
the huge dice are on the table. they are big.	merge
the huge dice are on the table. the three dice are big.	don't merge
The cat is big. the blue cat is nice.	Don't merge.
The cats are big. the blue cat is nice.	*****Do Merge.
the ball is on the table. the table is under the tree. the lion is under the table. it is facing the table.	single table

Table 6.1: Reference resolution criteria examples

process. For example, if a particular object (e.g. a dog) is referred to five times in the input text, the user can change the first instance of *dog* to *cat* and all instances will change.

This process of replacing references in such a manner adds a few new wrinkles and problematic cases. Since the system uses the whole noun phrase (excluding determininers) to identify one object from another, the system must be careful in replacing subsequent references. If a plural is modified, then individual elements of that collection should not be modified.

6.3.8 Future Work in Reference Resolution

There is much that could be done to improve reference resolution. What follows is a partial list:

1) There are currently no inferences performed to determine if two references are the same. For example, in [*The book is 3 feet below the table top. the sphere is on the table. the dog is right of the sphere. the rightmost object is red*] we would want to determine which of the objects is rightmost. This would require simulating the scene layout to determine the correct reference.

2) Gradable attributes and object default sizes could be used to merge references. For example, in [*The beagle is next to the great dane. the small dog was sleeping*], we would want the *small dog* to merge with *beagle*.

3) Incorporating partial results: When users are creating scenes with the system, it is natural to refer to the scene they see in the text that they add. The choices made by the system in terms of object selection can affect the interpretation of new text. This is a relatively common occurrence for users of the system and represents and interesting problem that goes beyond traditional methods for analyzing text. For example, assume the user first typed *the man is next to the table* and the system chose a 3D model of Santa Claus for the man. The user might then enter a second sentence saying *Santa is 10 feet tall*, thinking that they are referring to the same object. In this situation, the system will interpret *Santa* as a new reference, since there was no lexical match. This would be generally correct if the text was entered all at once, but in the incremental fashion of entering text, then seeing the scene, and then entering more text, one might expect the system to interpret the text differently. But this has problems too. Let us assume the system did merge Santa with the man, and the user then finished the scene. Another user who then read the text and looked at the scene might be puzzled as to why there was only one object and not two. This suggests that the interpretation of references, and text more generally, depends on the desired interaction style.

4) Quantification: One would like the system to be able to process text like *10 chairs are in the room.*

Some of them are blue.

5) Part merging: In *The house with the red door* we would want to link *door* to *house*. This is more of a semantic interpretation issue, but involves references, so we mention it here.

6.4 Semantic Analysis

At this point the system has on its hands a syntactic dependency structure with lexical references merged. Its next task is to transform the syntax to semantics and then the semantics to vignettes or other graphical primitives. It performs both sets of transformations using the same pattern matching rule mechanism. The format of the rules is identical for converting syntactic dependencies to semantic relations and for converting semantic relations to other semantic relations. This is because the structure of the both the rules themselves and the input and output data is uniform.

6.4.1 Pattern Matching Rules

In this subsection, we first describe the basic format of the semantic translation rules rules before moving on to the different stages of processing that use those rules. Rules are of the form:

```
(:input (i-relation1 i-relation2 ...)
 :types ((variable1 type1) (variable2 type2) ...)
 :kb-match (kb-relation1 kb-relation2 ...)
 :output (o-relation1 o-relation2 ...))
```

Input, output and KB relations are of the form:

```
(<relation> :keyword1 value1 :keyword2 value2 ...)
```

The variables are logic variables (denoted with a leading "?") and are constrained to match the specified TYPES and the designated input values. They can appear in an arbitrary number of input or output relations. Types can either be concepts or logical expressions referencing concepts. For example [:types ((variable1 (or donkey.n horse.n)))] enforces that the variable must either be a horse or donkey.

If KB-MATCH relations are specified, then the rule queries the database and matches any relations, binding the logic variables in the process. For example, this next rule queries the knowledge base to find out the default color and reflectivity values for a given substance and replaces the substance relation with explicit color and reflectivity primitives using those values. Note that it also uses a logical expression when checking

for one of the two possible input substance relation types. In the next sections we will look at more rules in action.

```
(:input ((?rel :entity ?subject :value ?substance))
 :types ((?rel (or we.substance-of.r we.surface-covering.r)))
 :kb-match ((we.reflectivity.r :entity ?substance :value ?reflectivity)
            (gfx.rgb-value-of.r :entity ?substance :rgb-value ?rgb))
 :output ((gfx.rgb-value-of.r :entity ?subject :rgb-value ?rgb)
          (we.reflectivity.r :entity ?subject :value ?reflectivity)))
```

6.4.2 Syntax to Semantics

The system must now convert the lexical dependencies to a semantic representation. It does this with a set of hand-constructed rules that take dependency structures and their arguments as input and then output semantic structures.

For example, given the sentence *the blue cat is 3 feet tall*, a multi-step process is invoked where *3 feet tall* translates into the TALL.R semantic relation and then translates the lexical dependent *blue* into the BLUE.R relation. Those are later translated into domain-specific semantic relations for GFX.COLOR-OF.R and GFX.SIZE-OF.R.

```
patterns=((?rel (:measure ?measure)) (?measure (:cardinality ?cardinality)))
input=((#<3-LexEnt (noun) "cat">
            (:ATTR-MODIFYING #<2-LexEnt (relation) "blue">)
            (:ATTR-PREDICATIVE #<7-LexEnt (relation) "tall">))
       (#<6-LexEnt (unit) "foot" PLURAL>
            (:CARDINALITY #<5-LexEnt (cardinal) "3">))
       (#<7-LexEnt (relation) "tall">
            (:MEASURE #<6-LexEnt (unit) "foot" PLURAL>)))
result=((#<3-LexEnt (noun) "cat">
            (:ATTR-PREDICATIVE #<7-LexEnt (relation) "tall">)
            (:ATTR-MODIFYING #<2-LexEnt (relation) "blue">))
        (#<7-LexEnt (relation) "tall">
            (:MEASURE-UNIT #<6-LexEnt (unit) "foot" PLURAL>)
            (:MEASURE-CARDINALITY #<5-LexEnt (cardinal) "3">)))

patterns=((?entity ((:attr-predicative :attr-modifying) ?rel))
          (?rel (:measure-unit ?unit) (:measure-cardinality ?cardinality)))
tests= ((?rel :we.size.r))
input= ((#<3-LexEnt (noun) "cat">
            (:ATTR-PREDICATIVE #<7-LexEnt (relation) "tall">)
```

```
             (:ATTR-MODIFYING #<2-LexEnt (relation) "blue">))
        (#<7-LexEnt (relation) "tall">
             (:MEASURE-UNIT #<6-LexEnt (unit) "foot" PLURAL>)
             (:MEASURE-CARDINALITY #<5-LexEnt (cardinal) "3">)))
result=((#<3-LexEnt (noun) "cat">
             (:ATTR-MODIFYING #<2-LexEnt (relation) "blue">))
        (#<7-LexEnt (relation) "tall"> (:ENTITY #<3-LexEnt (noun) "cat">)
             (:UNIT #<6-LexEnt (unit) "foot" PLURAL>)
             (:VALUE #<5-LexEnt (cardinal) "3">)))

// create blue (color) as a relation
patterns=((?entity ((:attr-modifying :attr-predicative) ?rel)))
input= ((#<3-LexEnt (noun) "cat">
             (:ATTR-MODIFYING #<2-LexEnt (relation) "blue">))
         (#<7-LexEnt (relation) "tall">
             (:ENTITY #<3-LexEnt (noun) "cat">)
             (:UNIT #<6-LexEnt (unit) "foot" PLURAL>)
             (:VALUE #<5-LexEnt (cardinal) "3">)))
result=((#<7-LexEnt (relation) "tall">
             (:VALUE #<5-LexEnt (cardinal) "3">)
             (:UNIT #<6-LexEnt (unit) "foot" PLURAL>)
             (:ENTITY #<3-LexEnt (noun) "cat">))
         (#<2-LexEnt (relation) "blue">
             (:ENTITY #<3-LexEnt (noun) "cat">)))
```

6.4.3 Object Selection

In order to fully resolve the scene we need to select 3D objects and images. Often the semantic decomposition process will determine if an object is an image or 3D object. Once that is known, the primary constraint for a 3D object is its type and for images is what it portrays or represents. The simplest case is when the input word specifies the object type. For example, if the input text specified *the inventor*, we will then constrain the type of the object to be INVENTOR.N. That type currently includes concepts for Thomas Edison, Nicola Tesla, and Robert Fulton. Of those, we only have a 3D object (a composite object vignette) for Edison. So he will be chosen.

If no objects exist for the given type, a fallback object is chosen instead. This is done by finding objects in related concepts. We use a simple metric of concept distance (counting IS-A links). This could be improved by adding weighting based on node type or including other external information into the calculation such as co-occurrence in corpora. A secondary fallback is to choose a 2D image representing that same node and

to texture map that onto a cube and use the textured cube as a stand-in. If no related 2D image or 3D obect is found, or if the input word is not in the lexicon, then 3D text for the word itself is used in the scene. See Figure 9.7

6.4.3.1 Selectional Constraints

In some cases, additional constraints relating to the choice of object are imposed by an adjective or other modifier in the input text. For example, there is no lexical entry for *Japanese-style teacup* but the knowledge base does have an assertion stating that a particular 3D object is of Japanese style or origin. The system will use the input constraint to narrow down the set of possible objects. Another example where selectional constraints are applied is with parts. For example, *the head of the cow is blue* will force the selection of a 3D cow that has a designated head as 3D part (not all of them do). See Figure 4.1.

Other adjectives such as colors or sizes are not used to constrain object selection. So we distinguish between *selectional constraints* that affect what object is selected versus *self-enforcing constraints* such as colors or sizes that cause the object to be modified so as to make the constraint true. For *the blue table*, the system directly enforces the constraint by making any chosen table blue. Attributes are only used as selectional constraints when they cannot be directly enforced.

We note that sometimes this strategy is non-optimal. For example, if the input text specifies *the huge dog*, the system will choose any dog in an unconstrained manner, even though it could in theory check to see if the default size of the dog is already huge (or close to it). As a result, it might pick a small dog (e.g. a dachshund) and make it huge rather than select one (a great dane) that is already huge.

Reference resolution ties tightly into object selection. A sentence like *the man is next to the door* can be interpreted as a man next to a standalone door. But it could also, more naturally, be interpreted as a man standing next to a door that is part of a house or other building. The latter interpretation is even more likely if the input text was *The man is near the bathroom. He is next to the door*.

References to parts of mentioned vignettes are resolved as follows: The hypothetical part will be linked to a whole if a compatible whole is mentioned and if that whole is a vignette that advertises its parts (currently, this is done by the basic room vignette). The standalone part reference case is trickier since many possible objects (or vignettes) could be evoked, and bringing them in would be a bigger (more disruptive) inference to make. Also, in general, there needs to be a way for a user of the system to get a door on its own without a whole room or building tied to it.

6.4.3.2 Parts, Wholes, Groups, Members, Regions

Sometimes the semantics specify a mereological relation. This will impose a new subsidiary entity. It is crucial that the system clearly interpret the differences between these types. The system currently has three main mereological types:

Parts: A *part* is linked to a *whole* via the WE.PART-OF.R relation. This can be imposed by text such as *the cow's head is blue*. See Figure 4.1.

Collections: A *member* is linked to a *group* via the WE.MEMBER.R relation. This can be imposed in text by plurals or any mention of cardinality as well as a reference to words like *group* or *row*. The VigNet ontology is queried by the semantic interpretation and decomposition rules to make these determinations. See Figure 4.1. Sometimes groups come "pre-loaded" with specific member objects (*the dog and the cat*) and sometimes there is just a cardinality (*the three blue dogs*). When only a cardinality is specified, the system creates new entities.

Regions: A *region* is linked to an *entity* via the the GFX.SUBREGION.R relation. Regions are used when text specifies sub-areas of objects, as in *the vase is in the middle of the table*.

There are many potential options to some of the above. For example, crowds are a type of group. If the graphical subsystem had a way of simulating crowds and supplied a primitive for it, then the semantic decomposition could map directly to that.

6.4.3.3 Abstract Entities

Sometimes a spatial relation will specify an abstract entity. For example, in *The dog is facing left*, the direction *left* is not a physical object. Instead, it denotes a direction. So in these cases we do not choose a graphical object.

Colors are another common type of abstract entity – they do not themselves exist as physical obects but instead denote properties of those objects. When we say [*the couch is bright blue*], then *bright* modifies *blue*. And if we say [*the couch is very bright blue*], then the intensifier *very* modifies *bright*. The semantic model also handles numeric modification of colors and other scalar attributes. For example, *the 20% opaque cube* will modify the transparency of the cube and represent it as follows:

```
(#<NewEnt-10 (RELATION): we.transparency.r>
     (:ENTITY #<4-LexEnt (noun) "cube">)
     (:RATIO #<2-LexEnt (ratio) "20%">)))
```

6.4.4 Supplying Defaults

Before we can decompose high-level semantics into vignettes, we must first have settled on the semantics we want to see. High-level semantic relations have many slots that may not all be filled. For example, *the man fried the fish* could invoke a vignette for APPLY-MACHINE-TO-PATIENT which would involve a stove (as the machine), and a frying pan on the PATIENT-AREA affordance of the stove, and the fish in the frying pan. Or instead, the high-level semantics could first resolve that the INSTRUMENT was a frying pan and then a separate translation rule would choose the appropriate vignette.

Our general strategy with this, in tests so far, is to fill in slots at the high-level semantics first so that the functional-level vagueness is resolved. And then we separately convert the resolved semantics to vignettes. We could, instead, directly convert the partially resolved high-level semantics directly into vignettes. In that case, the translation to the vignette would in one step resolve both the high-level (functional) semantics and the graphical semantics.

6.4.5 Semantic Ambiguity and Vagueness

We define *ambiguity* as a measure of the number of branches in a translation or interpretation process. We say a word or sentence is ambiguous if it leads to multiple semantic interpretations. Similarly we can say that semantics is ambiguous if it leads to multiple decompositions into graphics.

For example, in Section 3.3.1 we presented several plausible alternatives for graphic interpretations of *John broke the bottle*. In translating that sentence from language to semantics, there is little ambiguity – it has a clear propositional meaning. However, in going from that semantic-level meaning (function) to graphics (form), it becomes very ambiguous. We do not know what it really means.

When WordsEye chooses an object or vignette for the scene, it is in effect disambiguating the meaning. In doing so, the system will consider a set of all possible objects of the specified type and then choose one randomly. In some cases, the type of object will be constrained by constraints specified in the input text (e.g. that the object has a certain part). The user is then allowed to override the system's choice of object and select their own (via an image thumbnail) using the scene object browser utility available on the site. Once an object has been chosen in that manner, the system will remember the choice and continue to use that same object until the user makes a different choice or clears the scene or the sentence that produced that object. As a result, the level of ambiguity of the text changes once it has been entered and the scene displayed. But newly entered text, unless it references the same objects, will be ambiguous since the graphical choices have

not yet been made.

We define *vagueness* as a form of ambiguity where the choices are more numerous and closer together and less distinct, more like a cloud or linear scale than a tree structure. Gradable attributes such as *big* are inherently vague because the exact value could be one of many from a continuous range of values.

Handling ambiguity and vagueness are important issues in text-to-scene conversion. We have seen how we use semantic features in translation rules to try to pick the best interpretation, and how we order the rules so that the most specific have a chance to run first. As a result we can sometimes make a sub-optimal selection. Another approach would be to add weights to rules based on fit, run them in parallel, and follow the overall highest scoring branch. That remains an interesting avenue of investigation.

6.4.6 Semantic Decomposition

We have already seen (Section 6.4.1) an example of semantic decomposition of the WE.SUBSTANCE-OF.R relation into separate relations GFX.RGB-VALUE-OF.R and WE.REFLECTIVITY.R. And an example of decomposition for an action to graphical primitives is shown in Section 6.9

We show, here, a couple more examples. The first example is the defining form and semantic decomposition rule used by all character vignettes (Figure 2.5).

```
Defining Form:
  (defvignette human-as-head-on-body.vr ()
     :core-args ((:self human.n) (:head human-head.n) (:body human-body.n)))

Translation Rule:
  (:input ((:human-as-head-on-body.vr :self ?object))
   :types ((?object human.n)
           (?head human-head.n)
           (?body human-body.n)
           (?y y-axis.n))
   :kb-match ((we.part-of.r :whole ?object :part ?head))  ;; this binds ?HEAD
   :output ((gfx.connect-attachment-points.r :figure ?head :ground ?body)
            (we.spatial-grouping.r :group ?object :element (?head ?body)
                :alignment-axes "xz"
                :support-axis "y")
            (gfx.orientation-with.r :figure (?head ?body) :ground ?object)
            (we.balloon-elements.r :group ?object :element (?head ?body)
                        :balloon-together t
   :axis ?y)))
```

The semantic decomposition phase is responsible for interpreting all aspects of the semantic translation process. This includes spatial frames of reference. In this example, the general-purpose (ambiguous) WE.ON.R relation is decomposed to a GFX.ON-FRONT-SURFACE.R primitive. In doing so, it specifies that the wall establishes a frame spatial frame of reference.

```
(:match ((?rel :figure ?fig :ground ?ground))
 :types ((?rel we.on.r)
         (?fig :f-wall-item.n))
 :kb-match ((gfx.preferred-surfaces.r :entity ?ground :surface "front-surface"))
 :output ((gfx.on-front-surface.r :figure ?fig :ground ?ground
                 :reference-frame ?ground)
          (gfx.preferred-surfaces.r :entity ?ground :surface "front-surface")))
```

6.5 Adding Inferred Graphical Constraints

Once the semantics has been decomposed to a set of graphical constraints, the system then processes the scene further to add implicit constraints. These are graphical conststraints that are not specified in the input text but which are necessary to produce a coherent scene. For example, the input text normally will rarely specify every object's position along all spatial axes, so the system must decide where to put the object in those unspecified axes. This can be in a *support axis* or a *secondary axis*.

6.5.1 Assigning Supports

The system starts by ensuring that all objects have some means of support. It does this by examining all the existing graphical constraints. Thus, if the GROUND (chair) object is rotated, the FIGURE (dog) will keep its default orientation. See Section 3.8.1.

Note that a sentence that specifies that *the dog is a foot above the chair* counts as a *weak support* constraint. In other words, even though the object is not physically supported, it is in fact constrained to be at the vertically specified position which is enough to exempt the FIGURE object from needing another support. But since it is a weak support, the GROUND object is not used to establish a spatial reference frame.

If an object is not supported, then a support must be assigned. The system has a number of heuristic rules to assign supports. If an object is laterally related to another object and is otherwise unsupported, it will be made vertically "level with" that other object. This counts as a weak support.

Since all objects (other than the ground) must have support, this means that ultimately we must construct a tree of supports rooted by the global ground. If a group of objects is mutually supported but not attached to the ground, the root of that tree will be put on the ground. This occurs recursively until all objects are supported either directly or indirectly by the global ground.

6.5.2 Centering on Secondary Axes

Normally textual constraints are underspecified. When we say *The mouse is left of the piano*, we have not specified the support for the mouse. As mentioned above, we will make it level with the piano unless some other sentence specifies a support. Also we have not specified how far forward or backward the mouse is with respect to the piano. It could be anywhere, even in front of the piano. In cases like this, we infer a new spatial relation to center the figure object along that *secondary axis*, in this case the Z (front-back) axis. If we had said *the mouse is in front of the piano*, the secondary axis would be the X (left-right) axis.

The same principle applies when all that is known is an object's support. For example, with *the dog is on the table* we were not told how far left or right or how far front and back on the table to place the dog. As a result we center the dog on the table in both the X axis and the Z axis. The same principle also applies for objects that are on vertical surfaces like walls. The only difference is that the Y axis (up-down) becomes a secondary axis and the Z axis becomes the support axis. Wall-items also can also have a DISTANCE-ABOVE-GROUND property that declares how far above the ground they should be by default.

This process becomes more complicated when there are collections of objects spatially related to another object. In those cases, the collection as a whole is centered. This often involves recursively aligning both the collection bounding box with respect to the support and aligning the objects in the collection with each other.

6.5.3 Avoiding Colocation

If we say *the cats are on the table*, it would be unfortunate (for the cats) if we put them in exactly the same location on the table. To avoid this cat-astrophe, we must keep the cats separate. This is done by logic that centers objects on secondary axes and determines if those objects are already constrained to one another (even if indirectly through other objects). If not, then a new inferred relation puts one next to the other. This logic-based inference procedure is not optimal since objects tend to be aligned in rows and columns and do not look totally natural. A future area of investigation would to apply statistical models to arrange

objects according to patterns found in the real world along with physics-based collision avoidance and physical psuedo-forces to ensure objects are separate. In some cases, a physics simulation could determine the spatial arrangements, such as when objects such as leaves on the ground are arranged by natural physical forces or the way that a stick is leaning against a tree.

We note, though, that users love to have objects interpenetrating. So techniques and imposed constraints like these have to be defeasible.

6.6 Building the Scene

Once the final set of graphical constraints is in place, the system must build the scene. This involves geometric aspects (scene layout, mostly involving positioning objects) as well as material selection.

6.6.1 Spatial Layout Constraints

Scene layout works as follows: A set of primitive graphical constraints specifies relative positions, orientations, and sizes. Each constraint includes a test to see if the constraint holds and a method to enforce the constraint. Spatial constraints take a figure and ground object as arguments. Each can have a relevant region or part designated that is the target of the constraint. These are normally provided by the object's affordances but can also be regions on the object specified by the input language.

In addition to explicit (viewable) 3D objects, the scene contains grouping constructs that play a role in the layout. For example if we say *the bird is above the trees*, the bird will be positioned above the center of the group of trees. The group itself is a bounding box that conforms to the individual trees within it. Those trees can be arranged in any manner and the bounding box will adjust to those positions. The bird will then be placed relative to that bounding box. The layout, thus, can take a few iterations to stabilize (assuming it is well formed). In cases where the layout does not stabilize, the user is alerted that the layout could not converge, in which case the user can adjust their scene by adding or removing constraints (via language of course). In theory, the system could give the user alternatives as to what constraints are in conflict.

6.6.2 Surface Property Constraints

Surface properties can be materials (such as wood or glass) or individual components (textures, decalsm colors, reflectivity, and transparency). Materials are sets of individual components. Any component or

material can be modified by other constraints. So, for example, a color can be made bright or desaturated, and the size of a texture can be modified, and a wood material can eg made shiny (*the wood texture on the table is blue*).

The system resolves the surface property constraints for each object by recursively applying modifiers, starting with those that are already resolved. Special logic takes care of the case of layered materials, such as a decal on top of a textured object.

6.7 Rendering the Scene

Once all stand-in objects have been moved into place, the system writes out a scene description file for the renderer. This specifies all objects, lights, materials, and textures in in the scene. For each object it specifies the original polygonal model rather than the stand-in. It also gives the 3D transform on every object in order to scale, position, and orient the object as desired. Texture maps can have *uv* scales and offsets. The renderer handles transparency, reflectivity, and glass materials and multiple light sources. The rendering (raytracing) for the online system is done on state-of-the-art GPUs by NVIDIA. An average scene takes a couple seconds to render.

6.8 User Interfaces and APIs

When the scene is rendered, a few hundred thumbnail-sized images from different points of view are rendered at the same time. These are sent to the browser and allow the user to interactively pan through the thumbnails, as a sort of flipbook, in order to select a new view to recompute the full-sized scene. Also, a buffer image is rendered that records the 3D XYZ coordinate of each pixel. This allows the user to click anywhere on the scene and have that point become a new aimpoint.

6.8.1 Scene Coherence

It is important to keep assigned objects stable while the user is working on a scene. If the user starts by typing *The book is on the table* and then displays their scene, the system will choose a particular book and a particular table for them. The user can specify the type of table more specifically (e.g. *the coffee table*), but even then, there may be more than one possible matching object. The user can also at any point choose from among the possible matching objects for every image and 3D object object in their scene from an object

browser. In either case, the system will store the mapping from words to objects and also remember object choices inferred by the system (e.g. via vignettes). So when the user works on their scene, the set of objects already specified will remain stable.

The same principle applies to the camera. When the user adjusts the camera position, the system remembers it. So when the scene is changed, the camera remains in the same place. This is tricker than it may seem since adding new objects changes the size and layout of the scene. The system keeps things stable by remembering the camera position and its aimpoint relative to objects in the scene. It can then maintain the relative position of the camera to the object closest to its aimpoint.

Other problems can occur with the camera. When a scene is created, a default camera position is computed to frame the scene. However, this does not always give the best possible viewpoint, in particular in cases when the objects vary in size, or are far apart, or where one occludes the other. See Section 8.4 and Section 8.6.1 for more on this issue in one of our evaluations of the system.

6.9 Action Vignette Example

In this example, we show successive stages of the meaning representation:
text → *phrase-structure* → *syntactic dependency* → *semantics* → *graphical primitives* → *rendering*.

Text: *The truck chased the man down the road.*

Syntax

```
(SS
 (S (NP (DT "the") (NN "truck"))
  (VP (VBD "chased") (NP (DT "the") (NN "man"))
   (PREPP (IN2 (IN "down")) (NP (DT "the") (NN "road")))))
 (ENDPUNC "."))
```

Syntactic Dependency

```
((#<lex-3: "chase">
  (:SUBJECT #<lex-2: ”truck">) (:DIRECT-OBJECT #<lex-5: ”man">)
  (:DEP #<lex-6: "down">))
 (#<lex-6: "down">
  (:DEP #<lex-8: "road">)))
```

Semantics

```
((#<lex-3: cotheme.chase.r>
  (:THEME #<lex-2: ”truck">) (:COTHEME #<lex-5: ”man">) (:PATH #<lex-6: "down">))
 (#<lex-6: "down">
  (:OBJECT #<lex-8: "road">)))
```

Vignette

```
((#<lex-3: cotheme-on-path.vr>
  (:THEME #<lex-2: ”truck">) (:COTHEME #<lex-5: ”man">) (:PATH #<lex-6: "down">)
  (:POSE "running"))
 (#<lex-6: "down">
  (:OBJECT #<lex-8: "road">)))
```

Graphical Primitives

```
((#<relation: gfx.in-pose.r>  (:OBJECT #<lex-5: "man">) (:POSE "running"))
 (#<relation: gfx.orientation-with.r>
    (:FIGURE #<lex-2: ”truck">) (:GROUND #<lex-8: "road">))
 (#<relation: gfx.behind.r>
    (:FIGURE #<lex-2: ”truck">) (:GROUND #<lex-5: ”man">) (:REF-FRAME #<lex-8: "road">))
 (#<relation: gfx.on-top-surface>
    (:FIGURE #<lex-5: "man">) (:GROUND #<lex-8: "road">))
 (#<relation: gfx.on-top-surface>
    (:FIGURE #<lex-2: ”truck">) (:GROUND #<lex-8: "road">)))
```

Rendered Scene

The preceding translation sequence utilizes the COTHEME-ON-PATH.VR vignette (shown below). The input of ROAD.N matches type PATH.N for the vignette, and hence the COTHEME.CHASE.R semantic to COTHEME-ON-PATH.VR vignette translation is made. The semantic translation rule also provides the default pose of "running".

Vignette declaration and translation rules

```
// Vignette definition:
(defrelation we.cotheme-on-path.vr (fn.cotheme.chase.r)
             :core-args ((:theme t) (:cotheme t) (:path path.n) (:pose t)))

// Rule to translate from semantics to vignette:
(:match ((?rel :agent ?agent :cotheme ?cotheme :path ?path-locative)
             (?path-locative :ground ?path))
     :types ((?rel :fn.cotheme.chase.r)
             (?agent :animate-being.n)
             (?path :f-ground-path.n)
             (?path-locative (or :we.on.r :we.up.r :we.on.r :we.across.r :we.along.r)))
     :output ((cotheme-on-path.vr :agent ?agent :cotheme ?cotheme
                                  :pose "running" :path ?path-locative)))
}

// Rule to translate from vignette to primitives:
(:match ((?rel :agent ?agent :cotheme ?cotheme :path ?path-locative :pose ?pose)
         (?path-locative :object-of-prep ?path-ob))
     :types ((?rel cotheme-on-path.vr))
     :output ((gfx.in-pose.r :subject ?cotheme :value ?pose)
      (gfx.in-pose.r :subject ?agent :value ?pose)
      (gfx.orientation-with.r :figure ?agent :ground ?path-ob)
      (gfx.behind.r :figure ?agent :ground ?cotheme)
      (gfx.on-top-surface.r :figure ?agent :ground ?path-ob)
      (gfx.on-top-surface.r :figure ?cotheme :ground ?path-ob)
      ))
```

6.10 Conclusion

WordsEye's semantic processing leverages the uniform representational scheme used by VigNet: in the ontology, input sentence representation, the database assertions, and the rules patterns themselves. Several areas of future improvements have been identified: a) merging of parts with semantically plausible wholes, b) parallelized semantic interpretation to better handle ambiguity, c) better handling of referring expressions, and d) continued investigation of supplying semantic defaults.

Chapter 7

Evaluation in Education

7.1 Introduction

Text-to-scene conversion has potential application in several areas. WordsEye has undergone extensive testing as an online graphical authoring tool and been evaluated in the classroom as a way to build literacy skills. Seeing sentences spring to life makes using language fun and memorable. This suggests many uses in education including ESL, EFL, special needs learning, vocabulary building, and creative storytelling. We describe, below, our testing of WordsEye in educational settings, including a formal experiment to help improve literacy with 6th grade students in a summer enrichment program in Harlem. The work on this evaluation was a joint effort with educator Michael Bitz, who created the pre- and post- test evaluation materials and with Cecilia Schudel who created new WordsEye content and organized the weekly sessions with the students. Cecilia also interfaced WordsEye to the FaceGen 3D software package [FaceGen,] and prepared the necessary graphics to let the students experiment by putting themselves and others into scenes. This work done for this project was supported by the National Science Foundation under Grant No. IIS-0904361.

The connections between visual perception and language acquisition have been a major focus of researchers in the fields of cognition, psychology, and neuroscience. Barbara Landau [Landau *et al.*, 1981], for example, presented subjects with a block on a box and told them "the block is acorp the box." Subjects interpreted *acorp* to mean *on*. In contrast, when shown a stick on a box, subjects interpreted *acorp* as *across*. These results, and those of many other similar experiments, highlight well-documented theories of cognitive development by Piaget [Piaget, 1999]: people form mental schema, or models, to assimilate the knowledge

they already have and accommodate new information presented to them. Gardner [Gardner, 1985] identified the importance of visual-spatial "intelligence," furthering the body of research linking visual perception to other important human activities, including language acquisition. Despite this research and base of knowledge, schools in the United States have placed a much larger emphasis on language acquisition than visual perception. Perceptual psychologist and art theorist Rudolf Arnheim [Arnheim, 1969] demonstrated that the dismissal of perception, and subsequently of the visual, has resulted in widespread visual illiteracy.

Lawrence Sipe [Sipe, 2008], a researcher of children's literature, thoroughly examined the interplay between words and pictures in a picture book. Sipe described a text-picture synergy that produces an effect greater than that which text or pictures would create on their own. He cited the work of numerous scholars who attempted to depict the relationships between text and imagery in a picture book, ranging from the idea that pictures extend the text [Schwarcz and Association, 1982] to the concept that pictures and text limit each other [Nodelman, 1990]. Sipe concluded, "As readers/viewers, we are always interpreting the words in terms of the pictures and the pictures in terms of the words...The best and most fruitful readings of picture books are never straightforwardly linear, but rather involve a lot of reading, turning to previous pages, reviewing, slowing down, and reinterpreting".

Sipe's connections between visual imagery and written text in children's books primarily relate to reading skills. Research in the realm of comic books in education further this connection to writing skills. Frey and Fisher [Walsh, 2007] present series of chapters on to this subject, highlighting the possibilities for writing development through visual imagery. Bitz [Bitz, 2010] demonstrated the power of student-generated comic books for writing development through a national program called the Comic Book Project. Similar outcomes for both reading and writing have been demonstrated through other visual media, including videogames [Gee, 2003]; film [Halverson, 2010]; and website development [Walsh, 2007].

7.2 Albemarle County Schools Pilot

We performed some preliminary testing of WordsEye in schools in Spring 2007. After seeing a demo of WordsEye at the Innovate 2007 Exposition (hosted by the State of Virginia Department of Education), K-12 public school teachers from the Albemarle county school system in Virginia asked if they could use it in their classes, believing it to be a useful tool for ESL (English as a second language) remediation, special education, vocabulary enhancement, writing at all levels, technology integration, and art. Feedback from

these teachers and their students was quite positive. In one school, with a 10% ESL population, a teacher used it with 5th and 6th graders to "reinforce being specific in details of descriptive writing" and noted that students are "very eager to use the program and came up with some great pictures." Another teacher tested it with 6th through 8th grade students who were "in a special language class because of their limited reading and writing ability," most reading and writing on a 2nd or 3rd grade level. In addition to its educational value, students found the software fun to use, an important motivating factor. As one teacher reported, "One kid who never likes anything we do had a great time yesterday...was laughing out loud."

7.3 HEAF Study

To further test the hypothesis that WordsEye could serve as an effective alternative literacy tool, we designed a controlled experiment for middle school children enrolled in a summer enrichment program run by the Harlem Educational Activities Fund (HEAF) [Coyne *et al.*, 2011b]. In the trial, twenty seven emerging 6th grade students in a HEAF literature course were given a writing pre-test (Figure 7.3). 41% of the students were female and 59% were male; 89% were African American, 4% were Native American, and 7% were other ethnicities. Half of the students (control) were randomly chosen to participate in a conventional literature course (i.e., group discussions and thematic analysis); the other half (treatment) were introduced to WordsEye and shown how to use it to construct pictures from text. The control curriculum consisted of a variety of activities ranging from book discussion groups to the development of an original puppet show. Over the next 5 weeks, the WordsEye group used WordsEye for 90 minutes a week to create scenes from the literature they read, including Aesop's fables and Animal Farm.

To protect children's privacy we have implemented classroom registration and private gallerys so that the students can post their scenes to a gallery only visible to others in their classroom. Likewise, they can only see scenes in their gallery and not those on the public website.

We developed a curriculum (Appendix B) that helped instructors integrate WordsEye into the learning goals of the summer academy. For the rest of the course, the WordsEye group participated in the same type of class discussions and writing exercises as the control group. At the end of the course, all of the students wrote essays based on the literature they had read; pre- and post-course essays were scored by independent, trained raters. The criteria were: a) Organization and Structure b) Written Expression c) Quality and Depth of Reflection d) Use of Vocabulary e) Mechanics: Grammar, Punctuation, Spelling. Each category was

Pre-test

- Describe a character from your favorite book, including physical traits, personality traits, and anything else that is important to understanding the character in depth.

- Retell a scene from your favorite book, including background information, sequence of events, and important details.

Post-test

- Describe a character from one of Aesop's fables, including physical traits, personality traits, and anything else that is important to understanding the character in depth.

- Retell a scene from Animal Farm, including background information, sequence of events, and important details.

Figure 7.1: Pre-test and post-test given to students in both WordsEye and control groups

Figure 7.2: HEAF sessions

Figure 7.3: HEAF pictures from Aesop's Fables and Animal Farm

judged on a scale of 1 (poor) to 5 (excellent).

The average pre-test treatment score was 15.8; the average pre-test control score was 18.0. We determined that this was as close to a baseline measurement as possible, given the parameters of the summer program. The raters met to review the evaluation rubric and then independently judge two writing samples. The scores were discussed among the group, and inter-rater reliability was determined sufficient at 92%. Students using WordsEye improved significantly over the students who had taken the more traditional version of the course (Table 7.1).

	Pre-test	**Post-test**	**Growth**
WordsEye Group	15.82	23.17	7.37
Control Group	18.05	20.59	2.54

Table 7.1: WordsEye group had significantly more improvement than control group ($p < .05$).

Note that, as this study was a straightforward comparison between the growth scores of two indepen-

dent and randomly assigned groups with a small sample size, the researchers used a two-sample t-Test to determine statistical significance: Difference = mu (1) - mu (2); Estimate for difference: 4.81; 95% CI for difference: (0.08, 9.54); t-Test of difference = 0 (vs not =): t-Value = 2.16 p-Value = 0.047 DF = 16.

7.4 Conclusion

We have shown that students who used WordsEye had significant improvement in literacy skills versus a control group. As one of the students said "When you read a book, you don't get any pictures. WordsEye helps you create your own pictures, so you can picture in your mind what happens in the story." The WordsEye group was also introduced to WordsEye's face and emotion manipulation capabilities – the children loved including themselves and other people in scenes and modifying the facial expressions.

Since concluding this study, we have approached several schools in the New York area to participate in further pilot studies. We have also had several classrooms from other places around the world using the system. We are also in contact with a group in China interested in using WordsEye in English language schools.

There is great enthusiasm by educators for the system. One teacher from the United Kingdom said "It was really engaging. (The kids) loved it. Got lots done. Did some work on adjectives and prepositions. They all wanted to do more...it was really really good. I hope you keep developing and producing the planned app and educational functionality." Another from New York said "As a literacy tool, WordsEye is amazing for reinforcing the importance of descriptive and figurative language... I imagine having students build a lexicon of language that works in WordsEye."

Chapter 8

Evaluation with Realistic vs Imaginative Sentences

8.1 Introduction

In Chapter 7 we described how the WordsEye text-to-scene system has been evaluated as a tool for education to help students improve their literacy skills. Here, we focus on evaluating the pictures produced by the WordsEye system more directly as accurate illustrations of input sentences. The Amazon Mechanical Turk evaluation described in this chapter was done in collaboration with Morgan Ulinski.

WordsEye addresses the problem of creating vs. automatically finding a picture to illustrate a sentence. Standard image search engines are limited to pictures that already exist in their databases, biasing them toward retrieving images of mundane and real-world scenarios. In contrast, a scene generation system like WordsEye can illustrate a much wider range of images, allowing WordsEye users to visualize unusual and fantastical things.

In this Chapter we evaluate the output of WordsEye vs. simple search methods to find a picture to illustrate a sentence. To do this, we construct two sets of test sentences: a set of crowdsourced *imaginative sentences* and a set of *realistic sentences* extracted from the PASCAL image caption corpus [Rashtchian *et al.*, 2010]. For each sentence, we compared sample pictures found using Google Image Search or produced by WordsEye and then crowdsourced judgments as to which picture best illustrated the sentence. We also crowdsourced numeric ratings as to how well each of the images illustrated its sentence.

In Section 8.2 we discuss related work. In Section 8.3 we describe the construction of *imaginative* and

Figure 8.1: Imaginative images: situational, iconic, abstract, fantastic

realistic sentences. In Section 8.4 we explain the collection of potential illustrations for these, using Google Image Search or WordsEye. In Section 8.5, we discuss the use of crowdsourcing to evaluate the illustrations. We discuss the results of the evaluation in Section 8.6. We conclude in Section 8.7.

8.2 Related Work

Others have used crowdsourcing to collect human-generated sentences, e.g. to create image captions. This includes the PASCAL image caption corpus [Rashtchian *et al.*, 2010], Flickr8k [Hodosh *et al.*, 2013] and Microsoft COCO [Chen *et al.*, 2015]. Our work differs from these in that we want to collect sentences describing anything Turkers can imagine, as opposed to descriptions of existing photographs. [Zitnick and Parikh, 2013] crowdsourced the evaluation of their scene generation system using 2D clip art: subjects created an initial set of scenes and wrote descriptions of the scenes. [Zitnick *et al.*, 2013] used several

methods to automatically generate scenes for these descriptions and asked subjects which picture matched the description better. While the pictures that the sentences describe are human-constructed scenes rather than photographs from sources like Flickr, the scenes use a fixed set of 80 objects and are limited to the domain of children playing outside. [Chang *et al.*, 2015] evaluate their text-to-scene system by asking people to rate the degree to which scenes match the text used to generate them. Their test corpus includes a much larger number of objects than [Zitnick *et al.*, 2013], but the sentences and scenes are realistic descriptions of configuration of objects in a room.

8.3 Elicitation and Selection of Sentences

In this section we describe using crowdsourcing to collect imaginative sentences and filtering the PASCAL image caption corpus [Rashtchian *et al.*, 2010] to obtain realistic sentences.

8.3.1 Imaginative Sentences

We used Amazon Mechanical Turk to obtain imaginative sentences for our evaluation. We gave Turkers short lists of words divided into several categories and asked them to write a short sentence using at least one word from each category. The words provided to the Turkers represent the objects, properties, and relations supported by the text-to-scene system.

To help Turkers construct sentences of different types, we organized the objects, properties, and relations into a few basic categories. including: animals, locations, props (small, holdable objects), fixtures (large objects such as furniture, vehicles, and plants), spatial terms, and colors. We restricted the lexicon to include only commonly known words that could be easily understood and recognized visually. We excluded super-types such as *invertebrate* and sub-types such as *european elk*. We also omitted more obscure terms such as *octahedron* or *diadem*. The resulting lexicon included about 1500 terms and phrases, split into 11 internal categories. Examples of words in each category are shown in Table 8.1. The categories presented to the turkers included combinations of the internal categories.

We created 12 different combinations of categories with 20 HITs per combination. Each HIT randomly presented different words for each category in order to elicit different types of sentences from the Turkers. This involved varying the types and number of categories as well as the order of the items in the categories. We wanted to encourage sentences such as “there is a blue dog on the large table" as well as different orders

Category	**Definition**	**Examples**
prop	Small objects that could be held or carried	*cellphone, apple, diamond*
fixture	large objects such as furniture, vehicles, plants	*couch, sailing ship, oak tree*
animal	Animals	*dolphin, chicken, llama*
spatial term	terms representing spatial relations	*above, against, facing, on*
number	small numbers	*one, four, nine, twelve*
color	common colors	*beige, green, scarlet, black*
size	general size or specific dimensions	*big, tiny, thin, 5 feet long*
distance	distances	*4 inches, five meters, 10 feet*
surface property	surface properties	*opaque, shiny, transparent*
location	terms representing terrain types and locations	*field, driveway, lake, forest*
building	buildings and architectural structures	*doghouse, castle, skyscraper*

Table 8.1: Categories of words in the lexicon

and constructs like “the dog on the large table is blue". Each HIT showed 4 or 5 categories, with three words per category. Table 8.2 shows all the combinations of categories that we used.

Our instructions specified that Turkers should write one sentence using a maximum of 12 words. Words could be in any order as long as the resulting sentence was grammatical. We allowed the use of any form of a given word; for example, using a plural noun instead of a singular.

We also allowed the use of *filler words* not listed in the categories, but asked them not to add any unlisted *content words*. We defined *filler words* as words with “little meaning on their own, but are used to make the sentence grammatical (e.g. *the*, *has*, *is*, *with*)" and *content words* as words that “refer to an object, action, or characteristic (e.g. *eat*, *shallow*, *organization*)." An example HIT is shown in Figure 8.2.

We restricted our task to workers who had completed at least 100 HITs previously with an approval rate of at least 98%. We paid $.04 per assignment. We started with 240 unique combinations of words and collected one sentence for each of these. After filtering out ungrammatical sentences, we ended up with a total of 209 imaginative sentences, all of which generated scenes. Some examples are shown in Figure 8.3(a).

<table>
<tr><th>AMT Column Headings</th><th colspan="2">WordsEye Lexical Categories</th></tr>
<tr><td>1. Noun1, Noun2, Spatial Term, Adjective
2. Adjective, Noun1, Noun2, Spatial Term</td><td>Noun1 is prop. Noun2 is fixture.</td><td rowspan="5">Spatial Term is spatial term. Adjective is size, color or surface property. Distance is distance. Location is building or location. Size is size. Color is color. Number is number.</td></tr>
<tr><td>3. Adjective, Noun1, Noun2, Spatial Term</td><td>Noun1 is animal. Noun2 is fixture.</td></tr>
<tr><td>4. Noun1, Noun2, Spatial Term, Distance, Adjective</td><td>Noun1, Noun2 are prop, fixture or animal.</td></tr>
<tr><td>5. Noun1, Noun2, Spatial Term, Location</td><td>Noun1 is animal. Noun2 is prop or fixture.</td></tr>
<tr><td>6. Noun1, Noun2, Spatial Term, Distance, Adjective
7. Adjective, Noun1, Noun2, Spatial Term, Distance
8. Noun1, Noun2, Spatial Term, Location
9. Noun1, Noun2, Spatial Term, Color, Size
10. Noun1, Noun2, Spatial term, Number
11. Noun1, Noun2, Spatial term, Number, Adjective
12. Adjective, Noun1, Noun2, Spatial Term, Number</td><td>Noun1, Noun2 are prop or fixture.</td></tr>
</table>

Table 8.2: Possible combinations of categories for the sentence construction task

8.3.2 Realistic Sentences

We began with image captions collected by [Rashtchian *et al.*, 2010] for the PASCAL dataset [Everingham *et al.*, 2011], which consists of 5000 descriptive sentences, 5 captions each for 1000 images. The images cover 20 object categories from the original PASCAL task, including *people, animals, vehicles*, and *indoor objects*. We used at most a single caption for each photograph.

To select a usable caption, we manually removed all ungrammatical sentences and fed the remaining sentences into WordsEye which was able to create scenes for about one third of the image captions; the captions that were rejected were omitted due to current limitations of the system's lexicon, object library, or parser. Since our goal was to evaluate WordsEye we excluded these sentences which are outside the domain of the system. For example, we omitted most sentences using action verbs since the system currently cannot pose characters to represent those verbs. We kept simple stative pose-related verbs such as *sit* and *stand* so the system could capture other aspects of the sentence. We also omitted sentences that could not be parsed or

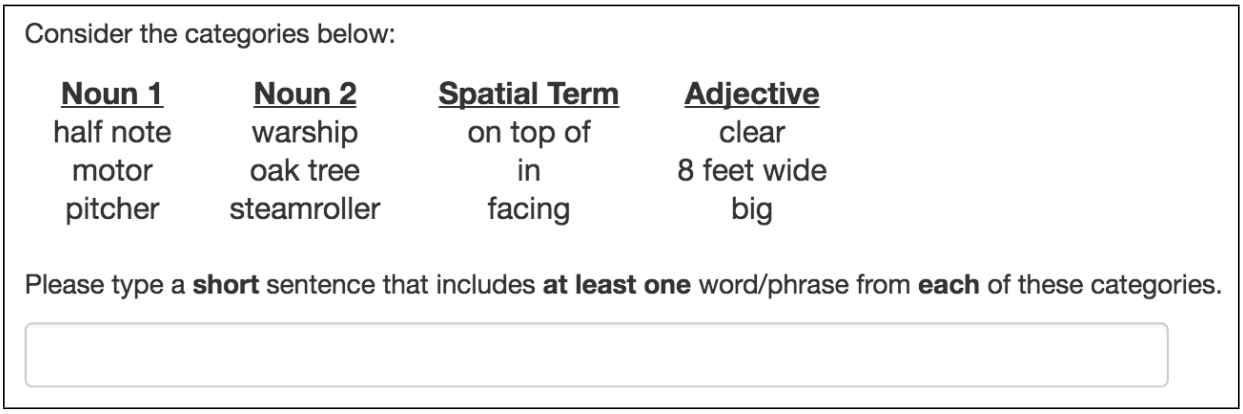


Figure 8.2: Example of sentence collection HIT

(a) Imaginative	**(b) Realistic**
• *The yellow checker cab is in front of a hydrant.*	• *A brown duck and white duck stand on the grass.*
• *Five pears are under the martini glass.*	• *A man is standing next to a yellow sports car.*
• *The large prawn is on top of the stool.*	• *A black dog in a grass field.*
• *The red clock is three feet above the desk.*	• *The big white boat is in the ocean.*
• *The spatula was close to eight eggs.*	• *A child sits in a large black leather chair.*

Table 8.3: Examples of imaginative and realistic sentences

that had concrete nouns with no corresponding 3D object. This resulted in a total of 250 realistic sentences. Some examples are shown in Table 8.3(b).

8.4 Collection/Generation of Illustrations

In this section, we describe how we obtained the possible illustrations for each sentence.

Google Image Search: We used each sentence as a query for Google Image Search. We did not strip punctuation, add quotation marks, or otherwise modify sentences. The first 4 results were downloaded and resized to a uniform width.

WordsEye Scene Generation:

Since WordsEye images are rendered 3D scenes, they can be easily viewed from different angles. Normally, users can interactively change the viewpoint in the scene they are creating and choose the best view. So our approach was to automatically generate four WordsEye scenes with slightly different camera views.

If one of the objects was occluded by another (and hence not visible in a front view of the scene), we would automatically produce an alternate view of the scene from the back. Likewise, the elevation of the camera was varied to allow an object to potentially be more visible. In addition to varying the camera, we also randomized the objects chosen in the scene among those compatible with the sentence. For example, Figure 8.3 shows four scenes for the sentence "*A furry dog lying on a glass table*."

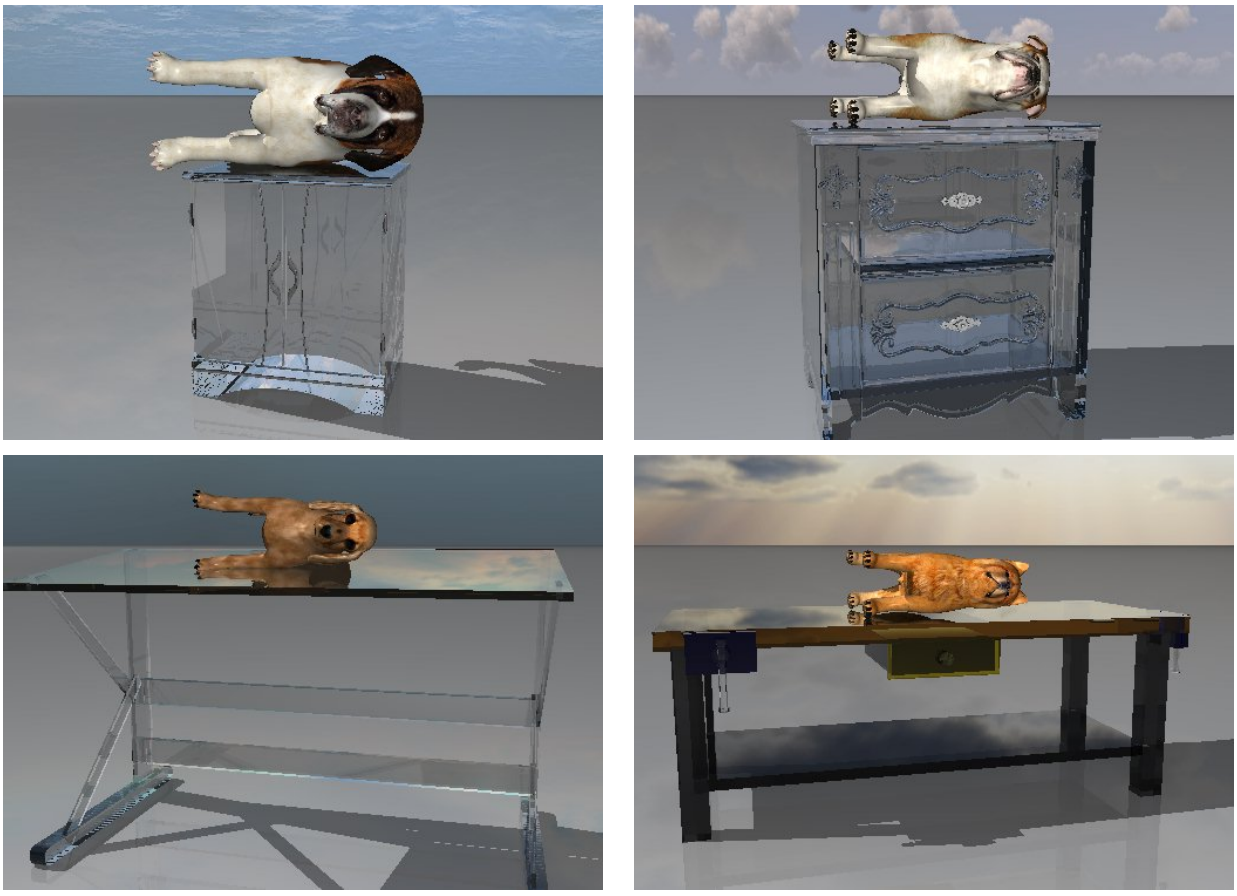


Figure 8.3: Generated scenes for the sentence "*A furry dog lying on a glass table.*"

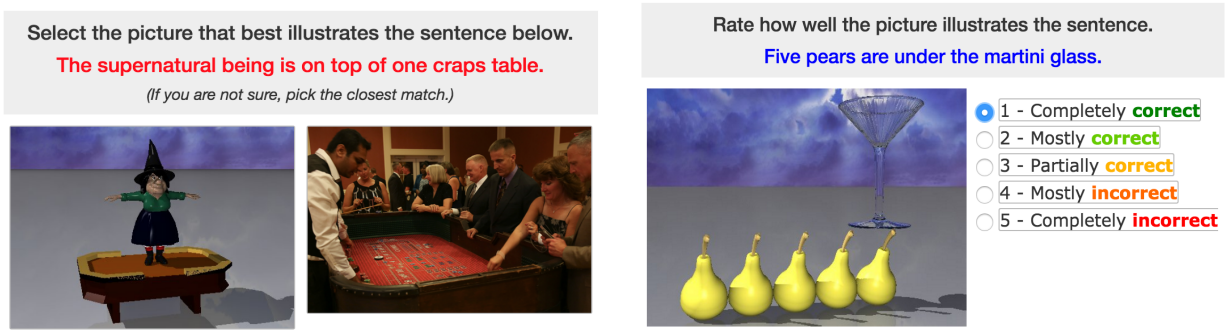
8.5 Evaluating Illustrations with AMT

The evaluation of the quality of the illustrations was done in two phases. In the first phase we determined the best image for each sentence from the downloaded Google results and for each sentence among the WordsEye-generated images. The second phase evaluated the quality of the best Google image and the best

WordsEye image. We did this second phase evaluation with two separate crowdsourced tasks. In the first, we compared the best Google image with the best WordsEye image directly. In the second, we obtained a rating for how well each of the images illustrated the sentence. For all tasks, we required turkers to have previously completed at least 500 HITs and to have a 98% approval rate. We paid $0.01 per assignment.

Each image comparison HIT showed a single sentence with the possible images below it. Turkers were asked to select the picture that best illustrated the sentence. In the first phase, we showed four pictures and collected 5 judgments for each HIT. In case of ties, we published additional assignments for that sentence until one image had more votes than any of the others. The image that received the most votes was used in the next phase, which compared the winning Google image with the winning WordsEye image. In the second phase, we collected 3 judgments for each HIT, which guaranteed no ties. A sample HIT from the second phase is shown in Figure 8.4(a).

For the rating task, each HIT showed a single sentence and a single image. Turkers were asked to rate how well the picture illustrated the sentence. The scale was from 1 (completely correct) to 5 (completely incorrect). We collected 3 judgments for each HIT and averaged these ratings to obtain the final rating for each picture. An example of the rating HIT is shown in Figure 8.4(b).



(a) (b)

Figure 8.4: Examples of the second phase AMT tasks: (a) image comparison task and (b) rating task. (Google image source: `https://en.wikipedia.org/wiki/Craps`)

8.6 Results and Discussion

In this section, we discuss results from the second phase of evaluation. In the image comparison task, we asked 3 Turkers to choose the picture that best illustrated the sentence. The distribution of outcomes is

Winner (votes)	**Imaginative**		**Realistic**	
WordsEye (3 to 0)	60.3% (126)	**85.6% (179)**	8.8% (22)	16.4% (41)
WordsEye (2 to 1)	25.4% (53)		7.6% (19)	
Google (2 to 1)	10.0% (21)	14.4% (30)	14.4% (36)	**83.6% (209)**
Google (3 to 0)	4.3% (9)		69.2% (173)	
Total	100.0% (209)		100.0% (250)	

Table 8.4: Distribution of Turkers' votes for WordsEye vs. Google images.

shown in Table 8.4. The winner is shown in bold for each category.

Next, we obtained a rating for each image. Average ratings for Google and WordsEye for each category of sentence are shown in Table 8.5(a). The better rating in each category is shown in bold. We also calculated the winning image for each category based on the ratings. For each sentence, the winning image was the one that had the lower rating. These are shown in Table 8.5(b). The winner is shown in bold for each category.

The trend for both votes and ratings is the same: WordsEye is superior for imaginative sentences and Google is superior for realistic sentences. The winning image based on votes is not always the same as the winning image based on rating. This can sometimes be the result of different Turkers doing the evaluations for judging versus ratings. It can also be the result of a single outlying numeric score swaying the average one way or the other. Table 8.6 compares the distribution winning images

For imaginative sentences, when the Google and WordsEye ratings were tied, WordsEye tended to win the votes. Even when Google had a better rating than WordsEye, WordsEye still tended to win by votes. In particular, out of the 42 cases where the Google image received a better rating, Turkers chose the WordsEye image for 24 (more than half) of them. This pattern is reversed for the realistic sentences. For realistic sentences, when both images had the same rating, Turkers tended to choose the Google image. However, when WordsEye had the better rating for a realistic sentence, Turkers still tended to choose the WordsEye image. Thus, while Turkers seemed to prefer to associate imaginative sentences with WordsEye-style pictures when forced to make a binary choice (even when the Google image had a lower rating), the reverse bias does not hold for realistic sentences: when a WordsEye image illustrated a realistic sentence better based on rating, the binary choices made by Turkers usually favored the WordsEye image as well.

	Imaginative	**Realistic**
WordsEye	**2.581**	2.536
Google	3.825	**1.871**

(a)

Winner	**Imaginative**	**Realistic**
WordsEye	**74.6% (156)**	25.6% (64)
Tie	5.3% (11)	13.6% (34)
Google	20.1% (42)	**60.8% (152)**
Total	100.0% (209)	100.0% (250)

(b)

Table 8.5: (a) Average ratings for WordsEye and Google images, where 1 is best (completely correct) and 5 is worst (completely incorrect). (b) Distribution of winner, based on ratings.

8.6.1 Error Analysis

The results show that Google is superior for the realistic sentences and WordsEye for the imaginative sentences. In some cases, both Google and WordsEye produced the same basic result for a sentence. For example, the realistic phrase *Two horses* produced unanimous ratings of "completely correct" for both systems, but the Google picture was judged better in the one-to-one comparison. The preference for Google, in this case, is likely a result of it showing realistic looking horses in natural poses and interaction with each other in a expected setting, while the WordsEye picture showed two less animated looking horses on a gray groundplane. We could refine our evaluation to identify these cases by having Turkers provide more than a simple preference. For example, there are other related criteria such as how natural the picture is and how much extraneous detail there is. A natural looking scene will often include extraneous detail, so there is a tension between those. Another area for future testing would be to include multi-sentence descriptions, which Google may have more trouble with as scenes become more complex and involve more objects, relations, and properties.

WordsEye had difficulties in the following conditions:

- **Graphical primitives.** One source of errors was from missing graphical primitives. For example, sentences that required a person or animal to be in a particular pose (e.g. sitting) are internally repre-

	WordsEye won rating	Tie rating	Google won rating	Total
WordsEye won votes	70.3% (147)	3.8% (8)	11.5% (24)	85.6% (179)
Google won votes	4.3% (9)	1.4% (3)	8.6% (18)	14.4% (30)
Total	74.6% (156)	5.3% (11)	20.1% (42)	100.0% (209)

(a) Imaginative sentences

	WordsEye won rating	Tie rating	Google won rating	Total
WordsEye won votes	14.0% (35)	0.8% (2)	1.6% (4)	16.4% (41)
Google won votes	11.6% (29)	12.8% (32)	59.2% (148)	83.6% (209)
Total	11.6% (64)	13.6% (34)	60.8% (152)	100.0% (250)

(b) Realistic sentences

Table 8.6: Distribution of winner (WordsEye vs. Google) based on ratings and votes.

sented, but the system is currently unable to actually put the 3D character into a pose.

- Other problematic cases related to limitations with graphical primitives (not shown) are with **clothing** (*the man with glasses*) and **multi-colored objects** (*the green and gray bird*).

- Another source of errors was in **anaphora resolution** in text like "*A field with many black cows in it.*" WordsEye currently processes anaphora and other coreference across sentences but not within a sentence.

- **Handling interiors**. Most 3D objects are constructed to be viewed from the outside and do not have well-defined regions in the interiors. In addition, they do not have large enough openings to be able to seen the inside from the outside. So, "*The scarlet stool is in the battleship*" will interpret *in* as meaning *embedded in* since there is no marked interior region.

- **Camera position**. There are a number of problems with our simple algorithm for automatically generating camera positions. This was especially an issue for imaginative sentences which sometimes involved very small objects in the same scene with large ones, making it hard to see both at the same time, given our default algorithm for positioning the camera to frame the full scene. A better strategy would be to position the camera aimed to frame the small object with the large object in the background.

For example "*The mauve flatbed truck is far from the three mugs*" correctly constructs the scene but chooses a view far from the much smaller mugs, making them almost impossible to see. It could provide alternatives that focus on the mugs with the truck in the background

In other cases, one object was inside another (e.g. within an enclosed area such as a building) and the default generated camera positions were outside the building, making it impossible to see the inner object.

Although techniques for automatically positioning the camera are outside the scope of this thesis, we note that work has been done by others for automatic cinemtography such as the Virtual Cinematographer project [He *et al.*, 1996] and the Camdroid system [Drucker and Zeltzer, 1995], and more general analysis of the mathematics of for positioning the camera to optimally fill a field of view [Blinn, 1988]. These and other techniques could potentially be applied to WordsEye to improve automatic camera placement to improve visability and presentational goals.

- **Semantic intepretation** Preposition and attribute interpretation did not handle various cases. For example, "*A grey house with a red door*" puts a door next to the house rather than as part of the house.

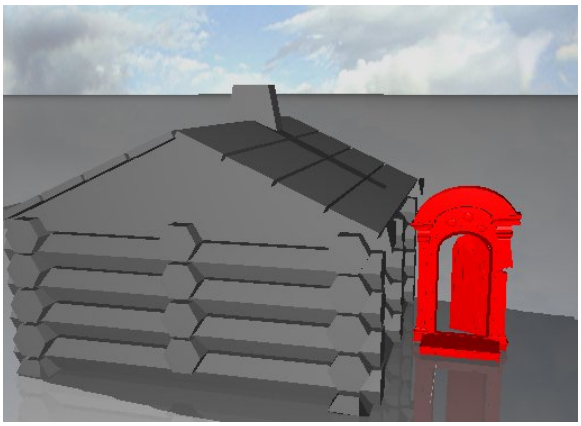
- Other errors occurred because of incorrect information stored in the **knowledge base** (e.g. incorrect real-world sizes resulting in strange relative sizes between objects in a scene) or from incorrect or unexpected semantic and graphical interpretations.

Examples of some of these errors are shown in Figure 8.5. To better understand the types of errors, we tagged each WordsEye image that had a rating worse than 2 (“mostly correct”) with the type of error that it exhibited and the WordsEye module where the error occurred. The WordsEye pictures for 114 imaginative sentences and 137 realistic sentences were tagged with errors. The distribution of the errors per module is shown in Table 8.7. Note that since some pictures were tagged with errors from multiple modules, the total of each column is greater than 100%. WordsEye made more knowledge base errors and camera errors on imaginative sentences. It made more semantic analysis, graphical analysis, and apply graphical constraints errors on realistic sentences.

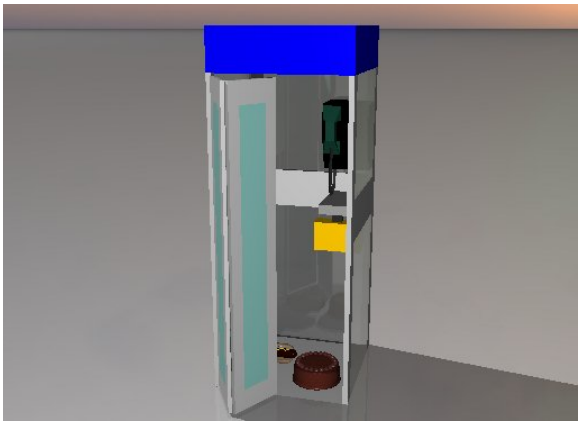
(a) *Five people sitting on one couch*

(b) *The scarlet stool is in the battleship*

(c) *The mauve flatbed truck is far from the 3 mugs*



(d) *a gray house with a red door*



(e) *The hotdog is next to the chocolate cake in the booth.*

(f) *Two men in a small wooden canoe on the water*

Figure 8.5: Example WordsEye errors: (a) Poses and object choice (fetus) (b) Handling interiors (c) Camera position and scale (d) Interpretation of prepositions (e) Camera viewpoint partial occlusion (f) Graphical interpretation and knowledge-base

WordsEye Module	Imaginative	Realistic
Knowledge Base	10.0% (21)	2.4% (6)
Graphics Library	3.3% (7)	6.0% (15)
Parsing	2.9% (6)	2.0% (5)
Reference resolution	0.5% (1)	2.0% (5)
Semantic analysis	3.8% (8)	15.2% (38)
Graphical analysis	10.0% (21)	25.6% (64)
Apply graphical constraints	4.3% (9)	21.2% (53)
Camera/Render	42.1% (88)	6.8% (17)
No Error	45.5% (95)	45.2% (113)

Table 8.7: Errors per module. Note: a given sentence could have more than one error. See Figure 8.6 for a system architecture diagram showing the relevant modules.

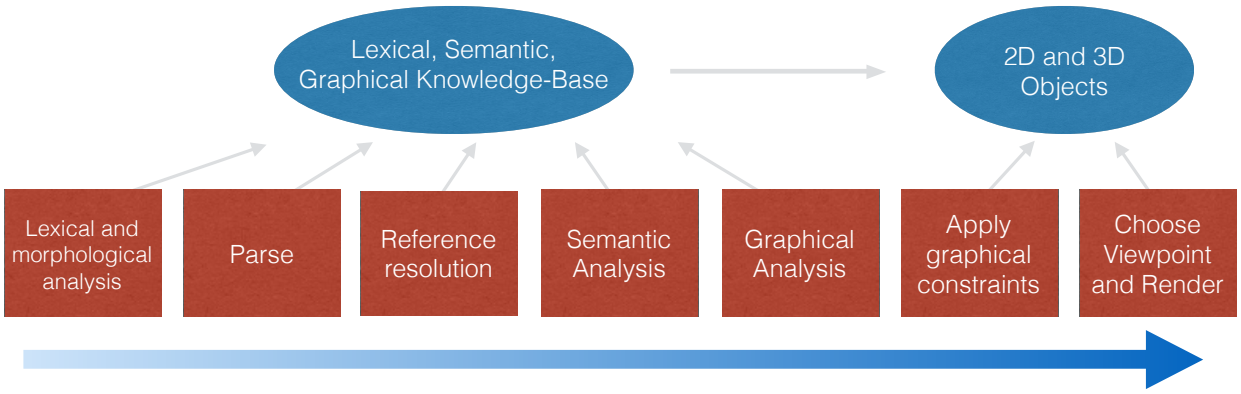


Figure 8.6: WordsEye system architecture. See Table 8.7 for a breakdown of which modules were responsible for different types of errors

8.6.2 Linking Error Analysis and Research Contribution

In order to partially quantify how the research contributions (Section 11.2) presented in this thesis help improve system accuracy, we continue the comparision with the AVDT text-to-scene system (Section 1.3.5) by estimating certain aspects of the performance of both systems on the realistic sentence corpus. Our focus here is only on the realistic sentences, since the imaginative sentences categories were created more specifically to test WordsEye's capabilities.

We seek to measure the benefit of WordsEye's broader coverage (and hence the benefit of its semantic theory) versus AVDTs performance. To do this we count the number of occurances of sentences of different common types in the realistic sentence corpus that are handled by WordsEye but would not be handled by the AVDT system. To do this, we consider sentences that involve surface properties, spatial affordances, size modifications, and non-default spatial preposition intepretation for "in" (e.g. in a bounded 2D area such as *in the field*), since these commonly appear in the realistic sentence corpus and are not handled by AVDT. We are unaware of any sentence types in the corpus that would be more accurately processed by AVDT than by WordsEye.

To collect this data, we examined all 250 realistic sentences and found that 154 (62%) depend on the following capabilities as follows:

- 106 sentences (42.4%) – surface properties (e.g. colors, color modifications, textures, opacity)
- 32 sentences (12.8%) – spatial affordances
- 30 sentences (12%) – size modification
- 41 sentences (16.4%) – ambiguous spatial preposition interpretation for "in"

None of the sentences in the categories under consideration would be handled properly by the AVDT system, due to the limits as outlined in Section 1.3.5. Of the 154 sentences that AVDT would fail on, WordsEye received a score of 2 ("mostly correct") or better by the Turkers on 76 (49%) of the sentences. Since sentences can rely on several capabilities, WordsEye might correctly depict a surface property but get a low score when the sentence also relied on a pose or when the camera position was poorly chosen. We also note the following:

- These sentences represent a subset of the capabilities of WordsEye – those that appear frequently in the sentences we tested – and do not fully represent the wider range of capabilities supported by WordsEye, either graphically (such as multiple constraints on the same object), linguistically (more

flexible coreference resolution), or semantically (various cases of preposition interpretation). Testing those would require a different evaluation corpus.

- There are also other sentence types (beyond the 154 sentences in the given categories) that AVDT would not be able to handle. We do not consider those, since they are fewer in number in this corpus (e.g. those related to an object's 3D orientation) or they would be areas not handled well by WordsEye either (e.g. with posed characters or other graphical limitations).

8.7 Discussion and Summary

We have described our evaluation of the WordsEye text-to-scene system; specifically, we have evaluated WordsEye's ability to create a picture that illustrates *imaginative* and *realistic* sentences, as compared to traditional image search methods (i.e. Google search). We found that WordsEye performs very similarly on both kinds of sentences (average rating of 2.581 and 2.536, respectively - on our rating scale, halfway between "mostly correct" and "partially correct"). While Google search does perform better than WordsEye on realistic sentences (average rating of 1.871 - between "completely correct" and "mostly correct"), performance breaks down when faced with imaginative sentences (average rating of 3.825 - between "partially correct" and "mostly incorrect"). Thus, we have shown that WordsEye is superior for imaginative sentences, and Google search is superior for realistic sentences. While this result is not unexpected, we can now quantify what the gap in performance actually is. In particular, while the average rating of WordsEye on realistic sentences was just 0.665 below that of Google, WordsEye's ratings on imaginative sentences was 1.244 higher than Google's. This suggests that as WordsEye and text-to-scene technology in general improve, they may become a viable alternative to image search even for realistic sentences, but it might be difficult to adapt traditional image search techniques to retrieve illustrations for imaginative sentences. In addition, as sentences get longer and more complicated (or if multiple sentences are involved), Google might be begin to have more trouble with realistic sentences as well.

The ability to generate both realistic and imaginative scenes from text input suggests that text-to-scene generation systems such as WordsEye can be used to supplement the results of image search engines such as Google. When users are freed from the normal constraints of what is possible or already exists they will often describe what they imagine – from situational, to iconic, to abstract, to fantastical. The majority of scenes created by actual users of the online WordsEye system are imaginative. One user commented "I

truly enjoy watching people unleash their minds here." Some examples of imaginative scenes that have been created in WordsEye are shown in Figure 8.1.

Creativity is something that too often gets overlooked in technology development, and our results show that research into text-to-scene generation could play an important role in addressing the issue. Our new corpus of imaginative sentences may also have applications for other researchers studying language in a visual context or those interested in spatial language in general.

Chapter 9

Evaluation as an Online Graphics Authoring Tool

9.1 Introduction

3D graphics authoring is a difficult process, requiring users to master a series of complex menus, dialog boxes, and often tedious direct manipulation techniques (see Figure 9.1). Natural language offers an interface that is intuitive and immediately accessible to anyone, without requiring any special skill or training. In this chapter we describe a) the market need for a text-to-scene system, b) the web and mobile user interface we have built around WordsEye, and c) what we have learned testing WordsEye in an online environment where real users have created and share rendered 3D scenes to an online gallery and to social media sites such as Facebook, Twitter, Tumblr and Reddit.

Most powerful 3D systems allow users to impose constraints. These can reduce the amount of detailed direct-manipulation. For example, if a book is put on a table and constrained to be there, then the table can be moved and the book will stay in place relative to the table. In traditional 3D authoring systems, constraints are placed by selecting from dialogs and menus. So while they are a powerful technique, they add to the complexity of learning and using a system. Language, by its very nature, imposes constraints – it is impossible to do direct manipulation with language since there is nothing to point and click at. Instead, language imposes constraints via the entities and relations it expresses. The key job of the text-to-scene system is to interpret what objects and constraints are intended.

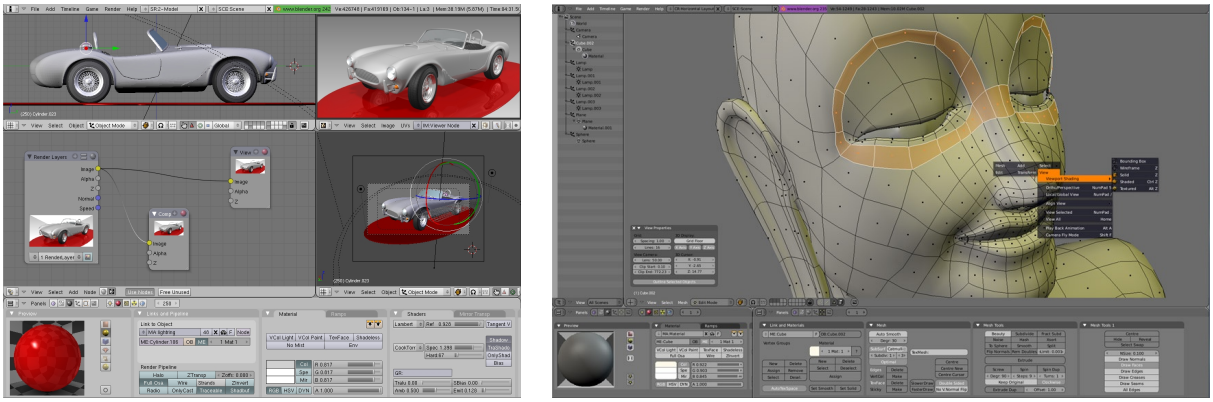


Figure 9.1: User Interface for the Blender 3D authoring system. [Wikimedia, a] and [Wikimedia, b]

9.2 Market Potential

In this section we describe the market potential and business case for using WordsEye as an online 3D graphics authoring tool for social media and other markets. Creating 3D scenes from text introduces a new technology and user interface paradigm for scene creation to the market. This same business case was presented to a panel of judges in the New York State Business Plan Competition in 2013, where WordsEye won the Grand Prize, from among 420 entries across the state.

Language-based 3D scene generation offers the following advantages over the traditional WIMP (Windows, Icons, Menus, Pointers) paradigm.

- Low entry barrier - no special skill or training required
- Give up detailed control, but gain speed and economy of expression
- Using language is fun and stimulates imagination. It enables novel applications in education, social media, gaming
- Amenable to hands-free and voice interfaces
- Image search as generation (and visualizating free-range text)
- All scenes come with their "DNA" visible (both lexical and semantic) – unlike scenes composed with typical GUIs where the procedure that created a scene is generally inaccessible and arcane. With WordsEye, the scene's input text is readily apparent and understandable by anyone. This enables easy entry point and modification.

9.2.1 Visual Social Media and Mobile Devices

The social media market is exploding with visual imagery and hungry for more and more visual content. Most online social media activity on sites like Facebook, Twitter, and Tumblr are centered around visual content. In 2013, facebook users uploaded 350 million photos every day. [Insider,] A Pew Interest survey [Rainie *et al.*, 2012] found that 45% of internet users post photos or videos online. And 41% "curate" photos and videos posted by others. We have observed these two classes of users (*creators* and *curators*) on the WordsEye website. The promise of text-to-scene technology like WordsEye is that it lowers the barrier so that a much larger group of people can become creators. And as more content is created, that opens up the door for more passive users to view and share the content. A typical rule of thumb for social media applications is that the ratio of creators to casual users to passive users is 1:9:90. [Wikipedia, 2016a]

Increasingly, social media activity takes place on mobile devices. The limited screen space and lack of a full keyboard presents some difficulty for many complex applications leading to the development of more and more language-based interfaces. See Figure 9.2.2. The difficulty of learning and using traditional 3D authoring software packages is that much harder on a mobile device. As such WordsEye's language-based approach fits into this same trend.

9.2.2 GUI Paradigm Shift

There has recently been a shift in graphical user interfaces away from the traditional desktop WIMP metaphor in several ways. In addition to the interaction style, the input and outputs are also different and often based around capabilities of mobile devices.

- Wider range of gestures that take advantage of location-based affordances. For example, XBox Kinnect [Wikipedia, 2016f] allows game players to move as they would in real life to control elements in the game. These new interfaces allow users to leverage skills they have already learned.

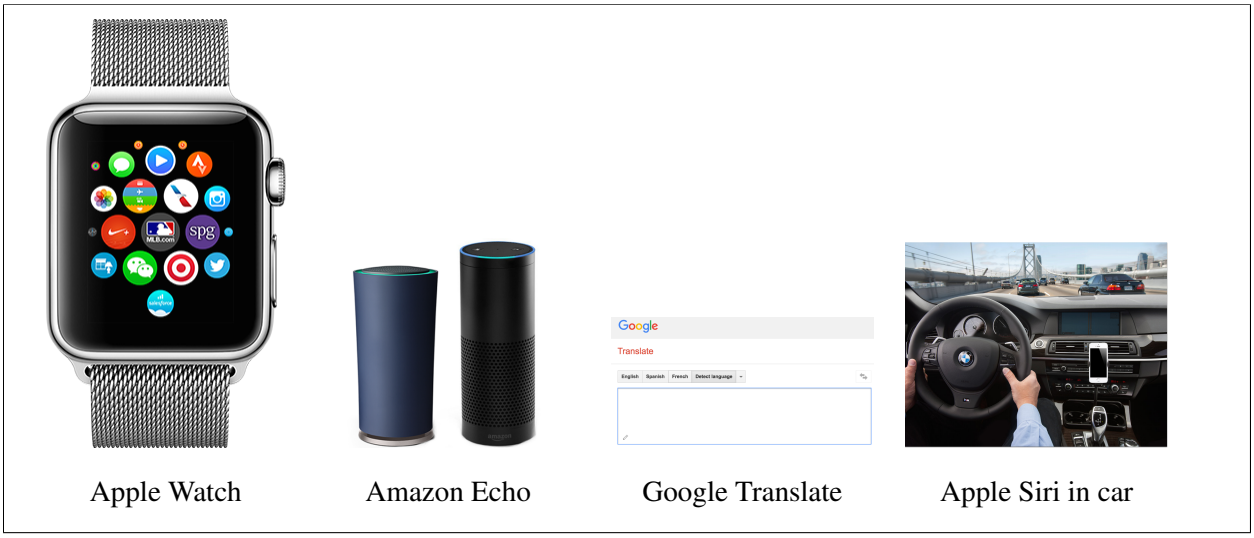
- Natural language processing on textual or voice input. Siri [Wikipedia, 2016l] is a language-based interface for Apple mobile devices that allows users to issue commands to control and access functionalities such as restaurant searches, calendars, and contact lists.

- Wearable interfaces such as Apple Watch [Wikipedia, 2016c] let users issue voice commands and utilize the watch display while keeping their hands unencumbered. Google Glass [Wikipedia, 2016e]

was a hands-free wearable display mounted to an eyeglass frame with a voice/language interface that allowed people to issue commands by voice while seeing augmented reality overlays in the display

- Virtual Reality (VR) and Augmented Realilty displays. Oculus Rift is a popular high performance VR display. As evidence of the market momentum for products like this, the company was acquired for 2 billion dollars in 2014 by Facebook. [Wikipedia, 2016i]

- Integrated input channels. A typical mobile phone includes a built-in camera and location-awareness. This allows applications and user interfaces to include the surrounding environment in different ways. As a result, the range of applications is much wider than found on the more closed world of a desktop-based paradigm.



Apple Watch Amazon Echo Google Translate Apple Siri in car

Figure 9.2: Hands-free and language interfaces

WordsEye's textually generated scenes are part of this trend. Rather than learning complex sets of menus, users can just describe what they want. Speed and accessiblity are given a premium over detailed control. In addition, photos and eventually user-scanned 3D elements can be integrated into scenes, leveraging the capabilities of mobile devices. A mobile text-to-scene app can take advantage of following:

- Touch screen to move the camera. Also, while our current system does not support it, a natural further step would be to allow users to draw with their fingers on the touch screen and give those elements names and then integrate them into language-generated scenes.

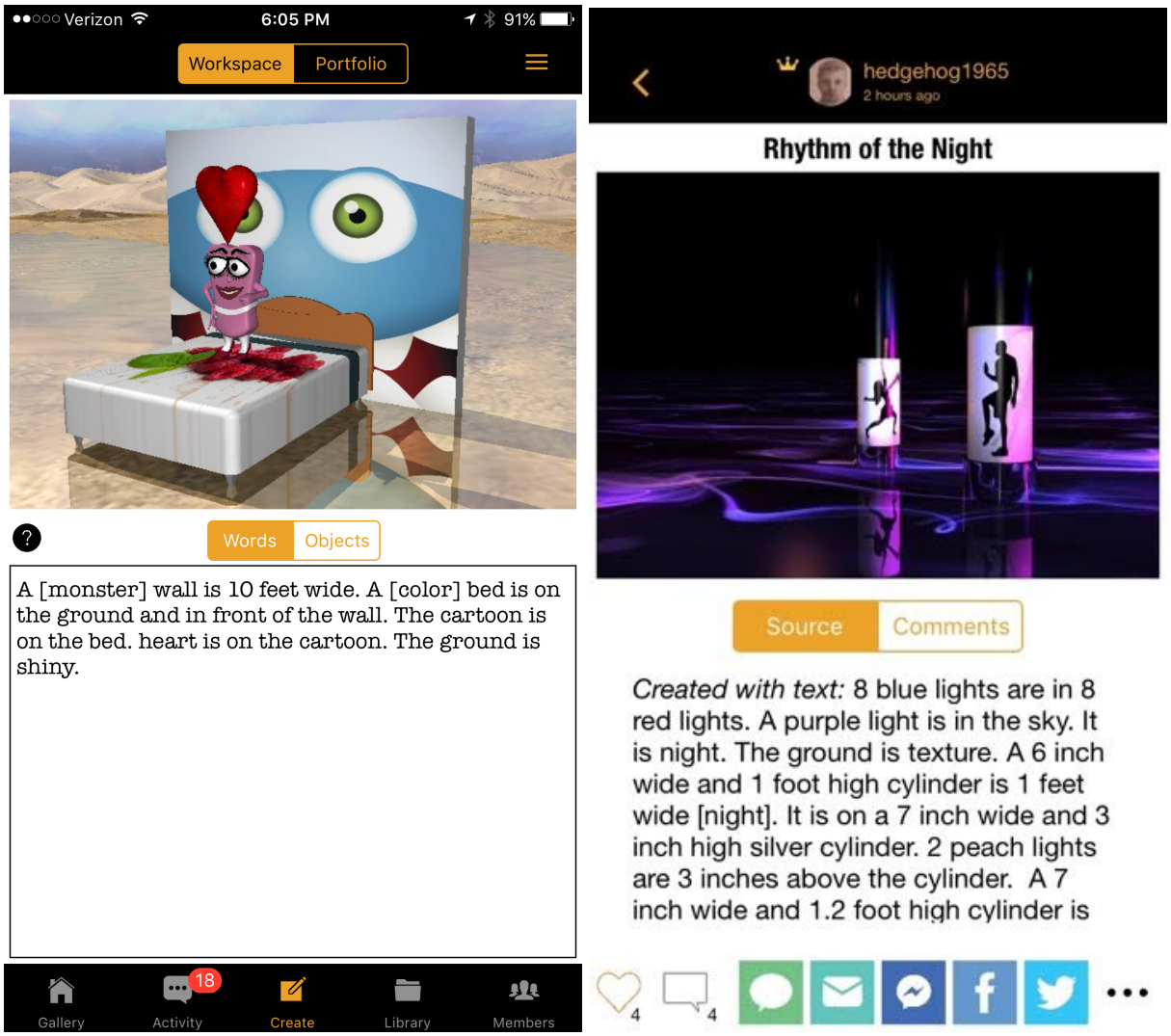
- Voice recognition to say rather than type the text
- Location. WordsEye scenes, in the future could be geo-located.
- Camera. A photo can be taken and inserted into the scene. This can be done as simply as saying *my last photo is on the wall*. Or the user can tag their photos with words. To differentiate these from the built-in lexicon, users must put the tagged wordsi in brackets. In the future WordsEye could support 3D scanning input.
- Text messaging integration

While voice interfaces offer a very easy way to enter text, they are less facile at modifying utterances. The process of creating scenes from language is an iterative one where the user builds up a scene gradually based on what they have already described. The user will often go back and change sentences as they refine their scene. As such, the process is much like any type of writing or drawing. Nothing is final. A pure voice interface does not allow this type of editing. Instead, a WordsEye app can adopt a hybrid interface where the voice input is automatically transcribed and presented to the user as normal text. The user can then insert new text or modify existing text either through voice or by typing.

Product market fit [Wikipedia, 2016k] is concerned with a market need and a given product's ability to meet that need. For WordsEye, we have identified 5 main business areas: social media, instant visual messaging, education, an online picture generation tool for general purpose use, and virtual reality applications. A number of features in the system (such as High Definition renders, user-uploaded content, and specialied thematic 3D objects) can be monetized in a so-called "freemium business model" [Teece, 2010]. In addition, for the education market, teacher portals, customized galleries and example scenes could be offered as a tiered level of service.

9.3 Web and Mobile User Interface

The WordsEye system (`http://www.wordseye.com`) has been used by over twenty thousand users on the web to create over 20,000 scenes over a 10-month period. Many of these textually generated pictures were posted by their creators to an online gallery, and to social media sites such as Facebook. The ease and speed of creating and modifying scenes has led to pictures being used as a form of social interaction that we call *visual banter*. For example, users will frequently open an existing scene they saw in the gallery and make a few changes in the text in order to create a variation of that scene. See Figure 9.4. In this section



a) Mobile App Create Screen

b) Mobile App View Picture Screen

Figure 9.3: WordsEye mobile app

we describe the interface elements, the types of scenes users have created, and present site statistics and an analysis of what we have learned from usage in the real world.

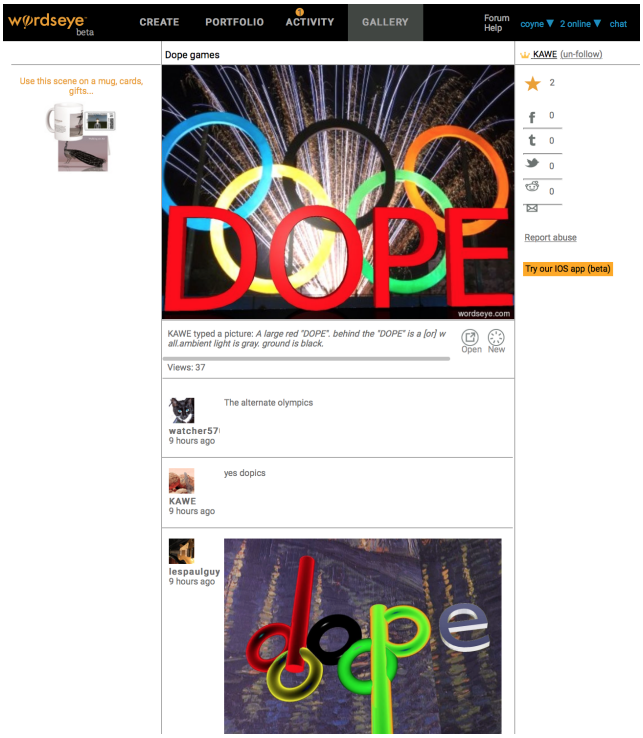
9.3.1 Scene Creation Page

The WordsEye web site consists of several pages. The main "Create" page is where users can create their scenes. See Figure 9.5. This provides the following elements:

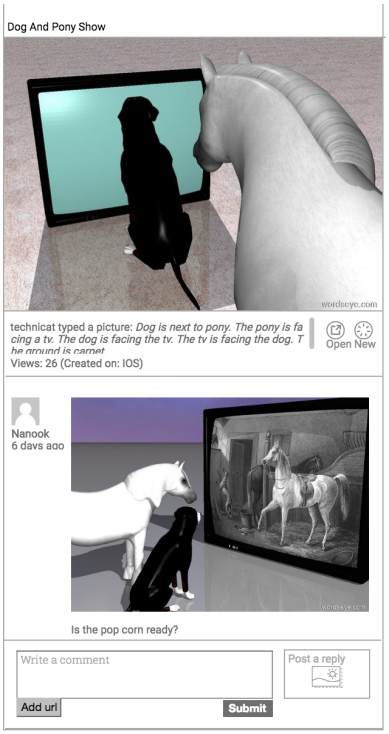
- A text entry area and an image area to display the resulting scene along with buttons for auxilliary functions such as undo and saving.
- A scene object palette to swap among different possible objects for the same word
- Camera navigation modes to interactively move the camera (pan, rotate, zoom, object aim)
- A notification mechanism to highlight any text that cannot be processed. We have found that notifications for errant text has helped users notice and correct their errors. See Figure 9.7.
- A walkthrough tour of the system's capabilities and a set of introductory examples that the user can modify
- A *template mode* that lets the user change certain parts of the text in a "Mad Libs" ([Wikipedia, 2016g]) style. The system uses its built-in coreference resolution mechanism to only allow the first reference to an entity to be modified. In addition some modifiers such as numeric values and non-attributive adjectives or colors in the predicate position are allowed to be changed. See Figure 9.6.
- An object browser. See Figure 9.9.

9.3.2 Gallery and Other Pages

Users manage their saved scenes in the personal Portfolio page (figure 9.9). Other pages include a Gallery (figure 9.8) where users can post their scenes. Every scene in the Gallery can be individually displayed, "liked", and commented on. Comments on scenes can include other newly created scenes, leading to a form visual banter that takes advantage of the ease of creating scenes (Figure 9.4). To promote social interaction, users can also "follow" other users. The Activity page shows a summary of recent posts, comments, and likes by people that the given user follows.



a) View picture and commenting page



b) visual banter (closeup)

c) Movies: Psycho

d) Movies: Nosferatu

Figure 9.4: a) and b) visual banter examples. c) and d) thematic (movies)

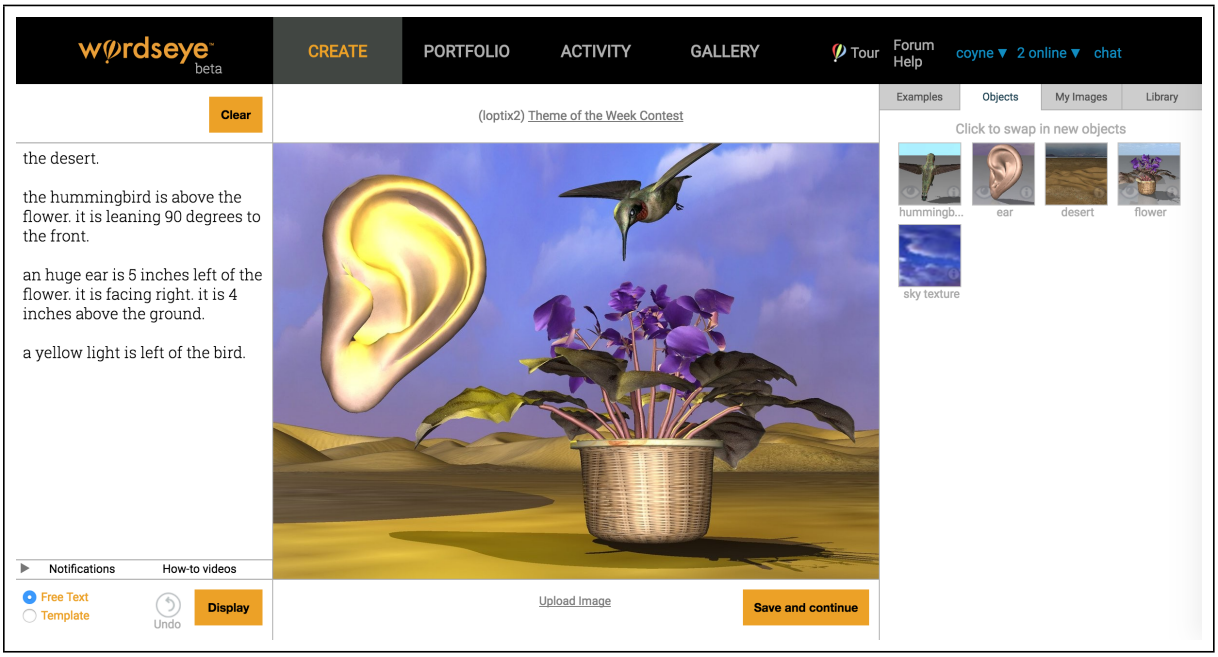


Figure 9.5: Create page user interface

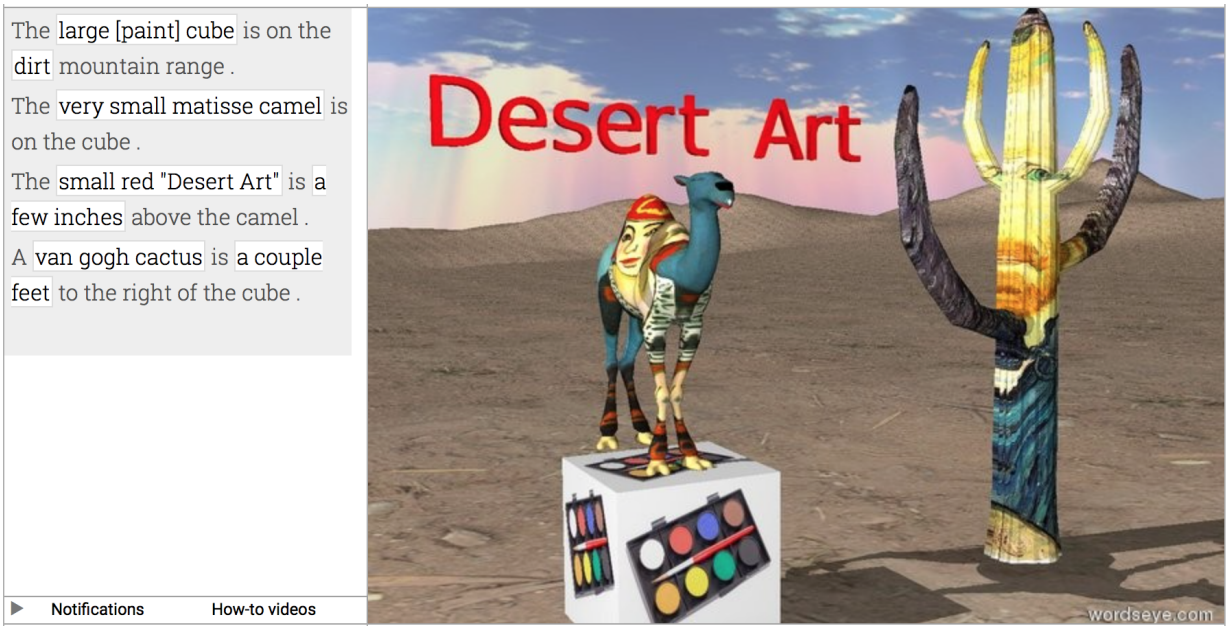


Figure 9.6: Template interface: only the first reference to each noun (eg “cube”) is editable.

Due to the textual nature of scenes, WordsEye scenes provide a unique advantage over other images – they can be easily searched (by automatically accessing the text and objects that created the scene) without

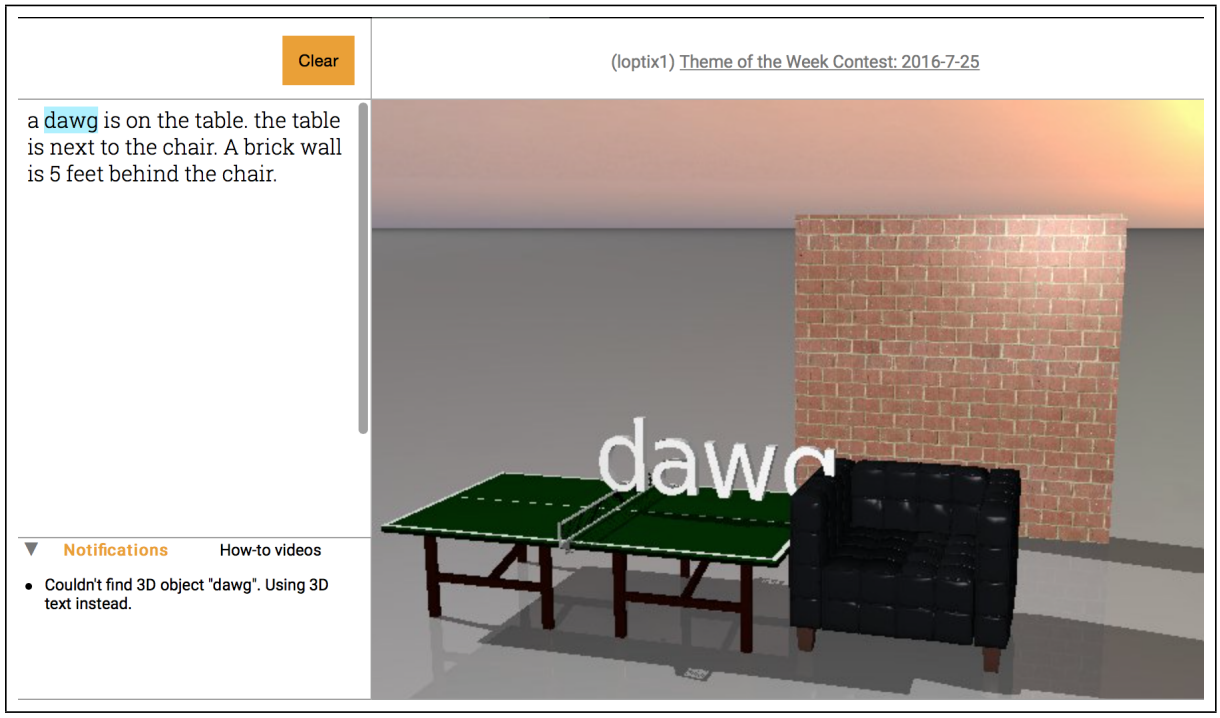


Figure 9.7: Error notification. Note that the incorrect word is highlighted. In this case, the system will also substitute 3D text into the scene to give the user additional visual feedback. This particular fallback "feature" is useful in educational settings where the student can see if they misspelled a word.

relying on assigned keywords or html embedded in the page. In fact, this even opens up the possibility (not implemented on the current system) to perform semantic search on scenes. For example: "find all scenes with more than 3 objects where one object is colored blue".

9.3.3 Web API

Almost all functionality on the website is accessible via a Web API. This same API is used by the ios Mobile app. The API provides functionality to the following:

Creation: workspace, vocabulary, image-keyword-vocabulary, user-image-keyword-vocabulary, choose-scene-object, scene-objects, object-dictionary, user-properties, promoted-content, save-preview

Gallery: gallery, view-picture, register-shared-scene, user-homepage-info, members

Comments: add-comment, edit-comment, remove-comment, get-comments, get-chat-channels, update-chat-read-timestamp, forum

Likes, Following, Leaderboards: add-like, remove-like, likes on scenes, follow, hashtag-follow, leader-

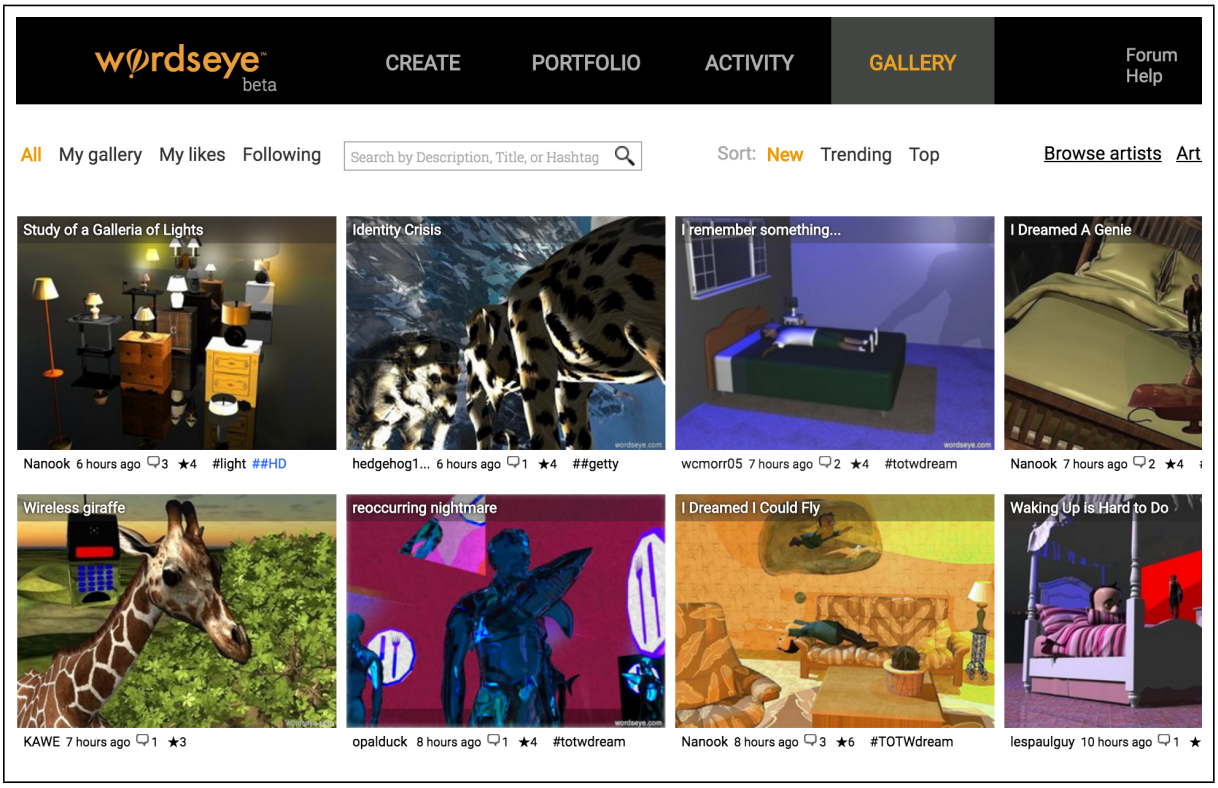


Figure 9.8: User gallery – over 6,700 scenes have been posted to the WordsEye Gallery by approximately 1,700 users during a 10-month period

board, hashtag-leaderboard

Activity: activity, activity-unread-count

Portfolio: picture-operations, portfolio

User Image Uploads: user-images, install-images, import-graphics

Misc: register-view-video

General Info: bbcode, json error structure and codes

Login and Account: login, logout, active-users, account, user-limits, system-limits

9.4 Analysis and Evaluation

In testing the system online for 10 months, 20,000 users have created 25,000 final rendered scenes and have posted 1,700 scenes to the gallery, plus other scenes to social media over a 10-month period (figure 9.4). The monthly active ratio of users to total registered users has averaged around 17%. As a point of reference,

LinkedIn has about a 25% monthly active user ratio. [Journal,].

9.4.1 Problems Observed

We have found, by examining logs of user input (especially by new users who have not yet learned the limitations of the system) that the types of errors made on the online site by users are similar to those made by the system in the WordsEye versus Google test (figure 8.5). In addition we have observed two general areas that make it hard for some users to be as successful with the system as they would like.

1) Users need to learn the system's limitations (both linguistically and graphically). When we first started testing, users would try to use the system to do more than it is capable of. This would often involve specifying background settings, human actions, and shape modifications to objects or their parts (for example, having a human character wear a certain type of clothing or having a particular hairstyle). To address these problem we added a walk-through that all users see before using the system as well as some short introductory videos they can optionally watch. And as mentioned, we added a notification mechanism to give user feedback when sentences were not able to be processed. This has not eliminated the problem but has helped. Ultimately, the goal should be to make the system more powerful and handle a wider range of input (such as action verbs). The work in this thesis has been focused on adding the knowledge and processing capabilities needed to eventually support these areas.

2) Not all users are good at composing visually appealing scenes. It takes some artistic skill and design sense, and also some practice. We have found that some of our best users have an artistic or graphic design background. Some users type quite elaborate text input but wind up with visually unattractive scenes. See Figure 9.11. To some degree, if the system had more graphical capabilities and rendering effects, then scenes would look better by default (since some of the look has to do with rendering and lighting defaults). Also, simple changes like supplying an interesting default background may go a long way. So that is something we plan on trying. Other areas for future work are investigating techniques that might be possible to make more automatic choices and context-dependent defaults to optimize the aesthetics. For example, there could potentially be a "smart camera" button that framed the scene in different appealing ways. .

9.4.2 Surveys

We asked our users in our weekly newsletter to fill out a survey. We received 24 responses. These surveys, while small in number and from a self-selecting sample, provide some additional insight as to what people

like and do not like about the technology and the site. Many people responded that they would like more 3D objects and render effects in the system, suggesting that the breadth of the system is an important factor. In other words, rather than being a sandbox where limited to a specific domain, users enjoy the freedom of being able to compose scenes with almost anything in it. We have noted this same tendency in two other ways. First, the system allows users to upload and name their own 2D images and use those as textures on objects. Quite a few users take advantage of this feature (653 out of the 4,500 users who have rendered a final scene). In addition, whenever we have added new 3D objects to the system, users immediately use them – content is king! In summary, the visual content and look of scenes is very important to users. The text interface is an easy entry point. As indicated in the survey, some users also very much enjoy "the process" of creating scenes.

9.4.3 Textually Generated Scenes – A New Artistic Medium

It is almost a truism that technology drives artistic styles. This is true in music from Beethoven's exploiting advances in piano technology to, more recently, the electric guitar and digital synthesizers and samplers in popular music [Katz, 2010]. See also [Smith, 1970] for a discussion of the interaction of science and technology with the visual arts. WordsEye scenes fall within the medium of computer graphics. Computer graphics can mimic other styles from photorealistic to cartoon rendering to other existing artistic media. So in that sense, text-generated scenes are no different. However, the techniques for producing a given work can be considered part of the work itself. This reframing of the artistic process is a hallmark of modern and contemporary arts. For example, John Cage's explorations in aleatory music are as much about the constraints of the process imposed as the "final" piece itself.

At a more prosaic level, one can easily observe that the means to produce a given work will often change the nature of the work. Word processing software fundamentally provides the same end-functionality as typewriters or even handwriting, but the resulting texts will be different due to the ease of cutting-and-pasting, editing, searching to replace phrases, and so on. Letters written in longhand in the 19th century are very different than short text messages replete with abbreviations and emoji composed on a mobile device.

Similarly, textually generated 3D scenes, necessarily impose a different aesthetic due to both their limitations (lack of detailed control) and the speed and freedom they offer. It is also a medium where chance and ambiguity plays a different and integral role, since even at the word-level, the particular 3D objects selected are not necessarily known or pre-visualized by the user from just the input text. Instead, the user (or

artist) sees them in the context of the scene after entering their text. Also, the scene is controlled at a more abstract level – at the level of language and objects and spatial relations versus at the level of actual graphics (polygons and 3D coordinates). As text-to-scene systems become more powerful, this indirect control can become even more indirect as higher level language is used to produce the scene (as shown in the examples in Figure 1.1). See Figure 9.15 for examples of the range of content produced on the current online system by its users.

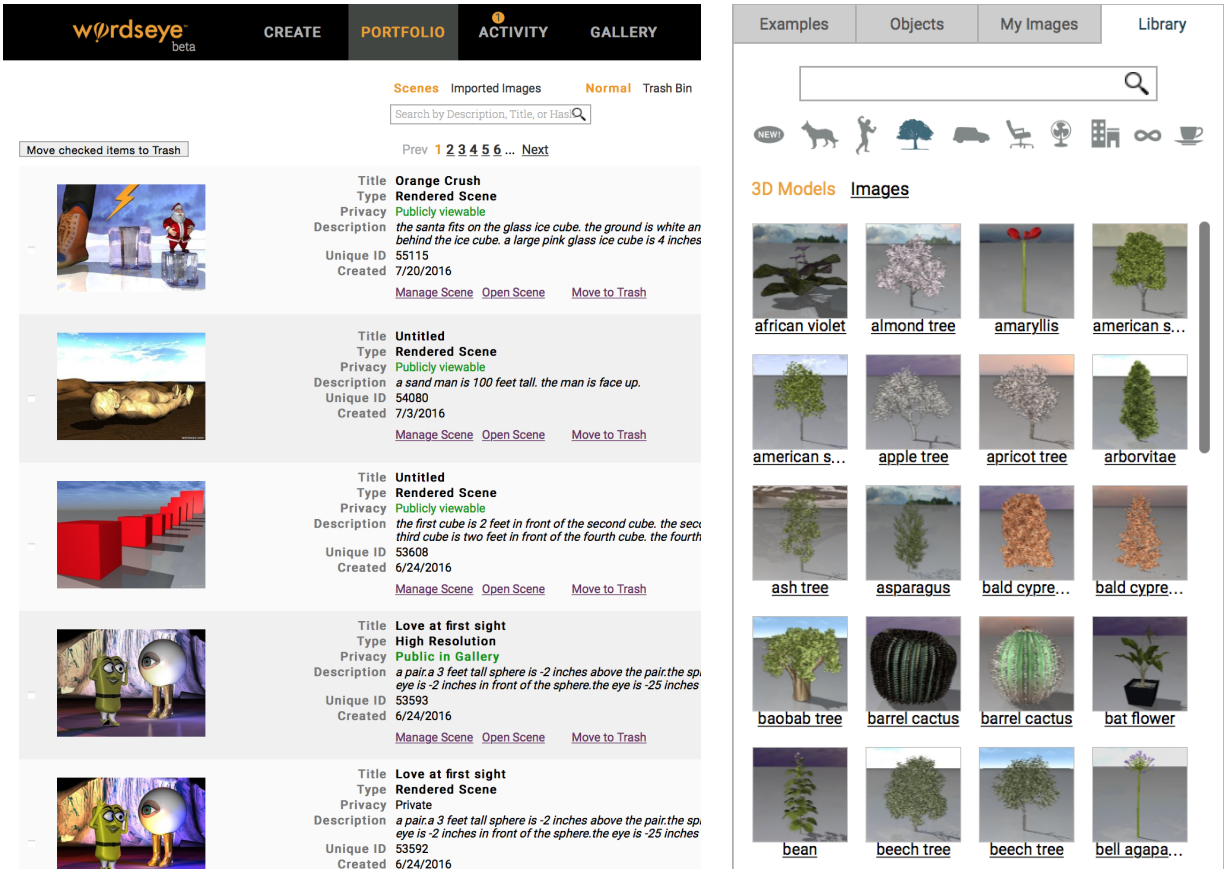
9.5 Conclusion

In our online testing we have packaged the technology into an actual online tool for real-world users. We have proven its utility as a 3D graphics authoring tool with 20,000 users having created 840,000 preview scenes and 21,000 final scenes using the system. Some hardcore users are online over 100 hours per month. We have found that others use it only once or twice but lose interest. We believe that the percentage of repeat users will increase as the system becomes more powerful.

WordsEye allows many users to create visual content they would not have been able to otherwise. The types of content created can vary. We have observed a wide range of content types our site: art (both realistic and abstract), humor, political or social commentary, storytelling, personal photos, and thematic (e.g. related to a topic of interest such as gaming or popular culture). Examples of the types of content produced by users are shown in Figure 9.15.

Of course, there are tradeoffs in creating scenes with textual descriptions versus traditional 3D authoring systems – one necessarily gives up a tremendous amount of control by using text, and the economy of expression comes at a price. Also, as pointed out above, there are several areas (e.g. posed characters and shape deformations) where WordsEye's graphical capabilities are limited. Nonetheless, there is a great market potential for text-based scene generation, due to the ease, speed, and power of the process versus other methods. It is well suited for mobile devices where screen real estate is limited. Many users enjoy the process of creating scenes with text.

> "I just can't wait for a @wordseye bot to be in messaging clients." – @whiteandwoke
>
> "A dog park for brains" – @anxieting
>
> "Everyone on Tumblr is going to be using this for memes, I can just tell." – @the-pluto-fandom

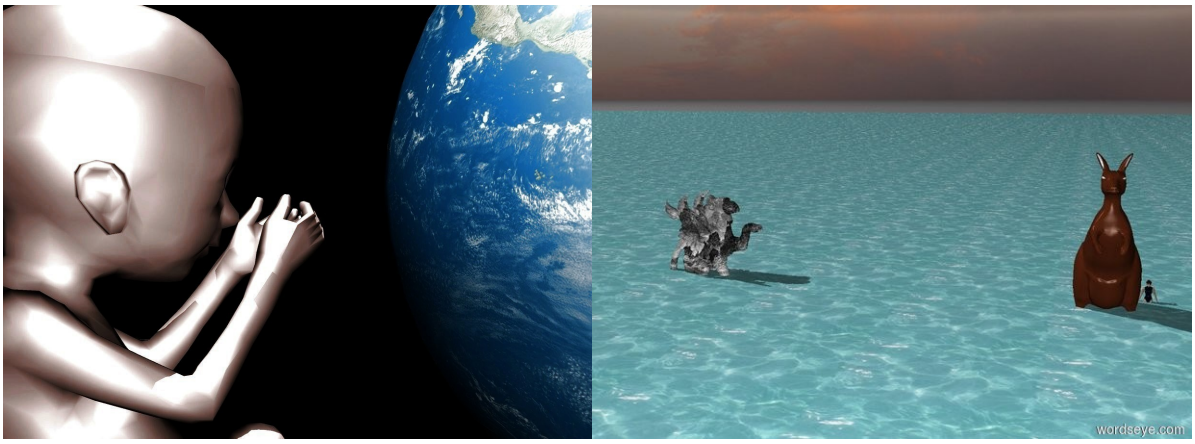


a) Portfolio b) Object Browser

Figure 9.9: User interface components: a) *Portfolio* page contains all of a given users saved scenes and allows their properties to be changed (e.g. given a title or shared to Gallery). b) *View Picture* page allows users to "like" a scene, shared it to social media, comment on it (including visual comments using other scenes) c) The *Object Browser* lets users browse or search for 3D objects and images

Users	20,272
Sessions	34,582
Repeat users	5,698
3-time users	2,071
5-time users	645
7-time users	346
10-time users	192
Preview depictions	843,273
Camera moves	705,711
Final renders	21,147
Users who did final renders	4,512
HD renders	4,269
2D effects	23,108
Likes made	12,731
Scene views by users	79,785
Users who viewed other scenes	3,993
Gallery posts	6,668
Users with gallery posts	1,667
Shares to social media	1,407
eCards sent	243

Figure 9.10: Usage statistics for WordsEye website over a 10-month period



(a) *The ground is black. The sky is black. A fetus. The tiny earth is in front of the fetus. A white light is above the fetus. It is night.*

(b) *the small girl in the ocean. the sky is scary. there is a kangaroo next to her. the kangaroo is very large. there is a sea monster far from the kangaroo.*

Figure 9.11: Scene quality (a) well-composed scene (b) poorly composed scene

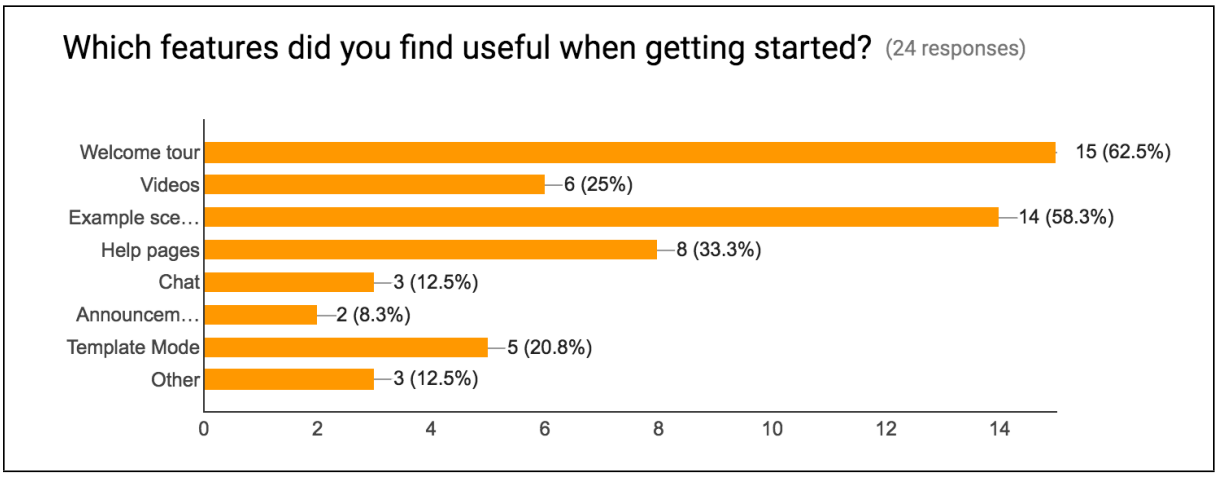


Figure 9.12: User survey (getting started): Users were asked what helped them learn the system. The most useful aids were the welcome tour (which briefly describes the features of the system) and a set of example scenes that are available for all users to try (and modify) as they work on their scenes.

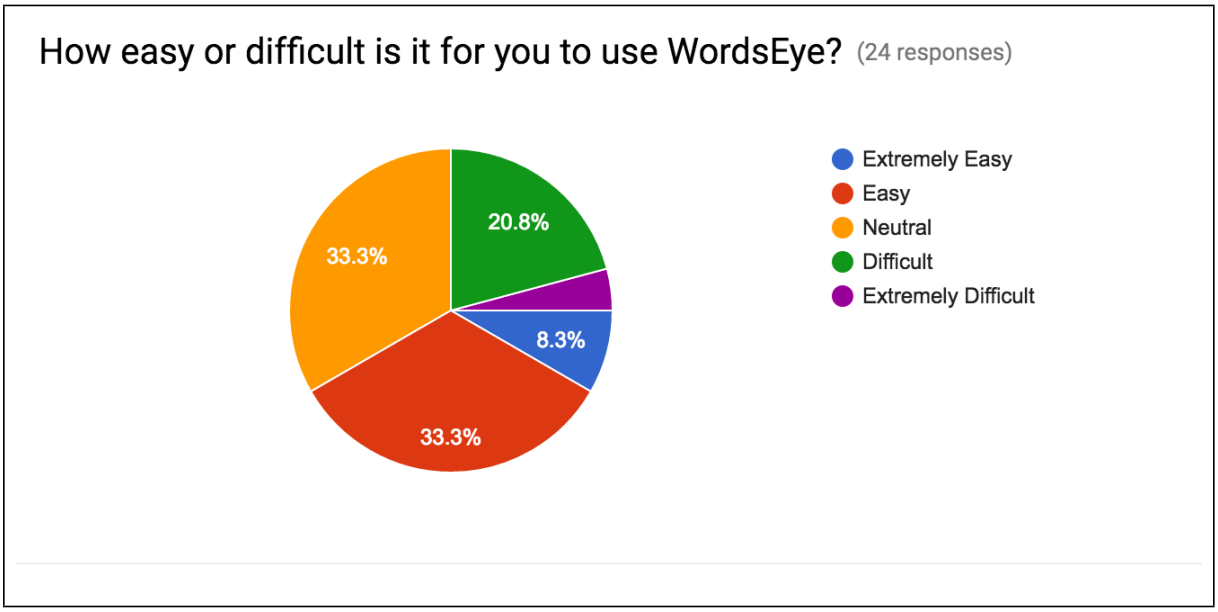


Figure 9.13: User survey (difficulty): About 42% of these users found the using system easy or very easy, 33% found it neutral, and 25% found it difficult or very difficult.

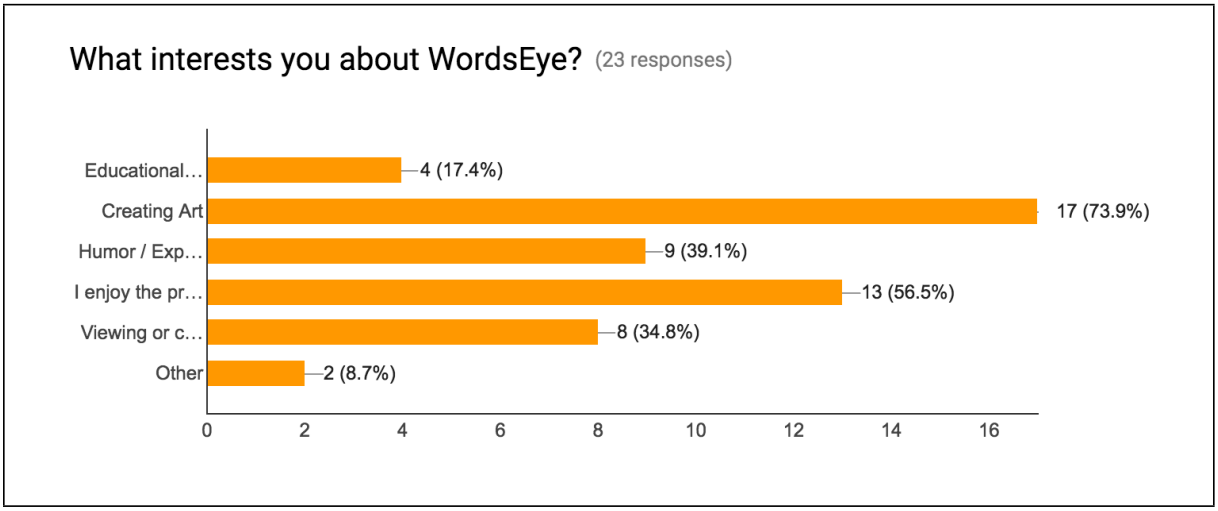
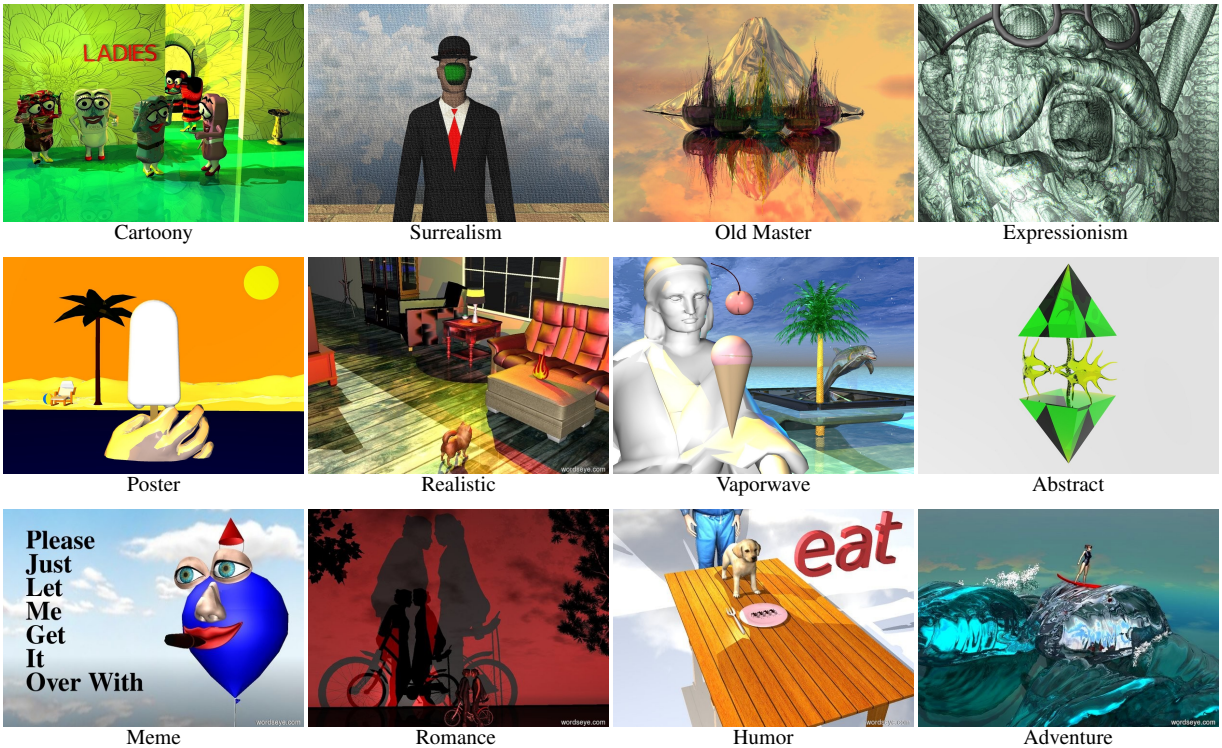


Figure 9.14: User survey (interests): A strong majority of users who answered the survey liked using the system for *creating art*. This is in keeping with its being presented as an easy way to create 3D graphics. A large number also *enjoyed the process*. This conforms with what we heard from some students in our education study 7 who said they enjoyed the challenge of decomposing their thoughts and visual ideas into simple concrete language. Other reasons with lower scores were: *humor and expressing opinions*, *education*, and *viewing or commenting on other users' scenes.*



Cartoony | Surrealism | Old Master | Expressionism

Poster | Realistic | Vaporwave | Abstract

Meme | Romance | Humor | Adventure

Figure 9.15: A variety of styles, genres, and content types. Images created onsite by users. See Appendix C for full-sized pictures and accompanying text and user information.

Chapter 10

Towards a Field Linguistics Tool

10.1 Introduction

In this Chapter, we describe a study we performed to evaluate the potential of WordsEye as part of a larger system for field linguists studying and documenting endangered languages. Our pilot work for this project was supported by a 12-month grant (#1160700) from the National Science Foundation. See [Ulinski *et al.*, 2014a] and [Ulinski *et al.*, 2014b].

Our goal was to create a first version of WELT (WordsEye Linguistics Tool), a linguistics toolkit based on WordsEye to elicit spatial language from speakers of endangered languages in order to document those languages. During this period, we created the WELT English user interface and tested it with a native speaker of Nahuatl, created a corpus of Arrernte-specific objects for WordsEye, developed a syntactic parser for Arrernte, researched case issues in Arrernte, and came up with an overview of the pipeline for WELT L2 (where WordsEye depicts text entered in a non-Engish langauge using grammars and lexicon developed for that language).

The endangered language lexicons, grammars, and diagrams and the overall WELT user interfaces are the work of Morgan Ulinski, and the description of Levinson's work below is due to Richard Sproat. The contribution made by the author is in the WordsEye system itself (which included specialized vignettes and graphical objects pertinent to the project) as well as modifications made to object highlighting in the WordsEye user interface to focus elicitation on specific objects or parts of objects.

Spatial relations have been studied in linguistics for many years. One reasonably thorough study for English is Herskovits [Herskovits, 1986], who catalogs fine-grained distinctions in the interpretations of

various prepositions.[1] For example, she distinguishes among the various uses of *on* to mean "on the top of a horizontal surface" (*the cup is on the table*), or "affixed to a vertical surface" (*the picture is on the wall*). Herskovits notes that the interpretation of spatial expressions may involve considerable inference. For example, the sentence *the gas station is at the freeway* clearly implies more than just that the gas station is located next to the freeway; the gas station must be located on a road that passes over or under the freeway, the implication being that, if one proceeds from a given point along that road, one will reach the freeway, and also find the gas station.

10.2 WELT English: User Interface for Elicitation

In this section we describe WELT English, the user interface we created, that provides the tools needed to elicit language with WordsEye. It creates elicitation sessions organized around a set of WordsEye scenes. Using 3D scenes for elicitation allows more flexibility if the linguist wants to focus on different aspects of the scene (ether by highlighting different objects or adjusting the camera position) or even modify the scene based on the subject's response. The scene that is currently open in the WordsEye application can be saved, along with textual descriptions, glosses, and notes. Audio for the session can be recorded, and the recording is automatically saved and synced with timestamps for the scenes open in WELT. The audio for a specific scene can thus be easily reviewed later. The user can import sets of scenes saved during previous elicitation sessions, allowing useful scenes from other sessions to be re-used and to easily compare the data collected. A screenshot of the WELT interface for annotating a scene is shown in Figure 10.1.

[1] It is important to realize that how spatial relations are expressed, and *what kinds of relations may be expressed* varies substantially across languages. Levinson and colleagues [Levinson, 2003] have catalogued profound differences in the ways different languages encode relations between objects in the world. In particular, the Australian language Guugu Yimithirr and the Mayan language Tzeltal use absolute frames of reference to refer to the relative positions of objects. In Guugu Yimithirr, one can locate a chair relative to a table only in terms of cardinal points saying, for example, that the chair is north of the table. In English such expressions are reserved for geographical contexts — *Seattle is north of Portland* — and are never used for relations at what Levinson terms the "domestic scale". In Guugu Yimithirr one has no choice, and there are no direct translations for English expressions such as *the chair is in front of the table*.

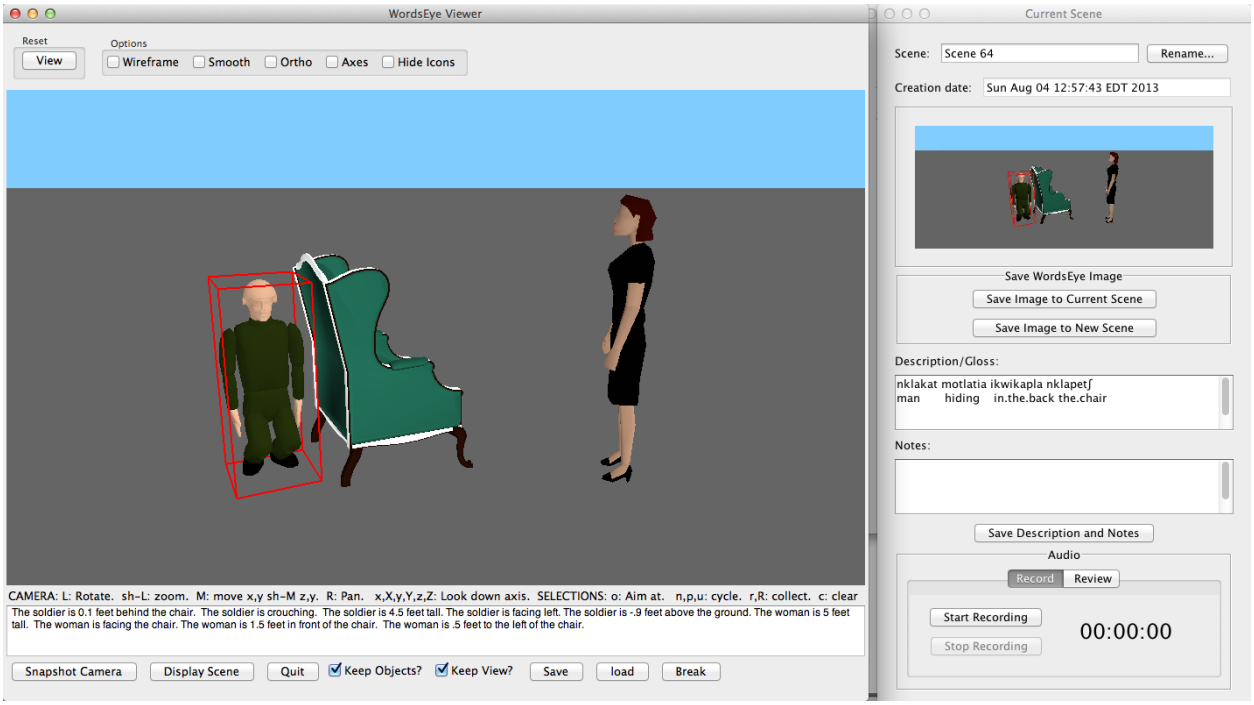


Figure 10.1: User Interface for WELT WordsEye

10.2.1 Evaluation of WELT Interface - Nahuatl Topological Relations

To evaluate WordsEye's usefulness in the creation of pictures for eliciting spatial language, we created a set of scenes based on the Max Planck topological relations picture series [Bowerman and Pederson, 1992]. See Figure 10.2. We were able to recreate 40 out of the 71 pictures in the series using WordsEye (Figure 10.3). One of the main issues that prevented us from creating the full set was that WordsEye does not currently have the objects needed to produce the desired scene. There were also cases where the graphical functionality of WordsEye would need to be enhanced to allow more precise positioning of objects. Some examples of the scenes and the descriptions we elicited are included in Figure 10.4.

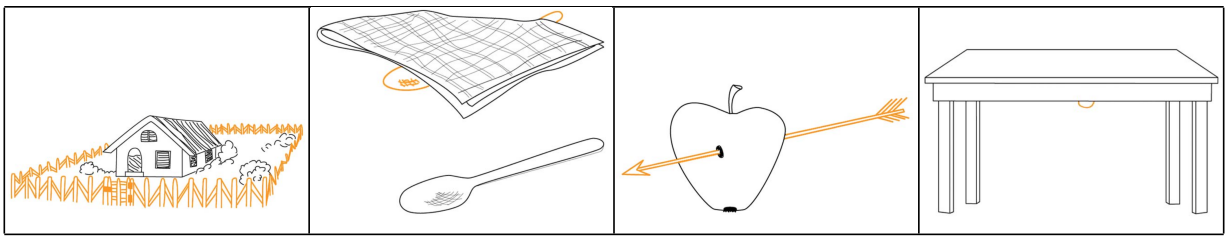


Figure 10.2: Max Planck topological relation picture series

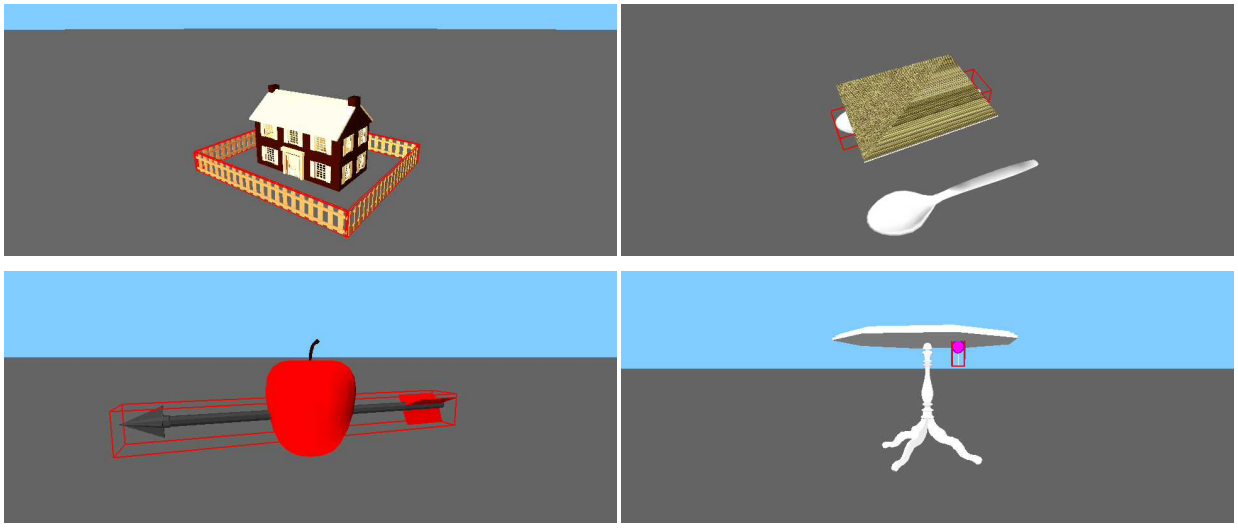


Figure 10.3: WordsEye scenes for topological picture series, highlighting the focal object in red

10.2.2 Elicitations

We used our prepared scenes and the WELT interface to elicit descriptions from a native Nahuatl speaker.

in kwawitɬ tɬapanawi tɬakoja se mansana
the stick pass.thru in.middle one apple

in amatɬ tɬakentija se kutʃara
the paper cover one spoon

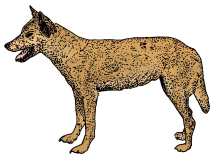
Figure 10.4: Nahuatl elicitations obtained with WELT

10.3 Arrernte-Specific WordsEye Objects and Vignettes

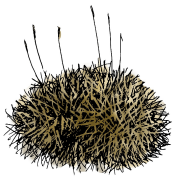
We created a corpus of images and 3D models for WordsEye that are specifically relevant to aboriginal speakers of Arrernte, including native Australian plants and animals and other culturally relevant objects and vignettes. For example, see Figure 10.9 for a scene with a compound object vignette for Australian

rules football (footy) goalposts. We will use these when we create scenes for studying Arrernte. Many of the pictures we created are based on images from IAD Press, used with permission, which we enhanced and cropped in PhotoShop. Examples are included in Figure 10.5. We also created examples of culturally specific vignettes (Figure 10.6).

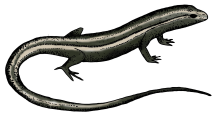
(a) Honey Ant (*yerrampe*)



(b) Dingo (*akngwelye artnwere*)

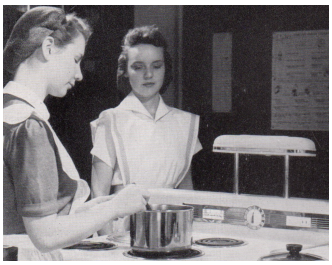


(c) Spinifex (*aywerte*)



(d) Skink (*ikwarre*)

Figure 10.5: Images created for Arrernte WELT (Arrernte translations in parentheses)



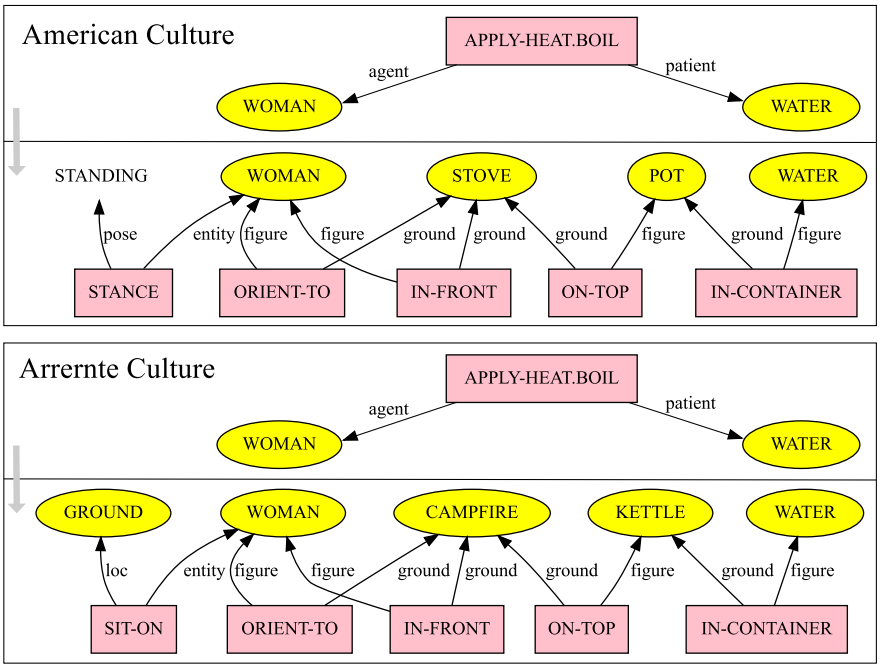


Figure 10.6: Culturally dependent vignette decompositions for English versus Arrernte

10.4 Investigation of Case in Arrernte

One of our goals for using WELT is to study the relationship between the meaning of a sentence and the case of the nouns in it. With the help of Myfany Turpin, a linguist who studies Arandic languages, we collected a set of Arrernte sentences, primarily from [Broad, 2008] and [Wilkins, 1989], that are interesting in terms of spatial language or case. We created a FieldWorks project for Arrernte that includes all these sentences, translated them, and glossed them at the morphological level. We also entered all of the phonological information necessary for the Fieldworks phonological parser. These sentences can now easily be searched either at the surface level or by the glossed morphemes, so we will be able to use them as we continue our work on Arrernte.

10.5 Grammar Development for Arrernte

We collaborated with researchers at Macquarie University to document Arrernte syntax using LFG (lexical-functional grammar) [Kaplan and Bresnan, 1982], automatically generate Arrernte text related to Australian football [Lareau, 2012]. In LFG, linguistic structure is represented by a parallel, linked combination of a surface-oriented constituent structure (c-structure) and a functional structure (f-structure). The f-structure is a dependency structure that models predicate-argument structure, and a suitable interface to VigNet. The grammar for Arrernte is in two parts, a finite state transducer for the morphology, developed with XFST [Karttunen *et al.*, 1997], and the syntactic grammar developed in XLE. It covers basic sentences and NP structure and a few unusual features of Arrernte: split case pronouns, verbless sentences, associated motion, spatial relationships, and same-subject inflection on the verb [Dras *et al.*, 2012].

10.6 Design of WELT L2 Pipeline

One major part of our work on the pilot project was designing the WELT L2 pipeline that takes a sentence in the target language and outputs a scene in WordsEye. The best way to understand the pipeline and its various components is to follow an example through the processing steps. We will use the Arrernte sentence (1) below. See Figure 10.9 for this same sentence entered on the WordsEye WELT user interface and depicted using a footy goalposts compound object vignette.

(1) artwe le goal arrerneme

```
man ERG goal put.nonpast
The man kicks a goal.
```

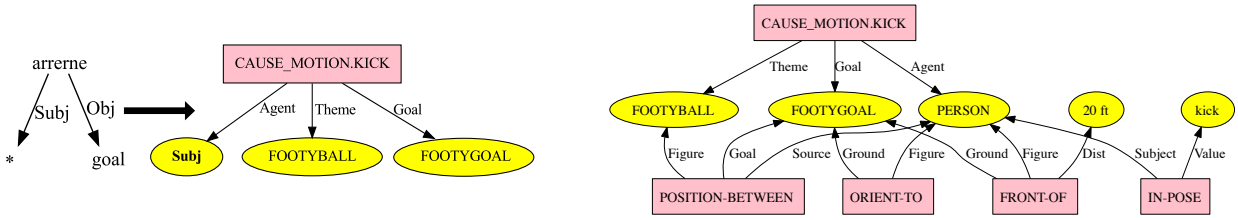
Morphology: The first step of processing the sentence is to run each word through the morphological analyzer in XFST, which transforms the verb *arrerneme* into 'arrerne+NONPAST.' The other tokens in the sentence remain unchanged.

Syntax: The next step is syntactic parsing via our Arrernte grammar. Processing the sentence with XLE gives the c-structure shown in Figure 10.7(a) and the f-structure shown in Figure 10.7(b).

(a) (b)

Figure 10.7: C-structure (a) and f-structure (b) for *artwe le goal arrerneme*.

Semantics: The mapping from syntax to semantics is done through a set of rules specified by the linguist, taking the f-structure as input and outputting the high-level semantic representation required for VigNet. Figure 10.8(a) is an example of a rule that could be included in a mapping from Arrernte syntax to semantics. The left-hand side of a rule consists of a set of conditions on the f-structure elements and the right-hand side is the semantic structure that should be returned. The linguist will also need to create a simple table mapping lexical items to the VigNet concepts corresponding to graphical objects. We created a mapping from each of the lexical items in the Arrernte LFG grammar to VigNet concepts; a partial mapping of the nouns is shown in Table 10.1.



(a) (b)

Figure 10.8: A syntax-semantics rule (a); the semantics (lexical and graphical) for sentence (1) (b).

Lexical Item	‘artwe’	‘panikane’	‘angepe’	‘akngwelye’	‘apwerte’	‘tipwele’
VigNet Concept	ent.person	ent.cup	ent.crow	ent.dog	ent.rock-item	ent.table

Table 10.1: A mapping from nouns (lexical items) to VigNet objects

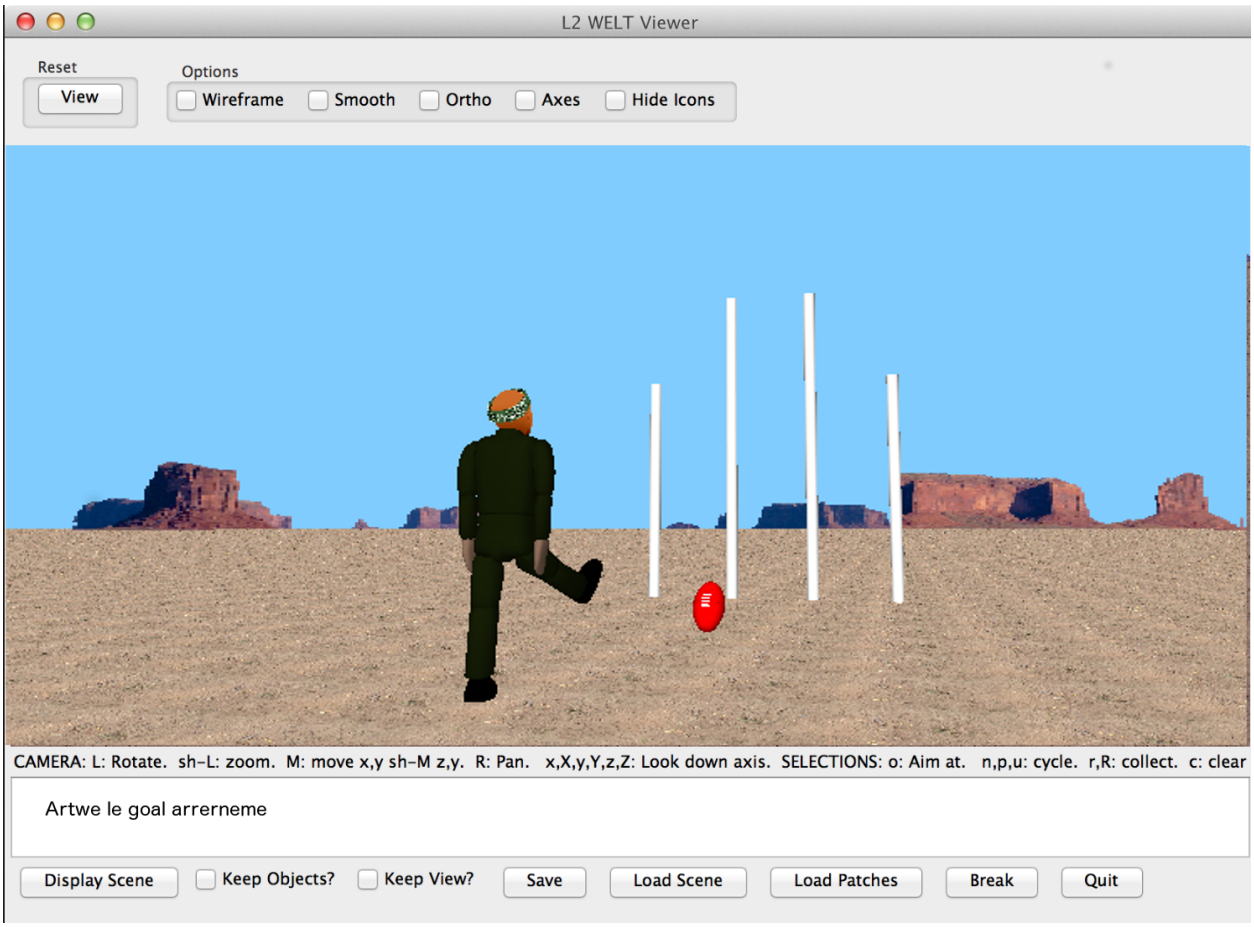


Figure 10.9: WELT interface depicting Arrernte sentence (1)

10.7 Conclusion

This was a preliminary study. We hope to perform future work in the area of field linguistics using WordsEye to dynamically generate scenes for elicitation and to study spatial language.

Chapter 11

Conclusion

In this chapter, we begin by discussing what we have learned while building and testing WordsEye in different application areas. We then summarize the contributions made by the work presented in this thesis and describe several potential areas for future research. Finally, we conclude with a brief observation.

11.1 Different Needs for Different Applications

In the previous chapters we described and evaluated text-to-scene technology in four different use-case scenarios (education, image search-as-generation, online 3D authoring, field linguistics). Each actual use-case has somewhat different (sometimes conflicting) requirements for the system in the areas of a) grammaticality and spelling, b) getting notifications, c) predictability versus accuracy for reference resolution, size constraints and other behaviors, and d) requirements for the underlying 3D content.

Graphical objects: The range of possible graphical effects and objects to be included in scenes is important for all scenarios but somewhat different. In some scenarios (such as education or field linguistics), it can also be important to upload into the system specific content, for example historical figures relevant for students studying a particular era of history or culturally relevant objects for field linguistics. In our authoring system, new objects were used as soon as they were made available, showing their importance. In a field linguistics application, culturally relevant graphical objects are important in the elicitation process.

Predictability versus aggregate accuracy: In applications such as authoring or educational systems, the overall *accuracy* of certain system behaviors can be in conflict with its *predictability*. For example, WordsEye's reference resolution (Section 6.3), size constraint (Section 4.3.2), entity selection (Section

6.4.3) and spatial reference frame behaviors (Section 3.8.1) are relatively predictable, with a simple set of behaviors that work reasonably well in most cases for the type of language used currently to create scenes.

For an authoring system, the computational model naturally becomes part of the user model as the user learns to predict the system's behavior. A slightly more accurate, but less inherently predictable system can be, in practice, more difficult for a user since they will not be able to adapt as easily to the system's patterns. Spatial reference frames, in particular, are very hard to predict from text, and inconsistent behavior in the pursuit of a higher Accurate Reference Frame (ARF) score is probably a fool's errand (at least for an authoring system, where predictability is paramount). In an image search-as-generation system, predictability is less of a factor than overall accuracy, and a high ARF score might therefore be more important. In an educational context the tradeoff can vary and depend on whether the emphasis is on learning to master language as it actually used in the world versus using the system as more of tool in a storytelling mode.

11.1.1 Needs by Application

Educational Scenarios:. For applications centered on literacy skills, it is important to give feedback if a sentence is malformed in some manner. Students need to learn spelling, punctuation, and grammar, and the system can inform them when it does not understand their input. In fact, in our HEAF experiments, learning to put periods between sentences was one of the first things students had to get in the habit of doing if they wanted to get a scene! As a result they started to form good writing habits. For image search-as-generation or authoring scenarios (as well as some process-oriented scenarios), a looser handling of input, in order to get a scene with the minimum amount of effort, is desirable. In fact, with our online authoring system, we have seen many users from non-English speaking countries. These users often make more mistakes in grammatical agreement and omit articles than English speaking users. We can tell this by looking at the domain names which often indicate the user's nationality. Of course, we assume that such users are using the system primarily for fun or to create pictures rather than to learn English.

An additional factor for process- or play- oriented systems such as education, is the enjoyment and value in-and-of-itself of finding new ways to get the system to achieve the desired effects. In our education study (Section 7), the students mentioned how they enjoyed the challenge of being able to translate their thoughts into words and into pictures. They specifically found value in both mapping stories into *specific* mental visual images and then mapping those mental images into text that worked with the system.

Authoring system: the user for this application is most concerned with being able to achieve the effects

they desire. Four main factors come into play: a) ease of learning how the system works in order to achieve those effects, b) efficiency of the language being used to achieve those effects with minimal effort, c) range of possible output scenes, and d) the visual quality of the results.

Image search-as-generation system: For this application (Chapter 8) the overall coverage and accuracy of the generated picture is paramount. The visual quality of the resulting images is also important since people typically search for images with some particular purpose in mind. The user of this type is not necessarily iterating on scenes with longer descriptions or learning the system in any deeper way, but that option could be useful. A related application to image search-as-generation would be *visual summarization* – the automatic depiction of free-range text on the internet (Section 11.3). Such an application, even more than image search would not care about predictability since it is not user-driven.

11.2 Contributions

This work presented in this thesis makes the following contributions:

- **Vignette Semantics**, a new lexical semantic theory based on *vignettes*, *affordances*, and graphical knowledge. The relational knowledge representation utilizes an ontology of concepts and an inventory of semantic relations (including lexically derived relations and vignettes) that apply to concepts. Semantic relations can express both high-level semantics (such as verb frames) as well as lower-level graphical and functional properties on concepts, including spatial affordances. (see Section 3.7)

  Vignettes are a new conceptual building block that bridge between function (lexical semantics) and form (graphical semantics). They can apply to any relational structure: actions, locations, figurative representations, compound objects, and even simple spatial relations. Vignettes decompose high-level semantics into configurations of objects that in some sense resolve and disambiguate that meaning. In doing so, they capture more meaning than the sum of their constituent graphical elements. The notion of vignettes is extended to include *abstract vignettes* (such as applying an instrument to a patient on a work surface) and the integration of spatial affordances. Together, these exploit the visual regularity of objects and actions and hence allow a common decompositional structure to be shared among many actions. Abstract vignettes can be thought of as higher-level graphical primitives that capture common configurations of visual meaning independent of any specific functional meaning.

Vignettes are decomposed to graphical relations by utilizing a set of semantic translation rules. One form of meaning is translated to another, all within the same representational system.

The semantic representations utilize a well-defined set of primitive graphical and ontological relations necessary to describe and modify any entity. The representational framework accommodates a) lexicon and word definitions b) general (unlexicalized) concepts as well as graphical and "sublexical" entities c) input text semantics d) world knowledge f) mappings and decomposition between relations g) contextual knowledge h) graphical semantics.

All parts, wholes, regions affordances and abstract entities are distinct concepts and defined relationally. This allows mereological and abstract concepts to all be individually referenced and modified, while retaining their identity in relation to a larger whole or semantic domain.

In contrast to more traditional function-oriented lexical semantic theories (such as Pustejovosky's Generative Lexicon or Jackendoff's Lexical Conceptual Structures), vignette semantics takes the point of view that the visual representation is a core part of the meaning and hence must be made explicit. Semantic relations and decompositions are used to both define graphical meaning and logical and functional meaning.

- **VigNet**, a unified ontology and knowledge-base for lexical-semantic representation and graphical grounding that instantiates our semantic theory. VigNet consists of a large ontology of concepts and semantic relations tied to a lexicon. This includes the concepts and related knowledge (such as type, parts, default size, and spatial and functional affordances) for a library of 3,500 3D object and 5,500 2D images. We have also formulated a set of vignettes and graphical primitives tailored to using specific spatial affordances as well as mereological primitives used to define reference.

  In addition, Vignet contains a large set of logic-based pattern matching rules that perform semantic translations. These, in combination with graphical semantic properties for objects, can be used to translate semantic representations and ultimately produce a grounded graphical realization. Semantic translation rules access the rest of the VigNet knowledge base and be applied to lexical-to-semantic transformations, semantic transformations, and other semantic decompositions.

  The grounding of the ontology in actual 3D objects allows inferences to be made upwards (inductively) as well as downwards (deductively). For example, we might know that all types and instances of elephants have trunks from the upper levels of the ontology. But we know that certain actual ele-

phants have trunks from the specific 3D objects. We can thereby infer trunkhood (and similary for any property) on concepts in the ontology without having it to be declared. This is discussed in Section 2.3.3.

VigNet will be made publicly available at http://www.cs.columbia.edu/~coyne.

- **New Lexical Semantic Methods**: We have developed ways to process and decompose a number of lexical-semantic constructs in a text-to-scene context. These include gradable attributes, regular polysemy, noun-noun compounds, preposition interpretation, and modifiers on prepositions. Others, such as regular metonymy and regular polysemy, fit nicely into the same framework. These constructs often involve implicit or ambiguous relations that must be resolved. This is enabled by the semantic translation rules and object properties (including graphical properties such as affordances) represented in VigNet. Lexical semantics applies throughout the system, including in reference resolution, where lexical semantic features encoded in the VigNet knowledge base are used to help determine whether lexical references should be merged or not. It also applies in parsing where lexical semantic concepts such as measure terms are used by the parser. We believe this points to the value of lexical semantic knowledge in parsing, more generally, and that such knowledge can be applied to other issues such as prepositional phrase attachment issues.

- A wide-domain text-to-scene system (**WordsEye**) that uses all of the above. WordsEye is the first widely used and evaluated text-to-scene system. It is shown to have applicability in several areas. In education, we have shown significant increase in growth in literacy skills for 6th grade students using WordsEye versus a control group. In image search, we have established a metric for realistic versus imaginative sentences and shown that WordsEye outperforms Google Image Search for imaginative sentences within WordsEye's vocabulary and graphic capabilities. In comparing WordsEye to the AVDT text-to-scene system on our realistic sentence corpus of 250 sentences, we found that the AVDT system would be unable to handle 154 (64%) of the sentences that rely on capabilities outside the range of what that system supports (Section 8.6.2). WordsEye's semantic framework (VigNet) and other research contributions help it get "mostly accurate" or better on half of those sentences. As a 3D authoring tool, we have demonstrated the viability of text-to-scene generation as a social online artistic tool with over 20,000 real-world users using WordsEye to produce thousands of finished rendered scenes. We have also demonstrated potential for WordsEye as a field linguistics tool.

11.3 Future Work

The WordsEye system and the VigNet resource has many areas in which to be improved. Since there are so many, given what is theoretically conceivable, we focus here on plausible next steps at both the research level and in application areas.

11.3.1 Future Research

Many limitations and areas in need of improvement have been listed throughout the thesis. We highlight below some areas of particular importance. The most limiting aspects of the current system are mostly all graphical: lack of posable characters, lack of deformable shapes, limited spatial layout capabilities, no ability to find affordance areas on the fly, non-modifiable parts on 3D obects. We hope to improve all these areas in the future.

Better handling of scene layout constraints: The current methods used by WordsEye for enforcing graphical primitives are too limited. For example, they don't allow enforce soft constraints where an object is left of another at an unspecified distance. Soft constraints can lead to more robust scene layout and more natural object positioning. Another major problem with the current layout mechanism is that all spatial relations are along the main XYZ axes or combinations of those. There is no support to have an object on a tilted surface.

Computationally derived affordances: All affordances are currently "baked" into the objects as parts. This is limiting in many cases. If an object is upside down the old top surface affordance is suddenly useless. One would like the system to compute an affordance on the fly. For example, for some objects like trees and other plants, the irregular geometry makes almost any part of the object a possible spatial affordance, including the branches and the irregular areas near the base of the trunk where roots might bulge up. Animals such as birds and squirrels can find innumerable places to sit.

Poses, facial expressions, and deformable shapes: As mentioned (Section 1.1.2), one major limitation of WordsEye is that objects cannot generally be deformed. In particular, we will continue working to support action verbs by posing 3D characters and emotional terms by deforming faces for facial expressions. These will allow much more text to be depicted. It would also be very useful to support fluids and flexible objects such as cloth and rope. This could be done using a physics engine [Wikipedia, 2016j]. A physics engine could also help in other ways with scene layout, for example, by detecting collisions and finding natural

resting states of objects.

Modifying 3D object parts: It is very natural in language to refer to parts of objects, but modifying them, other than in very simple ways, is not possible in our current system. In particular, modifying part shapes (*the swan's neck is 3 feet long*) would require knowledge and capabilities at the 3D object level that are not available. This suggests that what is really needed are an inventory of "smart objects" that have better defined geometries that allow articulated parts that can be individually positioned (e.g. to open a refrigerator door). In some cases this can also include ways of manipulating the size and shape of those parts while keeping the overall object shape coherent. We note that the FaceGen [FaceGen,] faces we experimented with have exactly that property. If the size of the nose is changed the rest of the face will adjust. See section 3.9.

Branching in semantic interpretation: The WordsEye semantic interpretation pipeline could be improved by maintaining multiple branches when there is ambiguity rather than using a greedy algorithm to resolve the most specific rule first. We note that this ambiguity can also occur across modules. For example, syntactically ambiguous input could be passed through as an n-best set to the semantics processing stage.

More vignettes: The vignette definition mechanism can be opened up to let users textually define their own vignettes. In fact, a number of users on the site have asked if they could save the object configurations they have created to be used in another scene. While they can do that now by including the entire original text, it would be conceptually much simpler to give the object configuration a name and invoke it with a single word. This is exactly what vignettes allow and the work we've done already for this is described in Section 5.5.1.2. We note that aspects of this idea, in some sense, date back to SHRDLU [Winograd, 1972] which allowed users to name configurations of blocks in a virtual world. In addition, future work will also include fleshing out action vignettes combined with posed characters and using the larger set of functional affordances.

Merging associational references and metonymy As the system handles more and more higher-level language, merging *associational references* will become more important. If one says *the man walked into the room and sat down on the couch*, then a reasonable assumption is that the couch is part of the room and the room is probably a living room. Likewise for *the man walked into the room and sat down*, we would look for *seat* affordances supplied by the room. We would want to choose a vignette that had seat affordances and then have the man sit in that rather than sitting on the floor. This implies that likely reference merging should affect object selection. Note that we already do this with parts. For example, *the head of the cow is*

blue will force us to pick a 3D cow that has a designated head as 3D part. See Section 6.4.3. The system should also handle visual metonymy, so that *the cereal is on the shelf* puts a box on the shelf rather than the cereal itself.

11.3.2 Future Application Areas

Some other areas of improvement would be at the application level and involve both application-level features combined with internal support for those features.

Virtual reality: Rendering to virtual reality displays combined with a voice input interface could allow users to create or modify virtual environments by describing them. This sort of application could require some new functionality since the distinction between intrinsic versus relative frames of reference would become more important as users immerse *themselves* in the scene. One also can imagine multi-modal techniques to both describe and point to indicate objects and references.

Fly-through and ambient animation: Another mixed-mode product-level feature would be to allow users to animate the scenes they create. A simple type of animation would be fly the camera through the scene. This could be done by setting keyframes in the user interface. A step beyond that would be to allow cuts between scenes to tell a story. A much more difficult task would be to animate the objects *within* a scene. Though even with this, there are half-measures that could be very interesting, for example, giving ambient motion to objects such as wind in trees or human characters moving in place as is often done in computer games.

Storyboards: Another reconfiguration of the system would be to produce storyboards. This could be done purely as a user interface. But a more interesting possibility would be to segment longer amounts of input text into different scenes and produce a "story". Needless to say this would involve tackling difficult issues. But even simple approaches may produce interesting results along the lines of [Joshi *et al.*, 2006] and [Schwarz *et al.*, 2010].

Visual summarization: There is a tremendous amount of free-range text on the internet, some of which could be automatically visualized with WordsEye. This would require the salient parts to be identified and depicted. The text could then be presented with thumbnail image that encapsulate some aspect of the meaning of passages of text.

WordsEye chatbot: Another application possibility is to use chat streams as input to WordsEye and to automatically create scenes (with no user input) from the messages. The WordsEye chatbot could reply with

a scene only when it had high confidence that it produced a good result. We note that chat messages present challenges since they are often ungrammatical and replete with misspellings, slang, emoticons, urls, and special characters. They also often contain text that is not visually oriented or that refers to objects outside of our 3D library. To address this, we could perform text-normalization to filter out "bad" messages and use fallback strategies to depict text that is not directly depictable.

11.4 Coda – How to *Make* Things with Words

Text-to-scene technology is an exciting and truly new way for people to use language and interact with visual content. We believe that in addition to its various applications, it illuminates many fundamental issues in linguistics and natural language understanding that would otherwise be hidden or in the shadows.

We argued earlier that language-generated scenes are a new artistic medium (Section 9.4.3) where language and images are recontextualized and play off each other. We also see a broader set of implications, reaching into other fields, such as cognitive science and philosophy.

In Austin's *How to do things with Words* [Austin, 1962], further elaborated by Searle [Searle, 1969], a number of fundamental performative speech act types are identifed, such as promising, warning, inviting, and ordering. Speech acts operate not only on the *locutionary* level of the utterance and its direct meaning, but also on the *illocutionary* level – the pragmatic effect or force of that utterance. For example, a judge's decree that *the court is adjourned* actually causes the court to be adjourned in addition to its purely propositional content describing a state of affairs.

Likewise, text-to-scene input is a locutionary act of describing what the scene looks like but also has the illocutionary force of causing the scene to come into existence. It is the description that creates what is described.

In the beginning was the word...

Part I

Bibliography

Bibliography

[Adorni *et al.*, 1984] G. Adorni, M. Di Manzo, and F. Giunchiglia. Natural Language Driven Image Generation. In *Proceedings of COLING 1984*, pages 495–500, Stanford, CA, 1984.

[Alam, 2004] Y. S. Alam. Decision trees for sense disambiguation of prepositions: case of over. In *HLT-NAACL Workshop on Computational Lexical Semantics*, Morristown, NJ, 2004.

[Andonova *et al.*, 2010] E. Andonova, T. Tenbrink, and K. R. Coventry. Function and context affect spatial information packaging at multiple levels. In *Psychonomic Bulletin & Review*, pages 575–580, 2010.

[Arnheim, 1969] R. Arnheim. *Visual thinking*. Univ of California Pr, 1969.

[Austin, 1962] John L Austin. Haw to do things with words. *O. Ormson et Ma*, 1962.

[Badler *et al.*, 1998] N. Badler, R. Bindiganavale, J. Bourne, M. Palmer, J. Shi, and W. Schule. A parameterized action representation for virtual human agents. In *Workshop on Embodied Conversational Characters*, Tahoe City, CA, 1998.

[Bailey *et al.*, 1998] D. Bailey, N. Chang, J. Feldman, and S. Narayanan. Extending Embodied Lexical Development. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, Madison, WI, 1998.

[Baker and Ruppenhofer, 2002] Collin F. Baker and Joseph Ruppenhofer. Framenet's frames vs. Levin's verb classes. In *28th Annual Meeting of the Berkeley Linguistics Society*. Berkeley Linguistics Society, 2002.

[Baker *et al.*, 1998] C. Baker, C. Fillmore, and J. Lowe. The Berkeley Framenet Project. In *Proceedings of the 17th international conference on Computational linguistics*, pages 86–90, 1998.

[Banarescu *et al.*, 2012] Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. Abstract meaning representation (amr) 1.0 specification. In *Parsing on Freebase from Question-Answer Pairs." In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. Seattle: ACL*, pages 1533–1544, 2012.

[Bangalore *et al.*, 2009] S. Bangalore, P. Boullier, A. Nasr, O. Rambow, and B. Sagot. Mica: A probabilistic dependency parser based on tree insertion grammars. 2009.

[Baran and Popović, 2007] Ilya Baran and Jovan Popović. Automatic rigging and animation of 3d characters. In *ACM Transactions on Graphics (TOG)*, volume 26, page 72. ACM, 2007.

[Barker and Szpakowicz, 1998] K. Barker and S. Szpakowicz. Semi-automatic recognition of noun modifier relationships. In *17th international conference on Computational linguistics*, Morristown, NJ, 1998.

[Bauer and Rambow, 2011] D. Bauer and O. Rambow. Increasing coverage of syntactic subcategorization patterns in framenet using verbnet. In *Semantic Computing (ICSC), 2011 Fifth IEEE International Conference on*, pages 181–184. IEEE, 2011.

[Bergen and Chang, 2005] Benjamin Bergen and Nancy Chang. Embodied construction grammar in simulation-based language understanding. *Construction grammars: Cognitive grounding and theoretical extensions*, 3:147–190, 2005.

[Berners-Lee *et al.*, 2001] Tim Berners-Lee, James Hendler, Ora Lassila, et al. The semantic web. *Scientific american*, 284(5):28–37, 2001.

[Bitz, 2010] M. Bitz. *When commas meet kryptonite: Classroom lessons from the comic book project*. Teachers College Pr, 2010.

[Blinn, 1988] Jim Blinn. Where am i? what am i looking at?(cinematography). *IEEE Computer Graphics and Applications*, 8(4):76–81, 1988.

[Boberg, 1972] R. Boberg. Generating line drawings from abstract scene descriptions. Master's thesis, Dept. of Elec. Eng, MIT, Cambridge, MA, 1972.

[Bobrow and Winograd, 1977] Daniel G Bobrow and Terry Winograd. An overview of krl, a knowledge representation language. *Cognitive science*, 1(1):3–46, 1977.

[Boggess, 1979] Lois Carolyn Boggess. Computational interpretation english spatial prepositions. Technical report, DTIC Document, 1979.

[Boonthum *et al.*, 2005] C. Boonthum, S. Toida, and I. Levinstein. Sense disambiguation for preposition 'with'. In *Proceedings of the Second ACL-SIGSEM Workshop on the Linguistic Dimensions of Prepositions and their Use in Computational Linguistics Formalisms and Applications*, pages 153–162. Citeseer, 2005.

[Bowerman and Pederson, 1992] Melissa Bowerman and Eric Pederson. Topological relations picture series. In Stephen C. Levinson, editor, *Space stimuli kit 1.2*, page 51, Nijmegen, 1992. Max Planck Institute for Psycholinguistics.

[Brachman *et al.*, 1992] Ronald J Brachman, Hector J Levesque, and Raymond Reiter. *Knowledge representation*. MIT press, 1992.

[Broad, 2008] Neil Broad. *Eastern and Central Arrernte Picture Dictionary*. IAD Press, 2008.

[Buitelaar, 1998] P Buitelaar. CoreLex:An Ontology of Systematic Polysemous Classes. In *Formal Ontology in Information Systems(FOIS 98)*, 1998.

[Chang *et al.*, 2014] Angel X. Chang, Manolis Savva, and Christopher D. Manning. Interactive learning of spatial knowledge for text to 3D scene generation. In *Association for Computational Linguistics (ACL) Workshop on Interactive Language Learning, Visualization, and Interfaces (ILLVI)*, 2014.

[Chang *et al.*, 2015] Angel X. Chang, Will Monroe, Manolis Savva, Christopher Potts, and Christopher D. Manning. Text to 3d scene generation with rich lexical grounding. *CoRR*, abs/1505.06289, 2015.

[Chen *et al.*, 2015] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO captions: Data collection and evaluation server. *CoRR*, abs/1504.00325, 2015.

[Choi and Bowerman, 1991] S. Choi and M. Bowerman. Learning to express motion events in english and korean: The influence of language-specific lexicalization patterns. *Cognition*, 41(1-3):83–121, 1991.

[Clay and Wilhelms, 1996] S. R. Clay and J. Wilhelms. Put: Language-based interactive manipulation of objects. *IEEE Computer Graphics and Applications*, 16(2):31–39, 1996.

[Cohn and Hazarika, 2001] Anthony G. Cohn and Shyamanta M. Hazarika. Qualitative spatial representation and reasoning: An overview. *Fundamenta informaticae*, 46(1-2):1–29, 2001.

[Coyne and Rambow, 2009] Bob Coyne and Owen Rambow. Lexpar: A freely available english paraphrase lexicon automatically extracted from framenet. In *ICSC*, pages 53–58, 2009.

[Coyne and Sproat, 2001] B. Coyne and R. Sproat. WordsEye: An automatic text-to-scene conversion system. In *SIGGRAPH Proceedings of the Annual Conference on Computer Graphics*, pages 487–496, Los Angeles, CA, 2001.

[Coyne *et al.*, 2010a] B. Coyne, O. Rambow, J. Hirschberg, and R. Sproat. Frame Semantics in Text-to-Scene Generation. In *Proceedings of the KES'10 workshop on 3D Visualisation of Natural Language*, Cardiff, Wales, 2010.

[Coyne *et al.*, 2010b] B. Coyne, R. Sproat, and J. Hirschberg. Spatial Relations in Text-to-Scene Conversion. *Computational Models of Spatial Language Interpretation Workshop at Spatial Cognition 2010*, 2010.

[Coyne *et al.*, 2011a] B. Coyne, D. Bauer, and O. Rambow. VigNet: Grounding Language in Graphics using Frame Semantics. In *Workshop on Relational Models of Semantics (RELMS) at ACL*, 2011.

[Coyne *et al.*, 2011b] B. Coyne, C. Schudel, M. Bitz, and J. Hirschberg. Evaluating a Text-to-Scene Generation System as an Aid to Literacy. In *Workshop on Speech and Language Technology in Education (SlaTE) at Interspeech 2011*, 2011.

[Cruse, 1997] D.A. Cruse. *Lexical semantics*. Cambridge University Press, 1997.

[Dale and Reiter, 1995] Robert Dale and Ehud Reiter. Computational interpretations of the Gricean maxims in the generation of referring expressions. *Cognitive science*, 19(2):233–263, 1995.

[De Marneffe *et al.*, 2006] Marie-Catherine De Marneffe, Bill MacCartney, Christopher D Manning, et al. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, volume 6, pages 449–454, 2006.

[Downing, 1977] P. Downing. On the creation and use of english compound nouns. *Language*, pages 810–842, 1977.

[Dras *et al.*, 2012] Mark Dras, François Lareau, Benjamin Börschinger, Robert Dale, Yasaman Motazedi, Owen Rambow, Myfany Turpin, and Morgan Ulinski. Complex predicates in Arrernte. In Miriam Butt and Tracy Holloway King, editors, *Proceedings of the LFG12 Conference*. CSLI Publications, 2012.

[Drucker and Zeltzer, 1995] Steven M Drucker and David Zeltzer. Camdroid: A system for implementing intelligent camera control. In *Proceedings of the 1995 symposium on Interactive 3D graphics*, pages 139–144. ACM, 1995.

[Dupuy *et al.*, 2001] S. Dupuy, A. Egges, V. Legendre, and P. Nugues. Generating a 3D Simulation Of a Car Accident from a Written Description in Natural Language: The CarSim System. In *Proceedings of ACL Workshop on Temporal and Spatial Information Processing*, pages 1–8, Toulouse, France, 2001.

[Ekman, 1992] Paul Ekman. Are there basic emotions? 1992.

[Everingham *et al.*, 2011] M Everingham, L Van Gool, CKI Williams, J Winn, and A Zisserman. The pascal visual object classes challenge results (voc2011). 2011.

[FaceGen,] FaceGen. `http://www.facegen.com`.

[Farhadi *et al.*, 2009] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1778–1785. IEEE, 2009.

[Feiner *et al.*, 1993] Steven Feiner, Blair Macintyre, and Dorée Seligmann. Knowledge-based augmented reality. *Communications of the ACM*, 36(7):53–62, 1993.

[Feist and Gentner, 2003] M. I. Feist and D. Gentner. Factors Involved in the User of In and On. In *Twenty-fifth Annual Meeting of the Cognitive Science Society*, pages 575–580, 2003.

[Fellbaum *et al.*, 2007] C. Fellbaum, A. Osherson, and P. Clark. Putting Semantics into WordNet's "Morphosemantic" Links. In *3rd Language and Technology Conference*, Poznan, Poland, 2007.

[Fellbaum, 1998] C. Fellbaum. *WordNet: an electronic lexical database*. MIT Press, 1998.

[Fillmore, 1968] C. J. Fillmore. The Case for Case. In E. Bach and R. Harms, editors, *Universals in Linguistic Theory*, pages 1–88. Holt, Rinehart, and Winston, New York, 1968.

[Friedell, 1984] Mark Friedell. Automatic synthesis of graphical object descriptions. In *ACM SIGGRAPH Computer Graphics*, volume 18, pages 53–62. ACM, 1984.

[Gardner, 1985] H. Gardner. *Frames of mind: The theory of multiple intelligences*. Basic books, New York, 1985.

[Gee, 2003] J.P. Gee. What video games have to teach us about learning and literacy. *Computers in Entertainment (CIE)*, 1(1):20–20, 2003.

[Gibson, 1977] J. Gibson. The Theory of Affordances. In Shaw, R. and Bransford, J., editor, *Perceiving, acting, and knowing: Toward an ecological psychology*, pages 67–82. Erlbaum, Hillsdale, NJ, 1977.

[Girju *et al.*, 2005] R. Girju, D. Moldovan, M. Tatu, and D. Antohe. On the semantics of noun compounds. *Computer speech & language*, 19(4):479–496, 2005.

[Glass and Bangay, 2008] Kevin Glass and Shaun Bangay. Automating the creation of 3d animation from annotated fiction text. In *IADIS 2008: Proceedings of the International Conference on Computer Graphics and Visualization 2008*, pages 3–10. IADIS Press, 2008.

[Goddard, 2008] Cliff Goddard. Natural semantic metalanguage: The state of the art. *Cross-linguistic semantics*, 102:1–34, 2008.

[Godéreaux *et al.*, 1999] Christophe Godéreaux, Pierre-Olivier El-Guedj, Frédéric Revolta, and Pierre Nugues. Ulysse: An interactive, spoken dialogue interface to navigate in virtual worlds, lexical, syntactic, and semantic issues. *Virtual Worlds on the Internet*, 4:53–70, 1999.

[Halverson, 2010] E.R. Halverson. Film as identity exploration: A multimodal analysis of youthproduced films. *Teachers College Record*, 112(9):2352–2378, 2010.

[Hassani and Lee, 2016] Kaveh Hassani and Won-Sook Lee. Visualizing natural language descriptions: A survey. *ACM Computing Surveys (CSUR)*, 49(1):17, 2016.

[Haugeland, 1995] J. Haugeland. Mind embodied and embedded. In Y. Houng and J. Ho, editors, *Mind and Cognition*. Academia Sinica, Taipei, 1995.

[Havasi *et al.*, 2007] C. Havasi, R. Speer, and J. Alonso. ConceptNet 3: a Flexible, Multilingual Semantic Network for Common Sense Knowledge. In *Proceedings of Recent Advances in Natural Languge Processing*, Borovets, Bulgaria, 2007.

[He *et al.*, 1996] Li-wei He, Michael F Cohen, and David H Salesin. The virtual cinematographer: a paradigm for automatic real-time camera control and directing. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 217–224. ACM, 1996.

[Heine, 1997] Bernd Heine. *Cognitive foundations of grammar*. Oxford University Press, 1997.

[Herskovits, 1986] A. Herskovits. *Language and Spatial Cognition: an Interdisciplinary Study of the Prepositions in English*. Cambridge University Press, Cambridge, England, 1986.

[Hodosh *et al.*, 2013] Micah Hodosh, Peter Young, and Julia Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *J. Artif. Int. Res.*, 47(1):853–899, May 2013.

[Hovy *et al.*, 2006] E. Hovy, M. Marcus, M. Palmer, L. Ramshaw, and R. Weischedel. Ontonotes: the 90% solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers on XX*, pages 57–60. Association for Computational Linguistics, 2006.

[Insider,] Business Insider. `http://www.businessinsider.com/facebook-350-million-photos-each-day-2013-9`.

[Jackendoff, 1985] R.S. Jackendoff. *Semantics and cognition*, volume 8. The MIT Press, 1985.

[Johnston and Busa, 1999] Michael Johnston and Federica Busa. Qualia structure and the compositional interpretation of compounds. In *Breadth and depth of semantic lexicons*, pages 167–187. Springer, 1999.

[Joshi *et al.*, 2006] Dhiraj Joshi, James Z Wang, and Jia Li. The story picturing engine—a system for automatic text illustration. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 2(1):68–89, 2006.

[Journal,] Wall Street Journal. `http://www.wsj.com/articles/microsoft-gains-link-to-a-network-1465922927`.

[Kahn, 1979] K. Kahn. *Creation of Computer Animation from Story Descriptions*. PhD thesis, MIT, AI Lab, Cambridge, MA, 1979.

[Kallmann and Thalmann, 2002] Marcelo Kallmann and Daniel Thalmann. Modeling behaviors of interactive objects for real-time virtual environments. *Journal of Visual Languages & Computing*, 13(2):177–195, 2002.

[Kaplan and Bresnan, 1982] Ronald M. Kaplan and Joan W. Bresnan. Lexical-functional grammar: A formal system for grammatical representation. In J. W. Bresnan, editor, *The Mental Representation of Grammatical Relations*. December 1982.

[Karttunen *et al.*, 1997] Lauri Karttunen, Tamás Gaál, and André Kempe. Xerox finite-state tool. Technical report, Xerox Research Centre Europe, Grenoble, 1997.

[Katz, 2010] Mark Katz. *Capturing sound: how technology has changed music*. Univ of California Press, 2010.

[Kilgarriff and Tugwell, 2001] Adam Kilgarriff and David Tugwell. Word sketch: Extraction and display of significant collocations for lexicography. 2001.

[Kingsbury and Palmer, 2002] Paul Kingsbury and Martha Palmer. From treebank to propbank. In *LREC*. Citeseer, 2002.

[Kipper *et al.*, 2000] K. Kipper, H. T. Dang, and M. Palmer. Class-Based Construction of a Verb Lexicon. In *Proceedings of AAAI 2000*, Austin, TX, 2000.

[Klyne and Carroll, 2006] Graham Klyne and Jeremy J Carroll. Resource description framework (rdf): Concepts and abstract syntax. 2006.

[Kurlander *et al.*, 1996] D. Kurlander, T. Skelly, and D. Salesin. Comic chat. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 225–236. ACM, 1996.

[Lakoff, 1987] G. Lakoff. *Women, fire, and dangerous things*, volume 111. University of Chicago press Chicago, 1987.

[Landau *et al.*, 1981] B. Landau, H. Gleitman, and E. Spelke. Spatial knowledge and geometric representation in a child blind from birth. *Science*, 213(4513):1275, 1981.

[Langacker, 2009] R. Langacker. *Investigations in cognitive grammar*. Mouton de Gruyter, New York, 2009.

[Lareau, 2012] François Lareau. Arrernte footy: A natural language generation project in Arrernte, February 2012.

[Levi, 1978] J. N. Levi. *The syntax and semantics of complex nominals*. Academic Press, New York, 1978.

[Levin, 1993] B. Levin. *English verb classes and alternations: a preliminary investigation*. University Of Chicago Press, 1993.

[Levinson, 1996] S.C. Levinson. Frames of reference and molyneux's question: Crosslinguistic evidence. *Language and space*, pages 109–169, 1996.

[Levinson, 2003] Stephen Levinson. *Space in Language and Cognition: Explorations in Cognitive Diversity*. Cambridge University Press, Cambridge, 2003.

[Litkowski and Hargraves, 2005] K. C. Litkowski and O. Hargraves. The Preposition Project. In *ACL-SIGSEM Workshop on "The Linguistic Dimensions of Prepositions and their Use in Computational Linguistic Formalisms and Applications"*, pages 171–179, University of Essex - Colchester, United Kingdom, 2005.

[Lockwood *et al.*, 2006] Kate Lockwood, Ken Forbus, D Halstead, and Jeffrey Usher. Automatic categorization of spatial prepositions. In *Proceedings of the 28th annual conference of the cognitive science society*, pages 1705–1710, 2006.

[Ma and Mc Kevitt, 2004a] M. Ma and P. Mc Kevitt. Visual semantics and ontology of eventive verbs. *Natural Language Processing Đ IJCNLP*, pages 187–196, 2004.

[Ma and Mc Kevitt, 2004b] Minhua Ma and Paul Mc Kevitt. Visual semantics and ontology of eventive verbs. In *International Conference on Natural Language Processing*, pages 187–196. Springer, 2004.

[Ma and McKevitt, 2006] M. Ma and P. McKevitt. Virtual human animation in natural language visualisation. *Artificial Intelligence Review*, 25:37–53, April 2006.

[Ma, 2006] M. Ma. *Automatic Conversion of Natural Language to 3D Animation*. PhD thesis, University of Ulster, 2006.

[Mahesh *et al.*, 1996] Kavi Mahesh, Stephen Helmreich, and Lori Wilson. *Ontology development for machine translation: Ideology and methodology*. Citeseer, 1996.

[Mandler, 2004] J.M. Mandler. *The foundations of mind: Origins of conceptual thought*. Oxford University Press, USA, 2004.

[Mani and Johnson-Laird, 1982] K. Mani and P. Johnson-Laird. The mental representation of spatial descriptions. In *Memory & Cognition*, pages 181–187, 1982.

[Mani *et al.*, 2008] Inderjeet Mani, Janet Hitzeman, Justin Richer, Dave Harris, Rob Quimby, and Ben Wellner. Spatialml: Annotation scheme, corpora, and tools. In *LREC*, 2008.

[Manning *et al.*, 2014] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014.

[Marr and Nishihara, 1978] David Marr and Herbert Keith Nishihara. Representation and recognition of the spatial organization of three-dimensional shapes. *Proceedings of the Royal Society of London B: Biological Sciences*, 200(1140):269–294, 1978.

[McCloud, 1993] S. McCloud. *Understanding comics: The invisible art*. Harper Paperbacks, 1993.

[Melcuk, 2012] Igor Melcuk. Phraseology in the language, in the dictionary, and in the computer. *Yearbook of Phraseology*, 3(1):31–56, 2012.

[Miller and Johnson-Laird, 1976] G.A. Miller and P.N. Johnson-Laird. *Language and perception.* Belknap Press, 1976.

[Mitchell *et al.*, 2011] Margaret Mitchell, Kees Van Deemter, and Ehud Reiter. Two approaches for generating size modifiers. In *Proceedings of the 13th European Workshop on Natural Language Generation*, pages 63–70. Association for Computational Linguistics, 2011.

[Ng and Cardie, 2002] Vincent Ng and Claire Cardie. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 104–111. Association for Computational Linguistics, 2002.

[Nodelman, 1990] P. Nodelman. *Words about pictures: The narrative art of children's picture books*. Univ of Georgia Pr, 1990.

[Norman, 1988] D. A. Norman. *The Psychology of Everyday Things*. Basic Books, 1988.

[Olivier *et al.*, 1995] P. Olivier, T. Maeda, and J.I. Tsujii. Automatic depiction of spatial descriptions. In *PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE*, pages 1405–1405. JOHN WILEY & SONS LTD, 1995.

[OWL,] OWL. `http://www.w3.org/TR/owl2-overview/`.

[Palmer *et al.*, 2005] Martha Palmer, Daniel Gildea, and Paul Kingsbury. The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, 31(1):71–106, 2005.

[Palmer, 2009] Martha Palmer. Semlink: Linking propbank, verbnet and framenet. In *Proceedings of the generative lexicon conference*, pages 9–15, 2009.

[Pastra, 2008] K. Pastra. PRAXICON: The Development of a Grounding Resource. In *Proceedings of the International Workshop on Human-Computer Conversation*, Bellagio, Italy, 2008.

[Pérez *et al.*, 2006] Jorge Pérez, Marcelo Arenas, and Claudio Gutierrez. Semantics and complexity of sparql. In *International semantic web conference*, pages 30–43. Springer, 2006.

[Peters and Peters, 2000] W. Peters and I. Peters. Lexicalised systematic polysemy in wordnet. In *Proc. Secondt Intnl Conf on Language Resources and Evaluation*, 2000.

[Philpot *et al.*, 2005] A. Philpot, E. Hovy, and P. Pantel. The omega ontology. In *Proceedings of the ONTOLEX Workshop at the International Conference on Natural Language Processing (IJCNLP)*, 2005.

[Piaget, 1999] J. Piaget. *The construction of reality in the child*, volume 20. Psychology Press, 1999.

[Pustejovsky *et al.*, 2003] James Pustejovsky, José M Castano, Robert Ingria, Roser Sauri, Robert J Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir R Radev. Timeml: Robust specification of event and temporal expressions in text. *New directions in question answering*, 3:28–34, 2003.

[Pustejovsky *et al.*, 2011] J. Pustejovsky, J. Moszkowicz, and M. Verhagen. ISO-Space: The Annotation of Spatial Information in Language. In *ISA-6: ACL-ISO International Workshop on Semantic Annotation*, oxford, england, 2011.

[Pustejovsky, 1995] J. Pustejovsky. *The Generative Lexicon*. The MIT Press, Cambridge, MA, 1995.

[Rainie *et al.*, 2012] Lee Rainie, Joanna Brenner, and Kristen Purcell. Photos and videos as social currency online. *Pew Internet & American Life Project*, 2012.

[Rashtchian *et al.*, 2010] Cyrus Rashtchian, Peter Young, Micah Hodosh, and Julia Hockenmaier. Collecting image annotations using amazon's mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, CSLDAMT '10, pages 139–147, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[Richardson *et al.*, 2001] D. C. Richardson, M. J. Spivey, S. Edelman, and A. D. Naples. Language is spatial": Experimental evidence for image schemas of concrete and abstract verbs. In *23rd annual meeting of the cognitive science society*, pages 873–878, Mawhah, NJ, 2001. Erlbaum.

[Rosario *et al.*, 2002] B. Rosario, M. A. Hearst, and C. Fillmore. The descent of hierarchy, and selection in relational semantics. In *40th Annual Meeting on Association for Computational Linguistics*, Morristown, NJ, 2002.

[Rosch, 1973] E.H. Rosch. Natural categories. *Cognitive psychology*, 4(3):328–350, 1973.

[Rouhizadeh *et al.*, 2011a] M. Rouhizadeh, D. Bauer, B. Coyne, O. Rambow, and R. Sproat. Collecting spatial information for locations in a text-to-scene conversion system. *CoSLI*, page 16, 2011.

[Rouhizadeh *et al.*, 2011b] M. Rouhizadeh, B. Coyne, and R. Sproat. Collecting semantic information for locations in the scenario-based lexical knowledge resource of a text-to-scene conversion system. *Knowledge-Based and Intelligent Information and Engineering Systems*, pages 378–387, 2011.

[Russell *et al.*, 2008] B.C. Russell, A. Torralba, K.P. Murphy, and W.T. Freeman. Labelme: a database and web-based tool for image annotation. *International journal of computer vision*, 77(1):157–173, 2008.

[Savva *et al.*,] Manolis Savva, Angel X Chang, Pat Hanrahan, Matthew Fisher, and Matthias Nießner. Pigraphs: Learning interaction snapshots from observations.

[Schank and Abelson, 1977] R. C. Schank and R. Abelson. *Scripts, Plans, Goals, and Understanding*. Earlbaum, Hillsdale, NJ, 1977.

[Schwarcz and Association, 1982] J.H. Schwarcz and American Library Association. *Ways of the illustrator: Visual communication in children's literature*. American Library Association Chicago, 1982.

[Schwarz *et al.*, 2010] Katharina Schwarz, Pavel Rojtberg, Joachim Caspar, Iryna Gurevych, Michael Goesele, and Hendrik PA Lensch. Text-to-video: story illustration from online photo collections. In *Inter-*

national Conference on Knowledge-Based and Intelligent Information and Engineering Systems, pages 402–409. Springer, 2010.

[Searle, 1969] John R Searle. *Speech acts: An essay in the philosophy of language*, volume 626. Cambridge university press, 1969.

[Seversky and Yin, 2006] Lee M. Seversky and Lijun Yin. Real-time automatic 3d scene generation from natural language voice and text descriptions. In *Proceedings of the 14th ACM International Conference on Multimedia*, MM '06, pages 61–64, New York, NY, USA, 2006. ACM.

[Simmons, 1975] R. Simmons. The CLOWNS Microworld. In *Proceedings of the Workshop on Theoretical Issues in Natural Language Processing*, pages 17–19, Cambridge, MA, 1975.

[Singh *et al.*, 2002] P. Singh, T. Lin, E. Mueller, G. Lim, T. Perkins, and W. Li Zhu. Open mind common sense: Knowledge acquisition from the general public. *On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE*, pages 1223–1237, 2002.

[Sipe, 2008] L.R. Sipe. *Storytime: Young children's literary understanding in the classroom*. Teachers College Press, New York, 2008.

[Siskind, 1995] J. M. Siskind. Grounding language in perception. *Artificial Intelligence Review*, 8:371––391, 1995.

[Smith, 1970] Cyril Stanley Smith. Art, technology, and science: Notes on their historical interaction. *Technology and Culture*, 11(4):493–549, 1970.

[Snow *et al.*, 2008] R. Snow, B. O'Connor, D. Jurafsky, and A.Y. Ng. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 254–263. Association for Computational Linguistics, 2008.

[SPARQL,] SPARQL. `http://www.w3.org/TR/rdf-sparql-query/`.

[Spika *et al.*, 2011] Christian Spika, Katharina Schwarz, Holger Dammertz, and Hendrik PA Lensch. Avdt-automatic visualization of descriptive texts. In *VMV*, pages 129–136, 2011.

[Spore,] Spore. Spore. `http://www.spore.com`.

[Sproat and Liberman, 1987] Richard W Sproat and Mark Y Liberman. Toward treating english nominals correctly. In *Proceedings of the 25th annual meeting on Association for Computational Linguistics*, pages 140–146. Association for Computational Linguistics, 1987.

[Suchanek *et al.*, 2007] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago : a large ontology from wikipedia and wordnet. Research Report MPI-I-2007-5-003, Max-Planck-Institut für Informatik, Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany, December 2007.

[Tappan, 1996] D. Tappan. Knowledge-based spatial reasoning for scene generation from text descriptions. In *AIII Proceedings*, pages 1888–1889, 1996.

[Teece, 2010] David J Teece. Business models, business strategy and innovation. *Long range planning*, 43(2):172–194, 2010.

[Tolani *et al.*, 2000] D. Tolani, A. Goswami, and N.I. Badler. Real-time inverse kinematics techniques for anthropomorphic limbs. *Graphical models*, 62(5):353–388, 2000.

[Tutenel *et al.*, 2009] T. Tutenel, R. M. Smelik, R. Bidarra, and K. Jan de Kraker. Using Semantics to Improve the Design of Game Worlds. *Proceedings of the Fifth Artificial Intelligence for Interactive Digital Entertainment Conference*, pages 100–105, 2009.

[Ulinski *et al.*, 2014a] Morgan Ulinski, Anusha Balakrishnan, Daniel Bauer, Bob Coyne, Julia Hirschberg, and Owen Rambow. Documenting endangered languages with the wordseye linguistics tool. In *Proceedings of the 2014 Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 6–14, 2014.

[Ulinski *et al.*, 2014b] Morgan Ulinski, Anusha Balakrishnan, Bob Coyne, Julia Hirschberg, and Owen Rambow. Welt: Using graphics generation in linguistic fieldwork. In *ACL (System Demonstrations)*, pages 49–54, 2014.

[Upstill, 1989] Steve Upstill. *RenderMan Companion: A Programmer's Guide to Realistic Computer Graphics*. Addison-Wesley Longman Publishing Co., Inc., 1989.

[van Deemter, 2016] Kees van Deemter. *Computational Models of Referring: A Study in Cognitive Science*. MIT Press, 2016.

[Vanderwende, 1994] L. Vanderwende. Algorithm for automatic interpretation of noun sequences. In *15th conference on Computational linguistics*, Morristown, NJ, 1994.

[Walsh, 2007] C.S. Walsh. Creativity as capital in the literacy classroom: youth as multimodal designers. *Literacy*, 41(2):79–85, 2007.

[Waltz, 1980] David L Waltz. Understanding scene descriptions as event simulations. In *Proceedings of the 18th annual meeting on Association for Computational Linguistics*, pages 7–11. Association for Computational Linguistics, 1980.

[Warehouse,] Google 3D Warehouse. `http://sketchup.google.com/3dwarehouse/`.

[Whissell, 1989] C. Whissell. The dictionary of affect in language. *Emotion: Theory, research, and experience*, 4:113–131, 1989.

[Wikimedia, a] Wikimedia. `https://commons.wikimedia.org/w/index.php?title=File:Blender_node_screen_242a.jpg&oldid=168654758`.

[Wikimedia, b] Wikimedia. `https://commons.wikimedia.org/w/index.php?title=File:Blender_2.36_Screenshot.jpg&oldid=161158636`.

[Wikipedia, 2016a] Wikipedia. 1% rule (internet culture) — wikipedia, the free encyclopedia, 2016. [Online; accessed 21-August-2016].

[Wikipedia, 2016b] Wikipedia. Amazon mechanical turk — wikipedia, the free encyclopedia, 2016. [Online; accessed 21-August-2016].

[Wikipedia, 2016c] Wikipedia. Apple watch — wikipedia, the free encyclopedia, 2016. [Online; accessed 21-August-2016].

[Wikipedia, 2016d] Wikipedia. British national corpus — wikipedia, the free encyclopedia, 2016. [Online; accessed 21-August-2016].

[Wikipedia, 2016e] Wikipedia. Google glass — wikipedia, the free encyclopedia, 2016. [Online; accessed 21-August-2016].

[Wikipedia, 2016f] Wikipedia. Kinect — wikipedia, the free encyclopedia, 2016. [Online; accessed 21-August-2016].

[Wikipedia, 2016g] Wikipedia. Mad libs — wikipedia, the free encyclopedia, 2016. [Online; accessed 28-November-2016].

[Wikipedia, 2016h] Wikipedia. Metonymy — wikipedia, the free encyclopedia, 2016. [Online; accessed 21-August-2016].

[Wikipedia, 2016i] Wikipedia. Oculus rift — wikipedia, the free encyclopedia, 2016. [Online; accessed 21-August-2016].

[Wikipedia, 2016j] Wikipedia. Physics engine — wikipedia, the free encyclopedia, 2016. [Online; accessed 21-August-2016].

[Wikipedia, 2016k] Wikipedia. Product/market fit — wikipedia, the free encyclopedia, 2016. [Online; accessed 21-August-2016].

[Wikipedia, 2016l] Wikipedia. Siri — wikipedia, the free encyclopedia, 2016. [Online; accessed 21-August-2016].

[Wikipedia, 2016m] Wikipedia. Tag cloud — wikipedia, the free encyclopedia, 2016. [Online; accessed 21-August-2016].

[Wikipedia, 2016n] Wikipedia. Tom clancy's endwar — wikipedia, the free encyclopedia, 2016. [Online; accessed 21-August-2016].

[Wikipedia, 2016o] Wikipedia. Uniform resource identifier — wikipedia, the free encyclopedia, 2016. [Online; accessed 20-December-2016].

[Wilensky, 1986] R. Wilensky. Knowledge representation – a critique and a proposal. *Experience, Memory and Reasoning*, pages 15–28, 1986.

[Wilkins, 1989] David Wilkins. *Mparntwe Arrernte (Aranda): Studies in the structure and semantics of grammar.* PhD thesis, Australian National University, 1989.

[Winograd, 1972] T. Winograd. *Understanding Natural Language*. PhD thesis, Massachusetts Institute of Technology, 1972.

[Yamada *et al.*, 1988] A. Yamada, T. Nishida, and S. Doshita. Figuring out most plausible interpretation from spatial descriptions. In *Proceedings of the 12th conference on Computational linguistics-Volume 2*, pages 764–769. Association for Computational Linguistics, 1988.

[Ye and Baldwin, 2008] P. Ye and T. Baldwin. Towards automatic animated storyboarding. In *Proceedings of the 23rd national conference on Artificial intelligence*, volume 1, pages 578–583, 2008.

[Younger, 1967] Daniel H Younger. Recognition and parsing of context-free languages in time n 3. *Information and control*, 10(2):189–208, 1967.

[Zitnick and Parikh, 2013] C. L. Zitnick and Devi Parikh. Bringing semantics into focus using visual abstraction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

[Zitnick *et al.*, 2013] C. L. Zitnick, Devi Parikh, and Lucy Vanderwende. Learning the visual interpretation of sentences. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2013.

Part II

Appendices

Appendix A

WordsEye vs. Google Evaluation Results

Imaginative Sentence	WordsEye Rating	Google Rating	WordsEye vs Google
The white diamond is 10 feet in front of the hummingbird.	3.33	3.67	(2 1 wordseye)
The bottle cap jar was facing the kiosk	3.0	4.0	(3 0 wordseye)
The bee is in the cherry laurel tree close to the houseboat.	5.0	3.0	(1 2 google)
The three blue balloons were above the pipe organ.	1.33	4.33	(2 1 wordseye)
Five red candy canes are on the water cooler.	1.0	4.67	(3 0 wordseye)
The circular saw and brown bear are near the church.	2.33	4.0	(1 2 google)
The brown shovel is three feet behind the flower.	1.67	3.67	(3 0 wordseye)
The shiny battery is in front of the twelve inch long workbench.	2.33	4.0	(2 1 wordseye)
The mauve flatbed truck is far from the three mugs.	3.0	4.67	(3 0 wordseye)
Five aqua dice are near the roulette table.	5.0	3.0	(0 3 google)
The colt was close to the huge horse trailer.	4.33	3.33	(2 1 wordseye)
The enormous pterodactyl is behind the swivel chair.	1.33	3.67	(3 0 wordseye)
The three notes are under the paper cutter.	1.0	5.0	(3 0 wordseye)
A huge seashell is next to the end table.	1.0	3.33	(3 0 wordseye)
A sea anemone is close to the wheelchair and far from the merry go round.	2.33	4.0	(3 0 wordseye)
One silhouette is in the mirror.	4.67	1.33	(0 3 google)
The red skateboard is three feet under the fence.	1.67	5.0	(3 0 wordseye)
The blue hard hat is in front of the cherry tree.	2.33	4.67	(3 0 wordseye)
The gray lamp is twelve inches behind the stool.	1.0	4.0	(3 0 wordseye)
The purple feline is on top of the judo mat.	1.67	4.0	(3 0 wordseye)
The tiny toaster is 10 feet in front of the bird cage.	4.33	4.0	(3 0 wordseye)
There is a beaver on the beige sports car.	2.33	4.67	(2 1 wordseye)
A small cube is near a gray bucket.	1.67	4.33	(3 0 wordseye)
The chimpanzee near the school bus was only 4 inches tall.	4.33	2.67	(2 1 wordseye)
The chartreuse melon is on top of the small cherry tree.	2.67	3.67	(3 0 wordseye)
The penguin is facing the dish rag booth.	1.67	3.33	(3 0 wordseye)
The sign for the park is near the fax machine.	3.67	3.33	(1 2 google)

Imaginative Sentence	WordsEye Rating	Google Rating	WordsEye vs Google
The red clock is three feet above the desk.	1.33	3.33	(3 0 wordseye)
The supernatural being is on top of one craps table.	1.0	4.33	(3 0 wordseye)
The motor is in the cabin above the gift.	4.67	4.33	(2 1 wordseye)
The two utensils are next to the nutcracker.	1.0	3.67	(2 1 wordseye)
The tiny blue microchip was under the soup bowl.	3.33	3.67	(1 2 google)
The gray brick is two feet under the bowling pin.	1.33	4.33	(3 0 wordseye)
The yellow checker cab is in front of a hydrant.	1.67	2.33	(3 0 wordseye)
Three enormous palm trees are near the green peppers.	3.0	1.33	(1 2 google)
My tiny telephone is three feet from the motorhome.	2.33	4.67	(1 2 google)
Nine enormous jugs are on top of a counter.	1.67	5.0	(3 0 wordseye)
the brick on the bunk bed is big.	1.67	4.67	(3 0 wordseye)
Ten melons are in the small frying pan.	4.67	4.33	(0 3 google)
The fast food restaurant is on top of the grave for the camper.	4.33	5.0	(2 1 wordseye)
The big musical instrument is three feet under the mirror.	1.67	5.0	(3 0 wordseye)
The killer whale and the casserole dish are in the river.	4.67	3.33	(2 1 wordseye)
The clear spray can was facing the huge carpet sweeper.	3.0	3.33	(3 0 wordseye)
The stocking is behind the trash can in the hotdog stand.	2.67	5.0	(3 0 wordseye)
A fireman's hat is facing two graduation caps.	3.0	4.0	(3 0 wordseye)
A maroon coffee cup is three feet behind the alligator.	1.0	4.33	(3 0 wordseye)
A land mammal is on a toilet in an arena.	4.33	4.67	(2 1 wordseye)
The shopping cart was in front of the huge blue door hinge.	2.33	4.67	(3 0 wordseye)
the hare was near the spaceship in the park	3.33	4.0	(3 0 wordseye)
The caliper was on top of the saw in the outhouse.	4.33	4.67	(1 2 google)
The pink pool table is 10 feet behind the ottoman.	1.0	2.67	(3 0 wordseye)
The purple hedgehog is above the box trailer.	1.67	4.0	(2 1 wordseye)
The gray handheld weapon is twelve inches behind the cigarette pack.	2.0	5.0	(3 0 wordseye)
The thin spiral notebook is two feet above the cookware.	1.67	3.33	(3 0 wordseye)
One dime was near the chartreuse panel.	4.33	3.0	(3 0 wordseye)
The seven candlesticks are on top of the gigantic wicker chair.	4.0	3.33	(2 1 wordseye)
A flower is near a housefly in a tennis stadium.	5.0	4.33	(0 3 google)
Above the log cabin are a puma and a pickle.	3.33	5.0	(3 0 wordseye)
There is a buck knife two feet in front of the gray book.	1.33	4.0	(2 1 wordseye)
There were seven pieces of celery on the gray computer printer.	4.0	4.67	(1 2 google)
Two tulip trees are close to a seashell.	3.33	3.0	(3 0 wordseye)
An enormous dinner plate is near four rubber stamps.	1.33	2.67	(2 1 wordseye)
The olive battleship faces the driveway.	4.33	3.33	(2 1 wordseye)
A fat blimp is above the violin.	4.33	3.0	(2 1 wordseye)
The eight tiny circular saw blades are far from the horse chestnut tree.	4.67	3.67	(2 1 wordseye)
The five juveniles are next to the huge helicopter.	3.33	2.67	(3 0 wordseye)
The three grapes are next to the white balcony.	2.0	2.67	(3 0 wordseye)
The red spatula was on top of the riverboat.	4.33	4.33	(0 3 google)
A tiny sculpture is three feet from a diamond.	4.33	3.67	(2 1 wordseye)
Two blackjack tables are in front of the piano bench	1.0	3.67	(3 0 wordseye)

Imaginative Sentence	WordsEye Rating	Google Rating	WordsEye vs Google
The person behind the temple was a baby.	3.33	3.67	(2 1 wordseye)
The book near the oak tree is white.	2.67	4.0	(3 0 wordseye)
Two gray carpets are close to a door hinge.	2.67	4.0	(1 2 google)
The enormous fan is on the dresser.	1.0	4.0	(3 0 wordseye)
The pizza is in the pink lounge chair.	3.0	4.33	(3 0 wordseye)
Nine lutes are under the yellow fireman's hat.	2.33	3.0	(2 1 wordseye)
A shiny, 4 inch tall penny is next to a bullet.	1.67	2.0	(3 0 wordseye)
The yellow watch was near the four notebooks	1.67	4.0	(2 1 wordseye)
Five pears are under the martini glass.	1.0	4.0	(3 0 wordseye)
The torched newspaper vending machine is facing the blacksmith shop.	3.67	4.67	(3 0 wordseye)
Eight huge coffee pots are facing the conduit.	1.67	4.0	(3 0 wordseye)
The fetus is against the display screen in the school building.	4.33	4.33	(3 0 wordseye)
There is one razorblade close to the toilet.	2.0	4.33	(2 1 wordseye)
The chartreuse glass is on the huge footstool.	1.33	1.67	(2 1 wordseye)
The lollipop was 10 feet from the huge answering machine.	1.33	4.67	(3 0 wordseye)
The fat cigar is on the mauve counter in front of the spice bottle.	4.0	3.33	(1 2 google)
The gigantic candy bar is in the sailboat.	2.33	4.33	(2 1 wordseye)
The arctic fox is on top of the telephone, in the bedroom.	5.0	4.67	(2 1 wordseye)
The gray cat is behind the ping pong table.	1.0	2.0	(2 1 wordseye)
The opaque chessboard is on top of the three clothes irons.	1.0	4.0	(3 0 wordseye)
The seabird is on top of the duffel bag in the coliseum.	5.0	4.0	(2 1 wordseye)
The huge bunny is on top of the grill.	1.33	1.67	(3 0 wordseye)
There is a camcorder on top of the towel in the shower stall.	5.0	3.33	(2 1 wordseye)
The platypus is in the backpack, which is near the water tower.	3.0	4.33	(1 2 google)
The green computer monitor is facing the plant stand	1.33	5.0	(3 0 wordseye)
The bowl is in front of the shiny counter.	1.67	3.0	(3 0 wordseye)
In front of the palace's oven is a napkin.	4.67	4.0	(1 2 google)
The big life preservers are on the twelve aircraft.	4.33	3.67	(1 2 google)
The two campers were next to the black unicycle.	3.0	4.67	(2 1 wordseye)
Five magenta cantaloupes are in the trash can.	1.67	4.33	(2 1 wordseye)
The tiny tangerine is near five hamburgers.	1.67	3.67	(3 0 wordseye)
The bench is far from the large bathroom sink.	1.0	3.67	(3 0 wordseye)
The sled is on the yellow snowball.	2.67	4.0	(3 0 wordseye)
Five tiny baseball caps are next to the drawing.	3.0	5.0	(2 1 wordseye)
The small apple pie was far from the piano bench.	2.33	3.0	(3 0 wordseye)
A red mouse is 10 feet under the streetlight.	3.0	5.0	(2 1 wordseye)
The tiny blue sailing ship is above the crib.	1.0	4.67	(3 0 wordseye)
The elephant is behind the grave by the building.	1.67	3.67	(3 0 wordseye)
The sycamore tree has nine duffel bags on top of it.	4.0	4.33	(1 2 google)
There are five artichokes above the kitchen counter.	1.67	4.0	(3 0 wordseye)
The hotdog is next to the chocolate cake in the booth.	2.33	4.0	(3 0 wordseye)
Five meters behind a tiny crate is a squeegee.	3.67	3.0	(2 1 wordseye)
The blowtorch was behind the large kitchen island	1.0	4.33	(3 0 wordseye)

Imaginative Sentence	WordsEye Rating	Google Rating	WordsEye vs Google
Six gigantic utensils were near the van.	2.33	5.0	(3 0 wordseye)
The large prawn is on top of the stool.	1.33	3.0	(3 0 wordseye)
The huge jewel is in front of the red rolling pin.	1.0	3.0	(3 0 wordseye)
The pink butterfly is behind the bicycle.	2.33	3.67	(3 0 wordseye)
The brown nail polish is behind the 4 inch bar.	3.67	4.33	(3 0 wordseye)
The electric outlet is close to the living room chair in the church.	4.67	4.67	(1 2 google)
On top of the office bunk bed was a possum.	1.67	3.33	(2 1 wordseye)
The black bucket is in front of the couch.	1.33	2.67	(3 0 wordseye)
The black handgun is on top of the motorboat.	3.33	4.0	(3 0 wordseye)
The hedge trimmer was next to the gigantic fir tree.	5.0	2.0	(0 3 google)
A tropical fish is close to a cigar in a temple.	5.0	4.33	(2 1 wordseye)
Four root vegetables are close to the gray organ.	3.0	5.0	(3 0 wordseye)
The egg biscuit sandwich is under the teacup on the popcorn stand.	2.33	4.33	(2 1 wordseye)
There is a red vending machine facing the alligator.	1.0	4.0	(3 0 wordseye)
The snail is facing the fire hydrant in the coliseum.	4.67	4.33	(3 0 wordseye)
The test tube is 4 inches away from the crescent moon symbol, which is 8 feet wide.	1.67	4.0	(3 0 wordseye)
There was a popsicle next to the hydrocraft on the beach.	3.0	4.67	(3 0 wordseye)
The surfboard is near the green scaffolding.	2.33	2.33	(3 0 wordseye)
The car is under six vending machines.	4.67	5.0	(1 2 google)
The huge percussion instrument is close to the six hourglasses.	1.0	4.67	(3 0 wordseye)
The swine is on top of the cigar near the schoolhouse.	3.0	4.67	(3 0 wordseye)
The little paper fan and the scarlet baggage tag were close to the tea set.	2.67	4.0	(3 0 wordseye)
A magenta cat is next to a houseboat.	1.33	4.0	(3 0 wordseye)
The thin chocolate eclair is in twelve inches of sheet	5.0	3.33	(0 3 google)
The gigantic beech tree is 10 feet away from the birch tree.	1.67	3.33	(2 1 wordseye)
A maroon feather is three feet above a melon.	4.0	5.0	(2 1 wordseye)
The spatula was close to eight eggs.	1.0	1.67	(3 0 wordseye)
Two meters away from the pumper truck is a magenta apple.	2.33	5.0	(3 0 wordseye)
The pen and the melon are far from the donut shop.	4.0	3.67	(2 1 wordseye)
The tiny 4 inch cookie is on top of the pew.	1.67	5.0	(3 0 wordseye)
Eleven blue hooks are close to the weathervane.	4.0	4.33	(3 0 wordseye)
The faucet facing the wreath is red.	3.0	2.67	(3 0 wordseye)
The black chicken is in front of the picket fence.	1.0	3.0	(3 0 wordseye)
The black backpack is far from the end table.	1.67	4.0	(3 0 wordseye)
The orange car was close to the flask.	1.33	3.33	(2 1 wordseye)
A fat cassette is on top of the trailer.	1.33	5.0	(3 0 wordseye)
The big asparagus is next to the bicycle.	2.67	4.67	(3 0 wordseye)
The fat, blue glue bottle was next to the holly.	3.0	4.33	(3 0 wordseye)
The black penguin is near the floor safety sign.	3.0	4.0	(3 0 wordseye)
There is a transparent screen twelve inches behind the soap bar.	2.0	4.33	(3 0 wordseye)
The weather vane is next to the laurel tree at the factory.	2.67	3.33	(3 0 wordseye)
A small mouse was above the tan spatula.	1.0	4.67	(3 0 wordseye)

Imaginative Sentence	WordsEye Rating	Google Rating	WordsEye vs Google
The fat dalmatian is behind the couch.	2.0	3.33	(3 0 wordseye)
A cyan cat is 4 inches above the panel.	2.33	4.0	(3 0 wordseye)
On the couch is a gray graduation cap.	1.0	3.0	(3 0 wordseye)
The gray camper is near the wolf	2.0	4.33	(3 0 wordseye)
The shiny coffee pot is behind eleven staplers.	1.0	4.0	(3 0 wordseye)
The harmonica is behind the enormous pink tricycle.	4.0	4.0	(2 1 wordseye)
The thin tire was close to the blue fence.	2.0	5.0	(3 0 wordseye)
The gray beluga whale is far from the motorboat.	1.33	2.0	(3 0 wordseye)
Ten sombreros are behind a pitcher.	5.0	4.0	(1 2 google)
A gigantic beast is under the air hockey table.	4.0	4.0	(2 1 wordseye)
The polar bear was in the castle in front of the paper.	4.0	3.0	(0 3 google)
There is a motor in the big steamroller.	4.0	1.67	(1 2 google)
The gray tricycle is 100 feet in front of the cabinet.	3.67	3.33	(1 2 google)
Nine tangerines are above the mailbox.	1.0	4.0	(3 0 wordseye)
A green utensil is near the living room chair.	1.33	3.0	(3 0 wordseye)
The lock was near the anvil at the dam.	4.0	4.0	(2 1 wordseye)
A thin spiked collar is near four ashtrays.	5.0	5.0	(1 2 google)
The hamburger is in front of the chess board in the hall.	4.0	3.33	(1 2 google)
A whale is far from a wicker chair on the beach.	2.67	4.33	(3 0 wordseye)
The green envelope is on top of the park bench.	1.0	3.67	(3 0 wordseye)
There are six shiny oranges next to the ball.	1.0	4.0	(3 0 wordseye)
The blue motorboat was close to two meters from the hemlock tree.	1.33	3.67	(3 0 wordseye)
The scarlet stool is in the battleship.	4.33	4.67	(3 0 wordseye)
Two bagels are in front of cutlery.	2.67	4.33	(3 0 wordseye)
The tiny blowtorch is two feet away from the controller.	2.33	3.0	(2 1 wordseye)
The thin, mauve propeller hat is next to the cash register.	1.33	4.33	(3 0 wordseye)
The device is 4 inches in front of the red buck knife.	3.0	4.33	(3 0 wordseye)
The tiny blue marble cherub was far from the breakfast roll.	2.0	4.33	(3 0 wordseye)
The brown cantaloupe is next to the high chair.	3.33	1.67	(3 0 wordseye)
The rowboat is on the white cake.	2.33	2.0	(3 0 wordseye)
Nine big olives are facing the roll of film.	1.33	3.0	(3 0 wordseye)
The gray dolphin is behind the pontoon boat.	1.0	3.33	(2 1 wordseye)
Ten cyan vessels are in front of a torch.	5.0	5.0	(3 0 wordseye)
The riverboat is close to the seven beech trees.	1.0	4.67	(3 0 wordseye)
There is a blue cellphone on the dresser.	1.0	5.0	(3 0 wordseye)
Ten tomatoes are against the dresser.	1.33	4.0	(2 1 wordseye)
The hazel handle is 20 inches in front of the wagon.	2.67	2.33	(3 0 wordseye)
The beige pipe is in front of the roulette table.	2.67	3.33	(3 0 wordseye)
There is one pink cellphone on the swivel chair.	1.0	4.67	(3 0 wordseye)
A wide wine glass is on a black desk tray.	1.0	4.33	(3 0 wordseye)
A gigantic brown pony is under the birch tree.	2.33	4.0	(2 1 wordseye)
The device is under the brown pool table.	2.0	2.0	(3 0 wordseye)
The glass streetlight close to the sign is enormous.	4.33	2.0	(0 3 google)

Imaginative Sentence	WordsEye Rating	Google Rating	WordsEye vs Google
A huge hex nut is against the transparent glass.	1.33	4.67	(3 0 wordseye)
Six tiny water drops were on top of the end table.	3.67	4.67	(2 1 wordseye)
The beige water bottle is two feet above the nightstand.	1.67	4.67	(3 0 wordseye)
The vase in the pink sleigh.	1.33	3.0	(2 1 wordseye)
The purple megaphone is against the sycamore tree.	3.0	3.67	(3 0 wordseye)
The magenta mouse is far from the swivel chair.	2.0	3.33	(3 0 wordseye)
The fuchsia handcuffs are near the bathroom sink.	2.33	2.33	(2 1 wordseye)
The chartreuse sombrero was on top of one counter.	1.67	2.67	(2 1 wordseye)
The big spider web is two feet above the bongo drum.	2.33	3.67	(2 1 wordseye)
Seven motorhomes are near the crimson shovel.	1.67	4.67	(3 0 wordseye)
The huge human was 10 feet away from the toothbrush.	3.33	4.67	(3 0 wordseye)

Realistic Sentence	WordsEye Rating	Google Rating	WordsEye vs Google
A kitchen area	4.33	1.0	(0 3 google)
A black cat on a pillow and a grey dog on a sofa.	1.33	3.33	(3 0 wordseye)
Close up of a seagull and other seagulls in the background.	2.0	3.33	(0 3 google)
A colorful train	2.33	1.33	(0 3 google)
A black dog stands in the middle of a construction site.	4.67	4.67	(0 3 google)
The back of the black SUV is wrecked.	3.67	3.33	(1 2 google)
A black dog in a grass field.	3.33	1.0	(0 3 google)
A yellow truck near a lifeguard station on a beach.	1.33	1.0	(0 3 google)
A white cat stands on the floor.	1.33	3.0	(3 0 wordseye)
The front end of an airplane at an airport.	5.0	2.0	(0 3 google)
A small tree and a plant inside a pot.	2.33	1.33	(0 3 google)
A hummingbird in front of a flower	1.67	1.0	(0 3 google)
A green and grey bird is sitting in a tree.	2.0	1.67	(1 2 google)
Two women in a sailboat.	3.0	1.0	(0 3 google)
Canoe on the beach close to the sea.	1.0	1.33	(0 3 google)
A small horse runs on the grass.	2.67	1.67	(0 3 google)
A woman with a bird on her head and a bird on her shoulder.	4.67	1.0	(0 3 google)
A brown horse stands in tall grass.	3.67	1.0	(0 3 google)
A sheep beside a road and a lake.	1.67	2.67	(1 2 google)
The front end of a yellow school bus.	1.0	1.0	(0 3 google)
The dog is looking at the sheep.	3.0	3.67	(0 3 google)
An empty room with glass windows and a chandelier.	2.0	1.33	(0 3 google)
Two women and a black dog are near a sofa and chair.	2.0	3.67	(3 0 wordseye)
Young man with glasses sitting on red couch.	4.0	1.0	(0 3 google)
A man with a top hat on a white horse.	2.67	1.0	(0 3 google)
A party room.	5.0	1.0	(0 3 google)
Wood table with four chairs.	1.67	1.0	(0 3 google)
An airplane facing the camera.	3.67	1.0	(1 2 google)
Brown and white sleeping dog.	4.0	1.0	(0 3 google)

Realistic Sentence	WordsEye Rating	Google Rating	WordsEye vs Google
A field with many black cows in it.	3.33	1.33	(0 3 google)
A black table with white fuzzy chair.	3.33	3.67	(2 1 wordseye)
Red boat on a river	1.0	1.67	(0 3 google)
Black motorcycle on bike path in forest.	2.33	2.33	(1 2 google)
A city skyline and river	1.0	1.33	(0 3 google)
A girl riding a tan horse.	1.0	1.33	(0 3 google)
A black dog	1.0	1.0	(0 3 google)
an empty glass vase and a colorful vase with paper flowers behind it made from a bottle	2.0	3.0	(2 1 wordseye)
A single engine plane is sitting on the runway.	1.67	1.0	(0 3 google)
A mountain goat is on snowy rocks near trees.	1.67	1.67	(0 3 google)
A group of children jump on the beach.	3.33	1.67	(0 3 google)
Black woman with flower hat sitting next to dog with the same hat.	3.67	2.0	(0 3 google)
A sheep sits near a metal fence.	2.0	1.67	(1 2 google)
A red train	1.0	1.0	(0 3 google)
A brown dog lying on the grass.	2.0	1.0	(0 3 google)
A train is on the railroad tracks	2.33	1.0	(0 3 google)
A man with a bicycle at a coffee house.	4.0	2.67	(0 3 google)
A grey cat lying on a wooden table.	1.0	1.0	(1 2 google)
A steam engine train.	2.0	1.0	(0 3 google)
A room with pink walls, a kitchen table, pink couch, and other furniture.	3.0	2.67	(1 2 google)
The lamb is looking at the camera.	1.33	1.0	(0 3 google)
A grey house with a red door.	2.33	1.33	(1 2 google)
A bicycle in a dining room.	2.0	1.0	(0 3 google)
A brown horse in a green field.	2.67	1.0	(0 3 google)
The blue train is at the station.	2.33	1.0	(0 3 google)
A laptop computer and a computer monitor.	1.0	1.0	(1 2 google)
A long train is on train tracks.	2.0	1.67	(0 3 google)
A man on a black motorcycle.	3.0	1.0	(0 3 google)
A child sits in a large black leather chair.	2.0	1.67	(0 3 google)
The big white boat is in the ocean.	1.67	1.0	(1 2 google)
Two cows standing in a large field.	1.0	2.0	(0 3 google)
An indoor plant in a yellow pot by a window.	2.0	1.67	(0 3 google)
A white sheep in a field of grass.	3.0	1.0	(0 3 google)
Two people with a bike standing under a sign.	2.67	3.0	(1 2 google)
A messy car	4.67	1.33	(0 3 google)
A brown cat sits by a window.	1.67	2.33	(0 3 google)
Four people are behind a table.	1.33	1.33	(3 0 wordseye)
Three bottles are on a table.	1.0	2.33	(3 0 wordseye)
a silver TV with a silver stand	1.0	1.67	(0 3 google)
A television screen over a room with two desks and a bed.	3.33	2.67	(0 3 google)
Brown dog with head on pillow on white sofa.	3.0	4.0	(2 1 wordseye)
Eight people at a table at a restaurant.	3.33	4.33	(2 1 wordseye)

Realistic Sentence	WordsEye Rating	Google Rating	WordsEye vs Google
Train in a station.	2.0	1.0	(0 3 google)
An adult rides a child's bike.	3.0	1.0	(0 3 google)
a desk with a computer in a room	3.67	1.33	(0 3 google)
An old fashioned passenger bus with open windows	1.0	2.0	(0 3 google)
Two black and white dogs at the bottom of stairs.	2.0	3.33	(2 1 wordseye)
a white sheep on the grass in front of trees	2.0	1.0	(1 2 google)
There is a small desk and chair in front of the laundry room.	2.67	2.0	(0 3 google)
The Blue Angels flying over a river with a skyline in the background.	3.67	1.67	(0 3 google)
Young woman in glasses sitting.	3.67	1.0	(0 3 google)
Aircraft carrier on the open ocean.	1.0	1.0	(0 3 google)
A dark brown cow with her calf.	1.67	1.33	(1 2 google)
A cat is sitting in between two bikes.	3.0	3.33	(3 0 wordseye)
A red double-decker bus drives toward the camera on a busy street.	3.0	1.67	(0 3 google)
A brown leather sofa with a wooden coffee table in front of it.	3.0	1.33	(0 3 google)
A small motorboat is in the shallow water.	2.67	1.0	(0 3 google)
a yellow school bus	1.0	1.0	(1 2 google)
Cream furniture and a television in a living room with palm tree decorations.	2.67	2.0	(0 3 google)
Table with blue table cloth in kitchen.	2.0	1.0	(0 3 google)
A table with bowls and plants.	3.67	1.67	(0 3 google)
a girl with glasses and a brown cow	1.67	4.67	(3 0 wordseye)
The brown train is sitting on the railroad tracks.	1.0	1.0	(1 2 google)
The black bird is sitting on the ground.	4.33	1.0	(0 3 google)
a sail boat with two people	2.0	1.0	(0 3 google)
Art items on a table.	2.33	1.67	(0 3 google)
The big red bus's hood is open.	4.0	1.33	(0 3 google)
Three sheep in middle of gravel road.	1.33	3.0	(2 1 wordseye)
A white lamb with its nose against a wooden box.	4.0	2.0	(0 3 google)
a laptop on a desk and a chair	1.33	1.67	(3 0 wordseye)
The cat is sitting on a bag of cat food.	3.33	4.67	(3 0 wordseye)
A sailboat at sea	2.33	1.0	(0 3 google)
The black and white cat is looking up at the camera.	4.0	1.0	(0 3 google)
A firehouse with firetruck out front	2.0	1.33	(0 3 google)
A small street in an urban area.	3.33	1.33	(0 3 google)
Three black cows and one brown cow stand in a snowy field.	1.0	2.67	(2 1 wordseye)
A man is riding his motorcycle.	1.67	1.0	(0 3 google)
A large green bus on the street.	1.0	1.67	(0 3 google)
The dining room table is ready for a meal.	3.33	1.0	(0 3 google)
Two men in a small wooden canoe on the water.	3.0	1.67	(0 3 google)
A gray and white cat sits on a table.	2.0	2.0	(0 3 google)
Wet outdoor furniture on a stone patio.	3.0	2.0	(0 3 google)
A red, double-decker bus.	4.33	1.0	(0 3 google)
A passenger jet at an airport.	5.0	1.67	(0 3 google)

Realistic Sentence	WordsEye Rating	Google Rating	WordsEye vs Google
A group of chickens walking along a dirt track.	2.67	4.67	(2 1 wordseye)
A small white dog looks down the street while a man on a bench watches the dog.	2.0	3.0	(1 2 google)
A room with a couch, television, bookcases, and various wall decorations.	2.33	1.0	(0 3 google)
A sheep dog and two sheep walking in a field.	2.0	4.67	(2 1 wordseye)
A cow in a grassy area.	1.33	1.0	(0 3 google)
A cat sits on top of a wooden railing as a large black dog looks up at it.	1.67	3.67	(2 1 wordseye)
Two kids sitting in folding chairs.	3.0	4.0	(2 1 wordseye)
A train car at a large group of railroad tracks	2.0	2.67	(0 3 google)
Two women sitting on brown couch.	4.0	1.33	(0 3 google)
A man and woman with helmets on brown horses at a farm.	3.0	2.67	(1 2 google)
A sculpture of a dolphin inside a building.	4.0	3.33	(0 3 google)
A pavilion with a man walking across it.	3.67	1.33	(0 3 google)
Cattle in building.	1.67	1.67	(1 2 google)
Plane on the ground	1.0	1.33	(0 3 google)
A wide screen TV sits in a living room.	1.33	1.33	(0 3 google)
A young cow in the field.	1.67	1.67	(0 3 google)
A big white ship is docked.	3.33	1.33	(0 3 google)
Modern room with TV in a round brown frame.	4.33	1.33	(0 3 google)
A cubicle with a large, black office chair and a telephone on the desk.	3.33	3.67	(2 1 wordseye)
Man and boy walking on beach.	2.67	1.0	(0 3 google)
A blue party bus	2.0	1.33	(0 3 google)
A large jet on the ground at the airport.	5.0	1.0	(0 3 google)
a yellow school bus in front of a house	1.0	1.0	(1 2 google)
Three goats are being rounded up by a dog.	2.33	2.33	(3 0 wordseye)
Three girls sit on racing horses.	4.67	4.33	(1 2 google)
A room with cartoon pillows on a couch.	3.0	1.33	(0 3 google)
A blue sofa in a modern room.	1.33	1.33	(0 3 google)
Several people standing beneath a large gray plane.	1.0	1.0	(1 2 google)
Dog on a flower print pillow	1.0	1.0	(0 3 google)
A yellow school bus is on a road near grass and trees.	1.33	1.0	(0 3 google)
The red bus is on the street with two other cars.	1.0	2.0	(2 1 wordseye)
A dog on a bed with a woman in it.	1.0	1.0	(0 3 google)
A television in a messy room with several bookcases.	3.33	2.67	(0 3 google)
A pink bicycle is in front of a building	2.0	2.67	(3 0 wordseye)
Front of vehicle image through a back vehicle window.	5.0	2.33	(0 3 google)
Various animals in a field with trees.	3.0	1.0	(0 3 google)
Five people sitting on one couch.	2.33	1.0	(0 3 google)
A yellow car is painted like a cartoon mouse.	3.33	3.0	(0 3 google)
A clear plastic chair.	1.0	1.0	(0 3 google)
A baby is sitting in the grass.	1.0	1.33	(0 3 google)
An interior room with a shelf of books and decorative items below a loft area.	3.67	1.0	(0 3 google)

Realistic Sentence	WordsEye Rating	Google Rating	WordsEye vs Google
A black dog is chasing an object.	3.67	2.67	(1 2 google)
A lamb stands near two tiny animals.	2.0	2.67	(1 2 google)
A man is standing next to a yellow sports car.	1.0	1.33	(1 2 google)
A man with glasses in a cubicle.	2.67	1.33	(0 3 google)
Two men watching the oncoming train.	2.67	4.0	(3 0 wordseye)
A woman with her child.	1.67	1.0	(0 3 google)
Two cats are looking at a window.	3.33	1.0	(0 3 google)
A cat sitting on the ground looks out through a clear door screen.	3.33	1.33	(0 3 google)
Birds fly low over water.	3.0	1.0	(0 3 google)
Two girls sitting at the breakfast table	3.67	1.67	(0 3 google)
Two goats near a pond.	1.33	1.0	(1 2 google)
A blue train in a mountainous area.	3.33	2.67	(0 3 google)
Red chair in living room with wooden floor.	2.0	1.33	(0 3 google)
Trains inside an indoor station	4.0	1.0	(0 3 google)
A large orange and black locomotive.	1.33	1.0	(0 3 google)
A Canadian train with the flag of Canada on the tracks.	1.67	1.33	(3 0 wordseye)
Group of teenagers being silly.	3.67	1.0	(0 3 google)
two horses	1.0	1.0	(0 3 google)
Several unusual monuments.	2.0	2.0	(1 2 google)
A white kid lying on the ground.	1.0	1.0	(0 3 google)
A large passenger plane on a landing strip.	2.0	2.0	(0 3 google)
Airplane on runway in front of buildings.	4.67	1.33	(0 3 google)
A water buffalo.	1.0	1.33	(0 3 google)
A man stands in front of a train.	1.0	1.0	(1 2 google)
A girl is seated on a donkey and an older woman stands to her right.	4.33	3.0	(0 3 google)
A white plane on the runway	1.0	1.0	(1 2 google)
The black and brown cow is looking at the camera.	2.0	2.67	(0 3 google)
A flat screen TV on the floor.	2.0	1.0	(0 3 google)
a close up head of an ostrich	2.67	1.0	(0 3 google)
A red bird and four other birds sitting in the snow.	1.67	3.67	(3 0 wordseye)
Two small black dogs on a furry white rug.	2.33	4.0	(3 0 wordseye)
A patio with a chair is beside a tool shed.	1.67	4.67	(2 1 wordseye)
A small watercraft at the edge of a lake.	1.67	2.0	(0 3 google)
A young horse trots.	2.33	1.33	(0 3 google)
A large ram at the top of a hill.	2.67	2.0	(0 3 google)
An ornate wooden chair.	2.67	1.0	(1 2 google)
A brown horse and brown pony in a forest.	1.67	3.67	(3 0 wordseye)
A small tiled kitchen area.	4.0	1.0	(0 3 google)
The face of a gray ram.	1.67	1.33	(2 1 wordseye)
A brown sheep and small black animal inside a pen.	3.0	2.67	(0 3 google)
Man with laptop computer sitting on a couch in a small room.	4.0	1.0	(0 3 google)
An empty bus station.	2.33	1.0	(1 2 google)
people, a bus and some cars in the street	2.33	1.67	(0 3 google)

Realistic Sentence	WordsEye Rating	Google Rating	WordsEye vs Google
An old photo of a gentleman near a bus.	4.33	1.33	(0 3 google)
A group of sheep in a field.	4.33	1.0	(0 3 google)
A brown dachshund beside the door.	1.33	3.0	(2 1 wordseye)
A yellow room with a table and chairs.	2.0	1.0	(0 3 google)
A trail car on the tracks at a station.	3.0	1.0	(0 3 google)
Two men and three women in a restaurant.	4.0	1.33	(0 3 google)
Two green and white trains are sitting on the train track.	3.0	4.67	(2 1 wordseye)
A boat is on the water near a small plane.	1.0	1.67	(3 0 wordseye)
Two cats are seated on a bed.	4.67	1.33	(0 3 google)
Woman with infant, and child sitting next to her on couch.	4.33	1.33	(0 3 google)
A dog stands near a person and an open refrigerator door.	3.0	3.67	(0 3 google)
A wooden ship sailing.	1.33	1.33	(0 3 google)
A woman and two children are looking at a piece of paper.	2.0	2.33	(0 3 google)
A train is railing between a dead end street and a stand of evergreens.	4.33	3.0	(2 1 wordseye)
People walk along a sidewalk near a red double-decker bus.	3.0	1.0	(0 3 google)
A adult and two young sheep in a field	2.67	1.67	(0 3 google)
The silver car looks very expensive.	2.33	1.0	(0 3 google)
Boats on the ocean with a hill behind it.	3.33	1.0	(0 3 google)
A two-seater bike is chained to a tree.	4.33	3.0	(0 3 google)
A raft with 3 adults and two children in a river.	4.67	1.33	(0 3 google)
A Thanksgiving meal with white daisies on a small table.	5.0	3.0	(0 3 google)
A computer and monitor on a desk	1.0	1.0	(0 3 google)
The wet cat is drinking water.	4.33	2.33	(0 3 google)
Two Indian women with two Indian girls near the water.	3.0	2.67	(2 1 wordseye)
A girl with a helmet on a bicycle near a river.	2.33	1.33	(0 3 google)
A young brown colt runs in the field near a white fence.	3.33	2.67	(0 3 google)
A large white bird is standing near the water.	1.67	1.0	(0 3 google)
A sheep stands alone in an empty bus stop.	2.33	3.33	(1 2 google)
Two men standing on beach.	2.0	1.33	(0 3 google)
A fighter jet with missiles on a runway.	2.33	1.33	(0 3 google)
A white bicycle stands on a floor.	1.0	5.0	(3 0 wordseye)
A black and white cat is sleeping next to a plant in the window.	3.33	2.67	(1 2 google)
Two black and white dogs behind chain link fence.	2.33	3.0	(1 2 google)
a sheep in the mountains	3.33	2.0	(0 3 google)
A black-and-white photo of a loveseat.	4.33	2.33	(0 3 google)
A brown duck and white duck stand on the grass.	3.0	3.33	(3 0 wordseye)
A white crane	1.0	1.0	(0 3 google)
A few people and a dog are inside a lean-to type structure.	4.67	3.33	(1 2 google)
Buildings and trees are at the base of a mountain.	1.67	1.0	(0 3 google)
A yellow motorcycle	1.0	1.33	(0 3 google)
Two black and white cows behind a metal gate against a partly cloudy blue sky.	1.67	4.67	(3 0 wordseye)
A man riding a small bicycle.	1.0	1.0	(0 3 google)

Realistic Sentence	WordsEye Rating	Google Rating	WordsEye vs Google
The passenger plane is sitting at the airport.	5.0	1.67	(0 3 google)
A large hand statue outside of a country store.	4.0	3.33	(0 3 google)
A tan dog on the cream carpet.	1.33	4.0	(3 0 wordseye)
a person on a canoe on a lake	1.33	1.0	(0 3 google)
A crowded double-decker bus station in the city	4.67	2.67	(0 3 google)
four people are on a boat on green water	2.33	3.0	(1 2 google)
A furry dog lying on a glass table.	1.0	2.67	(3 0 wordseye)
Animals stand near plants and a wire fence.	3.33	3.33	(0 3 google)
A green glass jar.	1.0	1.67	(0 3 google)
Three men in suits sitting at a table.	3.67	1.0	(0 3 google)
a small dog running	3.67	1.0	(0 3 google)

Appendix B

HEAF Pilot Curriculum

Session 1: Introduction of the Platform

a. Introduction:

- Give students the same sentence starter: The dog is on the _____ .
- Change image selection of dog.
- Change color of dog.
- Change size of dog with descriptors (large, huge, tiny) and numbers (10 feet tall).
- Change color of the sky.
- Change texture of the ground.
- Add something on top of the dog.
- Add something below the dog.
- Render the final image.
- View scene in My Portfolio and the Gallery

b. Scene re-creation:

- Give students the following scene and see who can recreate it most accurately: http://www.cs.columbia.edu/ coyne/images/clowns.jpg (the floor has a wood texture. the first clown is on the floor. the second clown is on the floor. the first clown is facing the second clown. the second clown is facing the first clown. there is a very large brick wall behind the floor. A large whiteboard is on the wall. the whiteboard is two feet above the floor.)

c. Literary exploration:

- Have students re-create a scene from one of Aesop's fables, including characters, backgrounds, and anything else important to understanding the scene in depth.

Session 2: Working with Fables

a. Introduction:

- Have students open their scene for: The dog is on the ...

- Change the point-of-view with the camera angles.
- Zoom in or out with the camera angles.
- Change the texture of the ground.
- Import an image and add it to the scene.
- Render the final scene.
- Use 2D effects to add another visual element to the scene.
- View scene in My Portfolio and the Gallery

b. Warm-up:

- Provide students with the following text from Aesop's fables. Put students in teams, one for each sentence. Ask them to recreate their assigned sentence in WordsEye so that the entire fable is recreated at the end of the activity: "A monkey perched upon a lofty tree saw some Fishermen casting their nets into a river, and narrowly watched their proceedings. 2) The Fishermen after a while gave up fishing, and on going home to dinner left their nets upon the bank. 3) The Monkey, who is the most imitative of animals, descended from the treetop and endeavored to do as they had done. 4) Having handled the net, he threw it into the river, but became tangled in the meshes and drowned. With his last breath he said to himself, 'I am rightly served; for what business had I who had never handled a net try and catch fish?'"

c. Literary Exploration:

- Have students use WordsEye to re-create their original fables in 2 to 4 scenes. Students can transform each sentence of their fable into a scene or combine a number of different ideas into a scene. The scenes should have enough detail for others to interpret what's happening.
- Save each scene in My Portfolio, and then turn the scenes into a Picturebook using the Picturebook editor.
- Ask students to volunteer to present their scenes. See if other students can discern the moral of the fable from the scenes and presentation.

Session 3: Animal Farm Study (Part 1)

a. Warm-up:

- Give students the following Animal Farm text to recreate as a scene in WordsEye, reminding them that they won't be able to include all of the details and vocabulary: At one end of the big barn, on a sort of raised platform, Major was already ensconced on his bed of straw, under a lantern which hung from a beam. He was twelve years old and had lately grown rather stout, but he was still a majestic-looking pig, with a wise and benevolent appearance in spite of the fact that his tushes had never been cut.

b. Literary Exploration:

- Explain that WordsEye will be a tool to storyboard their Animal Farm skits at the final presentation of the Summer Academy, where the storyboards will be enlarged to poster-size and presented. Start discussion on how they can use WordsEye to plan their skit scenes Have students work in groups to create their skit scenes. They can divide the skit into beginning, middle, and end. Each scene should include the background and foreground for what the audience will see during the performance.
- Save each scene in My Portfolio, and then turn the scenes into a Picturebook using the Picturebook editor.

c. Share Out:

- Each student presents their scenes to the class. Other students give feedback on how to improve for skits.

Session 4: Animal Farm Study (Part 2)

a. Warm-up:

- Ask students to think of their favorite Animal Farm character. Now create a WordsEye scene by placing that character in the middle of New York City. What does the character see? Does the street scene look different because of this character's presence there?

b. Literary Exploration:

- Have students continue designing scenes for their Animal Farm skits. Scenes should be finalized by the end of the session. Tell students that the completed scenes will be made into posters and displayed at the final presentation.
- Save each scene in My Portfolio, and then turn the scenes into a Picturebook using the Picturebook editor.

c. Share Out:

- Have each student volunteer to present their scenes to the class. Other students should provide feedback on how scenes could be improved to make a quality presentation.

Session 5: Final Presentation

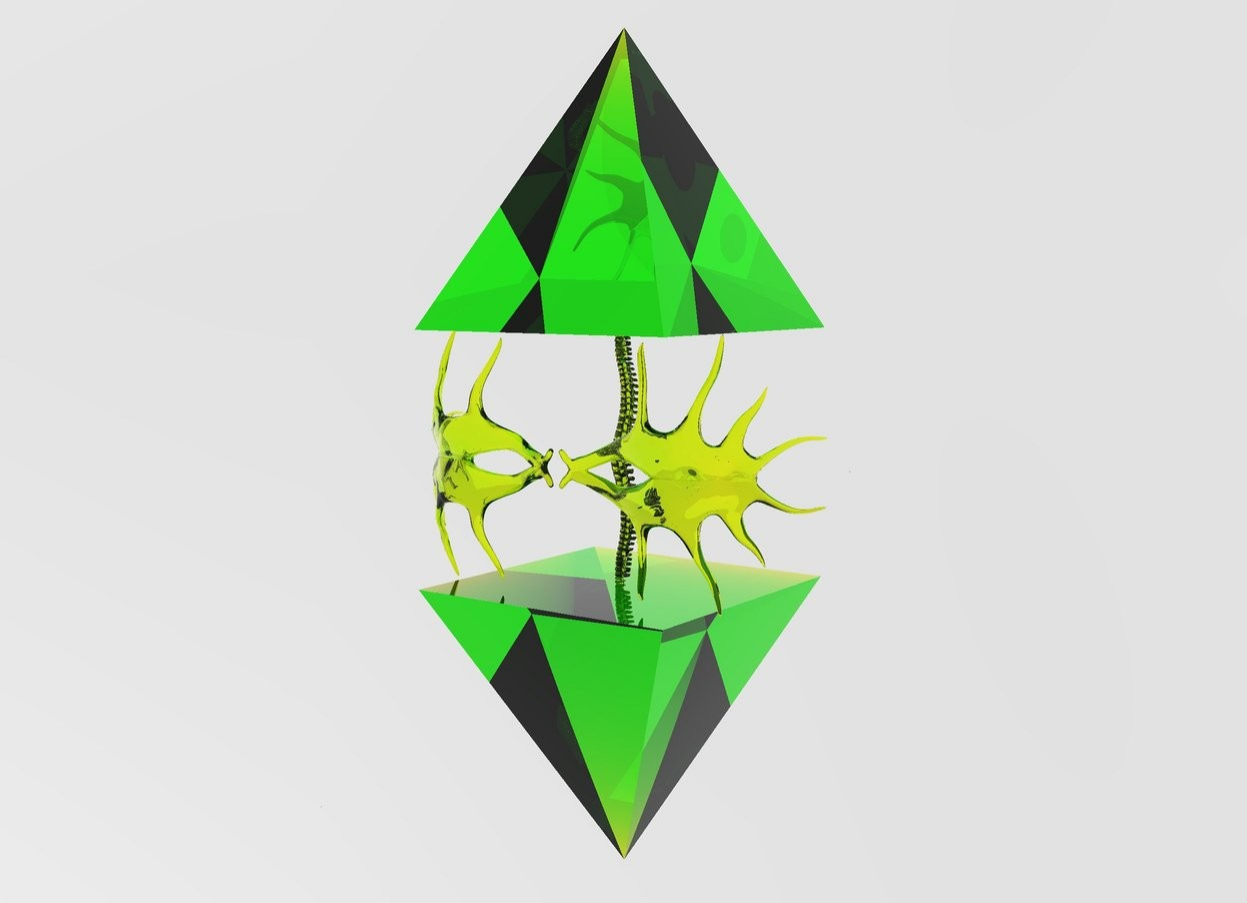
Have each group present their WordsEye creations to the class, giving a synopsis of their stories and showing panels of their work. Have class discuss each scene: characters, imagery, text, connection to Animal Farm.

- Conduct a focus group asking students to discuss:
- What was the one thing you liked best about WordsEye?
- What was the one thing you liked least about WordsEye?
- Did WordsEye help you better understand Animal Farm? If so, how?
- Would you use WordsEye on your own time?
- What did you have to change about your writing in order to interact effectively with WordsEye?
- How is communicating with a computer different than communicating with another person?
- What are some other uses of the WordsEye platform?
- What are other things would you like WordsEye to do?

Appendix C

Website Examples

Alien Composition II (by Whales of Jupiter) – Abstract

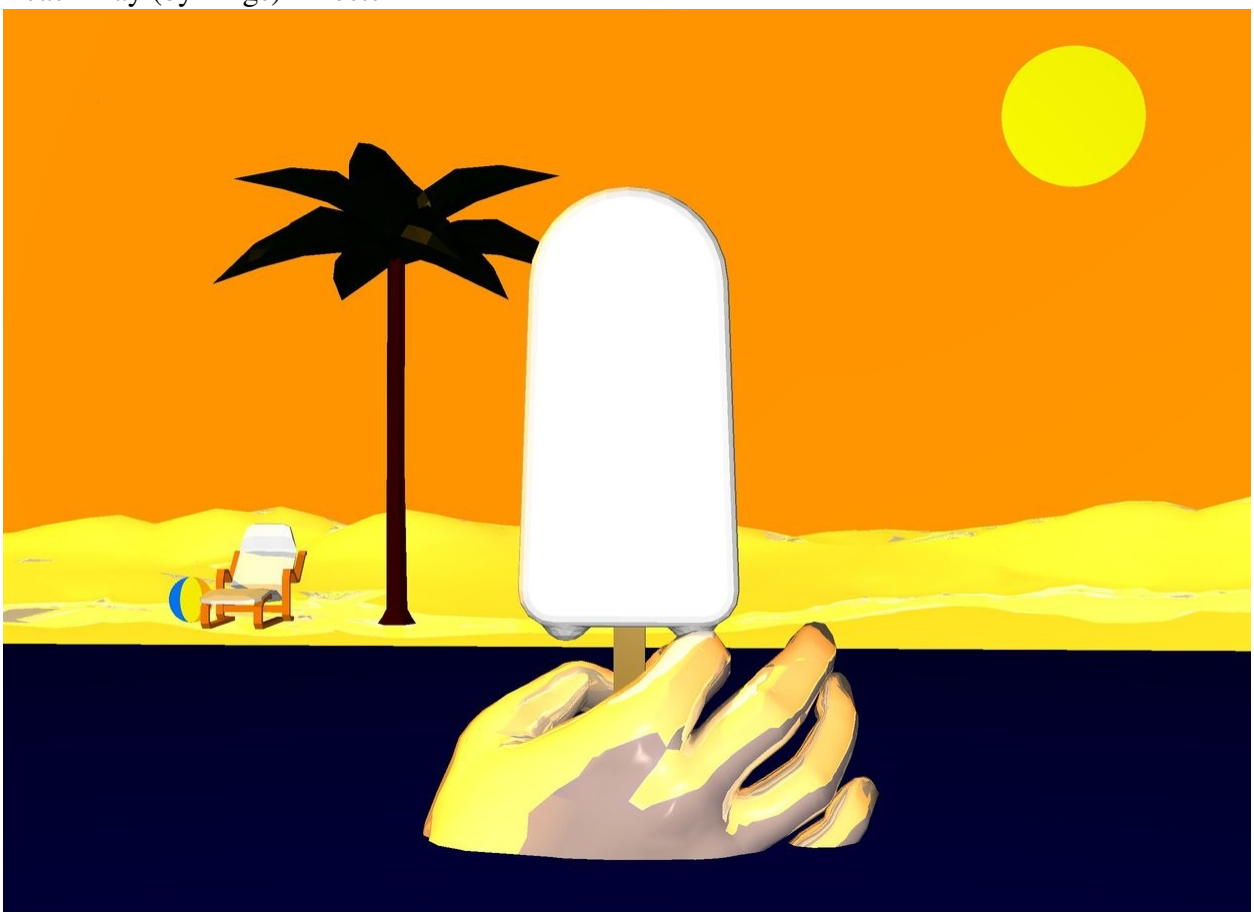


the ground is silver. the sky is white. the first translucent chartreuse irish elk is 5.8 feet in the ground. the 5 feet tall translucent chartreuse pyramid is above the irish elk. the translucent chartreuse spinal cord is beneath the translucent chartreuse pyramid.

The Son of Man (by Ketokeas) – Surrealism

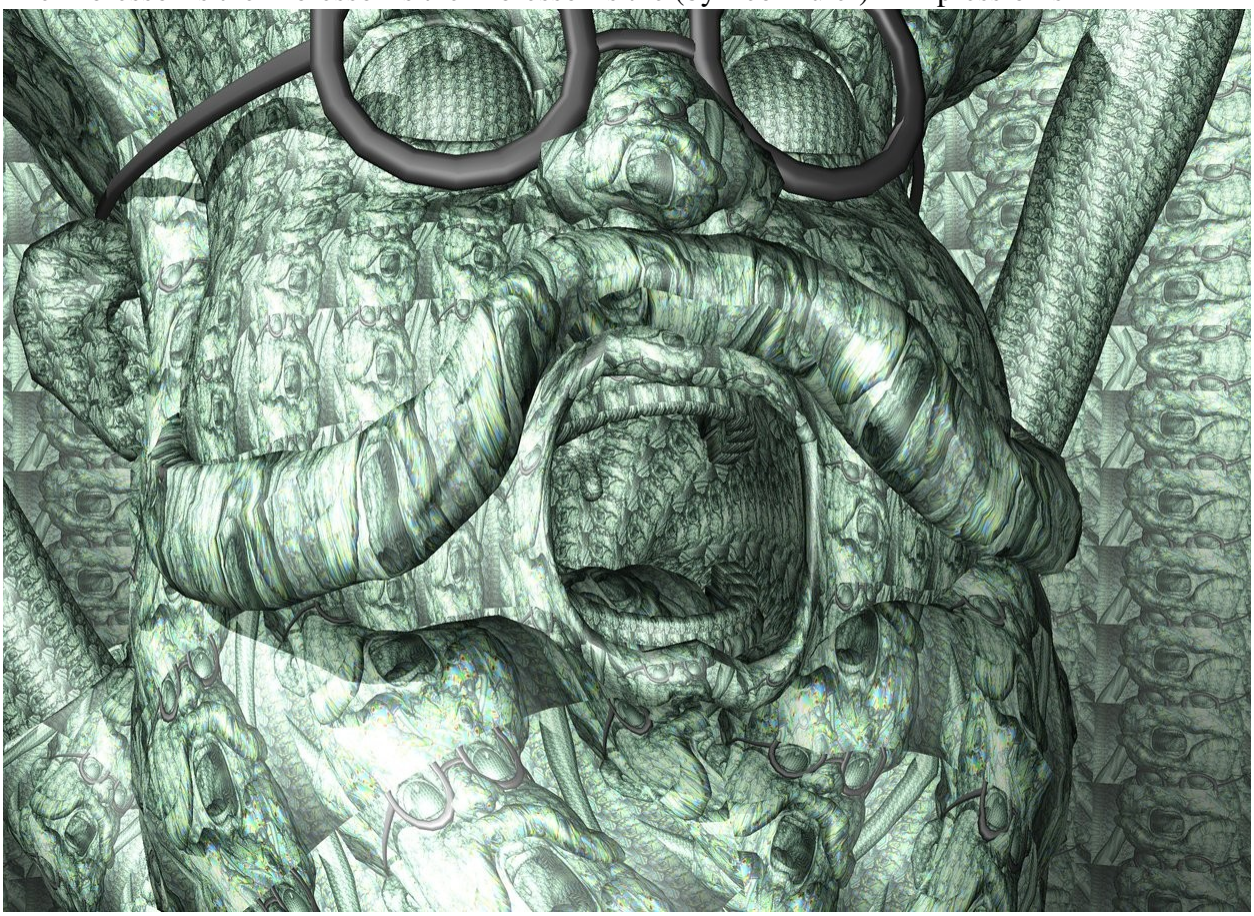
There is a businessman. There is a hat -0.22 feet above the businessman. There is a small green apple if front of the businessman. The apple is 5.25 feet above the floor. The apple is 2 feet in front of the businessman. The ground is transparent. The floor is enormous brick.

Beach Day (by Diigs) – Poster



A extremely enormous white popsicle.the popsicle is facing east. the popsicle is above the huge floor. the sky is orange. the ground is reflective. the sun is orange. the floor is navy. a huge flat yellow sphere above the popsicle. the sphere is 10 feet to the right. a extremely enormous tan shiny hand -5 feet behind the popsicle. the hand is facing north. the hand is leaning 50 degrees to the right. the hand is -9 feet under the popsicle. a flat ball 15 feet behind the floor. the ball is -10 feet to the left. a big white isokon chair to the right of the ball. a tree -8 feet to the right of the isokon chair

The Professor is the Professor is the Professor is the (by LeonEuler) – Expressionism



professor. the professor's shirt is [orp1]. the professor is [orp2]. the professor's beard is [orp3]. the professor's tie is [orp15]. the professor's moustache is [orp5]. the professor's tooth is [orp17]. the professor's belt is [orp7]. the professor's eyebrow is [orp8]. the professor's shoe is [orp16]. the professor's hole is [orp10]. the professor's eye is [orp11]. the professor's gums is [orp12]. the professor's tongue is [orp19]. the sky is [orp14]. the ground is invisible. the [orp19] is 0.1 inches tall.

an unexpected evening in the park (by costlyblood) – Romance

the gigantic unreflective red [sky] wall. the bike is 40 foot east of the wall. the wall is facing west. it is night. the red light is 4 feet east of the bike. the 1st woman is 4 feet in the bike. the 1st woman is -2.3 feet north of the bike. the second woman is -1 inches south of the 1st woman. the 2nd woman is facing the 1st woman. the camera light is black. the second woman is leaning 1 degree to the back. the 1st black tree is 10 feet north of the 2nd woman. the 2nd black tree is 5 feet south of the 2nd woman. the grey light is 10 feet east of the 1st woman.

From time to time (by Kawe) – Cubism

a clear cube.in the cube is a clear sphere.a 1st clock is 10 inch in the sphere.a 2nd silver sphere is 10 inch in the sphere.ambient light is old gold.a 2nd clock is 11 inch in the cube.ground is texture.ground is forget me not blue.

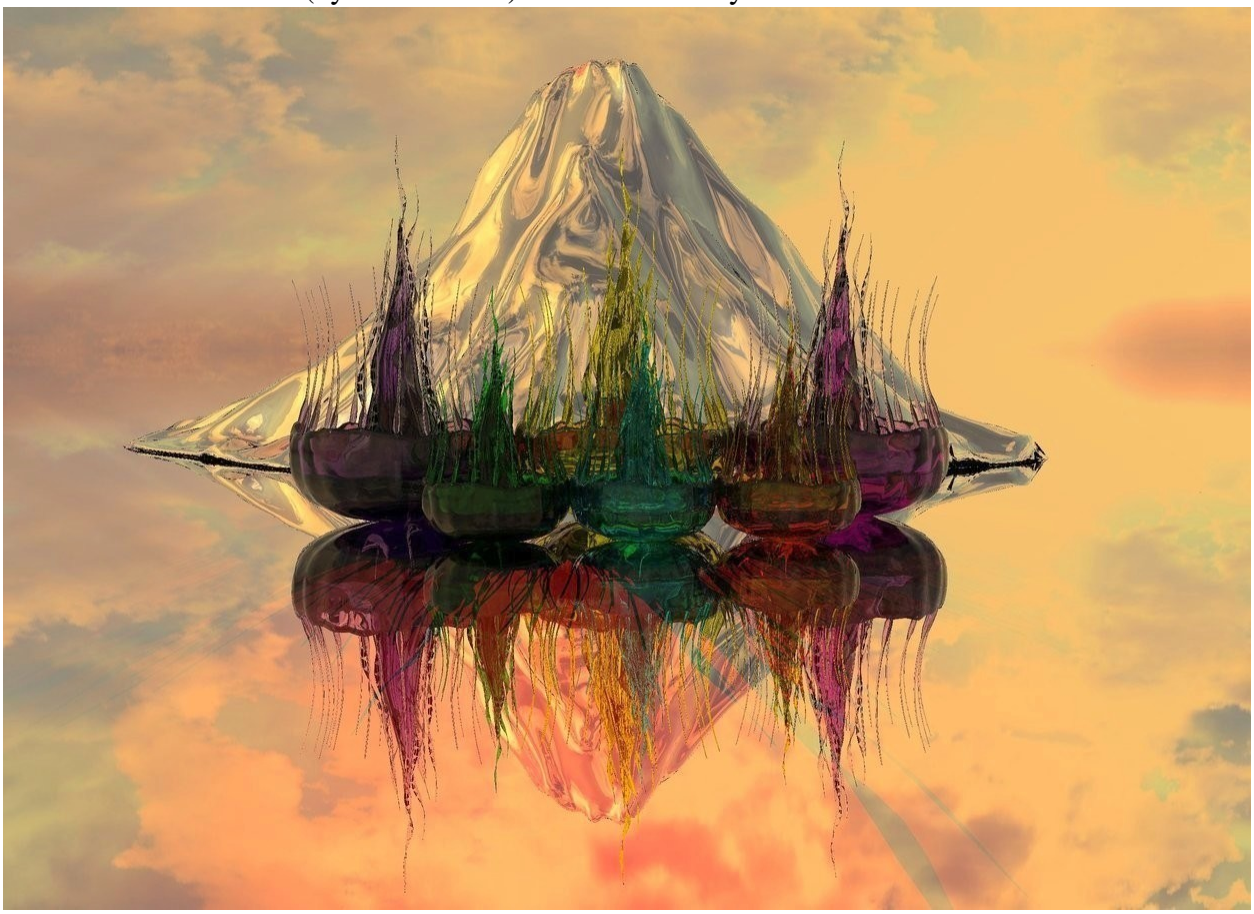
Untitled (by mayuthey) – Humor

There are 4 large ants on a pink plate. The plate is on top of a table. A man is behind the table. A fork is to the left of the plate. A big dog is behind the plate. A light is on top of the dog. The ground is shiny. There is a scarlet "eat" 1 foot to the right of the dog.

lol (by Mrbigmad) – Meme

the ground is clear. the sky is sky texture. there is 2 eyes 2 feet above the ground. there is a nose. the nose is 22 inch above the ground. the nose is below the eyes. there is a mouth below the nose. there is a small cigar. the cigar is in front of the mouth. there is a huge bright white light above the nose. there is a tiny hat. the hat is 6 centimeters behind the eyes. there is a small balloon. the balloon is 0.5 inch below the hat. the hat is 1 inch in the balloon

Arctic Alien Plant Life (by Watcher570) – Old Master Style



a second 15 feet tall clear cyan glass jellyfish is upside down. a fourth 15 feet tall clear green jellyfish is -2 inches left of the second jellyfish. the fourth jellyfish is upside down. a sixth 15 feet tall clear orange jellyfish is -2 inches right of the second jellyfish. the sixth jellyfish is upside down. a third 25 feet tall clear yellow jellyfish is -2 inches behind the second jellyfish. the third jellyfish is upside down. a fifth 25 feet tall clear lavender jellyfish is -2 inches left of the third jellyfish. the fifth jellyfish is upside down. a seventh 25 feet tall clear violet jellyfish is -2 inches right of the third jellyfish. the seventh jellyfish is upside down. a silver volcano is 15 feet behind the third jellyfish. the ground is silver. the camera light is white. the sun is amber. a red light is above the volcano.

The Waiting Game (by Nanook) – Cartoony

1st [paisley] woman faces southeast. the lip of the 1st woman is red. the shoe of the 1st woman is pink. 2nd [pattern] woman is behind and left of the 1st woman. the lip of the 2nd woman is purple. the shoe of the 2nd woman is purple. 3rd [leaf] woman is left of the 2nd woman. she faces southeast. the shoe of the 3rd woman is red. 1st 15 feet long and 10 feet tall [tile] wall is -15 feet to the right of and 3 feet behind the 1st woman. 2nd 15 feet long and 10 feet tall [tile] wall is 4 feet to the right of the 1st wall. 3rd 10 feet tall and 16 feet long [tile] wall is 4 feet behind and -29 feet to the right of the 2nd wall. 4th [metal] woman is 0.3 feet right of and 0.4 feet behind the 1st wall. she faces northwest. a small red "LADIES" is 0.01 feet to the front of and -3.8 feet to the right of the 1st wall. it is 4 feet above the ground. the ground is [brick]. 3 yellow lights are left of the 4th woman. 4 dim yellow lights are 6 inches above the 2nd woman. a 1.7 feet tall [marble] drinking fountain is -2 feet to the left of and 0.1 feet to the front of the 2nd wall. a 4 feet tall mirror is -6.3 feet to the right of and 0.01 feet to the front of the 3rd wall. it is 0.7 feet above the ground. 2 dim green lights are 1 inch above the mirror. 4th 10 feet tall and 20 feet long [tile] wall is 18 feet to the front of the 1st woman. it faces back. the camera light is apple green. 5 dim orange lights are 3 inches above the drinking fountain. 5th woman is right of the 1st woman. she faces the 1st woman. 1 dim rust light is 6 inches above the 1st woman. 1 dim sky blue light is 2 inches in front of the 1st woman.

Appendix D

Words with Facial Expression Mappings

"abhorring" "affronted" "afraid" "agape" "aggravated" "aghast" "agitated" "agog" "alarmed" "amazed" "amused" "angry" "annoyed" "antagonized" "anxious" "appalled" "apprehensive" "astonished" "astounded" "averse" "awestruck" "bereaved" "bitter" "bitter" "blanched" "blessed" "blissful" "blithe" "blown-away" "blue" "bowled-over" "breathless" "bubbly" "captivated" "cheerful" "cheerless" "chipper" "chirpy" "confounded" "confused" "consternated" "contented" "convivial" "cowardly" "cowed" "cross" "crushed" "daunted" "dazed" "dazzled" "dejected" "delighted" "depressed" "despairing" "desperate" "despondent" "disconsolate" "discouraged" "disgusted" "disheartened" "dismal" "dismayed" "dismayed" "displeased" "displeased" "dissatisfied" "distressed" "distressed" "disturbed" "doleful" "down" "down-in-the-dumps" "downcast" "dumbfounded" "ecstatic" "elated" "electrified" "enraged" "euphoric" "exacerbated" "exasperated" "excited" "exuberant" "exultant" "fearful" "fed-up" "ferocious" "fierce" "fiery" "flabbergasted" "floored" "forlorn" "frightened" "frustrated" "fuming" "furious" "galled" "gay" "glad" "gleeful" "gloomy" "glum" "gratified" "grief-stricken" "grieving" "grossed-out" "happy" "hateful" "heartbroken" "heartsick" "heated" "heavyhearted" "hesitant" "horrified" "hotheaded" "huffy" "hurting" "ill-tempered" "impassioned" "incensed" "indignant" "inflamed" "infuriated" "insecure-personality" "intimidated" "irate" "ireful" "irritable" "irritated" "jolly" "joyful" "joyous" "jubilant" "languishing" "laughing" "lively" "livid" "low" "low-spirited" "lugubrious" "mad" "maddened" "melancholy" "merry" "mirthful" "miserable" "morose" "mournful" "nauseated" "nauseous" "nervous" "nettled" "nonplussed" "numb" "offended" "on-cloud-nine" "out-of-sorts" "outraged" "overjoyed" "overwhelmed" "overwrought" "panic-stricken" "peaceful" "peppy" "perky" "perturbed" "pessimistic" "petrified" "playful" "pleasant" "pleased" "provoked" "queasy" "raging" "rattled" "regretful" "reluctant" "repelled" "repulsed" "resentful" "revolted" "riled" "sad" "scandalized" "scared" "scared-stiff" "scared-to-death" "seething" "serene" "shaken" "shocked" "sick-at-heart" "sick-of" "sickened" "smoldering" "somber" "sore" "sorrowful" "sorry" "sparkling" "splenetic" "spooked" "squeamish" "startled" "stormy" "stunned" "stupefied" "sulky" "sullen" "sunny" "surprised" "suspicious-of" "taken-aback" "tearful" "terrified" "terror-stricken" "threatened" "thrilled" "thunderstruck" "tickled" "tickled-pink" "timid" "timorous" "tired" "tragic" "trapped" "trembling" "troubled" "tumultuous" "turbulent" "turned-off" "umbrageous" "uneasy" "unhappy" "unhappy" "unnerved" "unsettled" "unwilling" "up" "upbeat" "upset" "uptight" "vehement" "vexed" "walking-on-air" "weary" "weeping" "wistful" "woebegone" "wondering" "worried" "wrathful" "wretched"

Appendix E

Verb Annotation Target Vignettes, Primitives, and Core Verbs

Level 0: arrive, ask, believe, bounce, bow, bring, call, carry, close, come, connect, cover, crawl, cross, dance, dive, do, drag, drink, drive, drop, eat, end, enjoy, enter, face, fall, float, fly, get, give, go, grab, grasp, grin, hang, have, hear, hit, hold, hop, jump, kick, kneel, knock, know, laugh, lay, lean, leave, lift, like, listen, look, love, make, move, open, play, point, pray, pull, punch, push, put, reach, ride, rise, roll, run, salute, say, scream, see, shoot, shout, show, sit, sleep, smile, speak, stand, start, stop, surround, sweep, swim, swing, take, talk, tap, tell, throw, touch, turn, use, walk, watch, wave, wear, yell

Level 1: abut, aim, arrange, attach, bang, become, bend, blink, block, blow, break, brush, build, bump, catch, change, chase, chop, clamor, climb, collect, collide, combine, crash, cry, curve, cut, demand, deny, die, dig, draw, dream, exit, feed, feel, fight, fill, find, flatten, fold, follow, greet, grow, happen, hate, help, hike, hobble, hug, hurdle, jog, juggle, keep, kill, kiss, lack, lead, leap, let, let go, lie, light, limp, live, load, lower, mix, mow, need, paint, perform, phone, photograph, pick, poke, pour, present, press, protrude, punt, rain, read, recall, regret, remove, reply, rest, revolve, roam, rub, seem, send, separate, set, shake, shine, sing, sink, skate, ski, slap, slide, sneak, sneeze, snow, spill, spin, squeeze, stack, stay, step, stick, strangle, support, swat, telephone, think, toss, travel, trip, twirl, twist, wait, want, wink, wipe, write, yawn

Level 2: adjoin, appear, argue, blast, blend, branch, bridge, burn, bury, clean, clip, compress, cook, create, disappear, divide, dress, embrace, empty, encircle, examine, exercise, exist, explode, fix, flash, forget, form, frighten, frown, glue, grimace, hide, join, meet, nod, pass, pay, penetrate, pin, pitch, pucker, realize, receive, remark, remember, return, rush, sail, saw, scare, scrape, scratch, scrub, search, seek, serve, shampoo, slam, smear, snap, sneer, spit, splash, split, spray, spread, squash, stain, staple, stir, stretch, strike, tackle, tie, vanish, vibrate, warn, wash, whisper, whistle, wrap, wrestle

Level 3: accept, admit, agree, amble, announce, annoy, applaud, arc, arise, ascend, assert, assume, attack, attract, back, bake, barbecue, bark, bat, bathe, beat, beg, begin, bite, bleed, bless, boil, brag, breathe, brighten, buy, caress, carve, celebrate, chat, cheer, chew, choke, choose, claim, clap, clear, coat, complain, complete, consider, continue, cough, count, crack, cram, crave, creep, cringe, crochet, crouch, crush, dart, dash, decapitate, decide, decorate, defend, deflate, deliver, descend, destroy, dine, dip, direct, discover, discuss, display, dribble, drift, drill, drip, duck, dump, dust, emit, enclose, envision, erase, escape, exaggerate, exhale, explain, express, fade, fail, film, finish, flatter, flee, flick, flinch, fling, flip, flop, flow, flutter, forbid, force, fry, fumble, gamble, gaze, giggle, glide, glisten, grind, growl, guard, guess, halt, hammer, hand, head, heat, hitchhike, hope, hover, howl, hunt, hurl, hurry, hurt, imagine, impale, include, inflate, inhale, inspect, interrupt, iron, jam, joke, knife, land, lick, lob, lock, lose, lunge, lurch, lurk, march, marry, meander, measure, melt, moan, mop, nail, name, notice, observe, occur, offend, offer, operate, own, pack, paddle, park, pinch, place, plant, plead, please, plow, pluck, plunge, polish, pop, pose, possess, postpone, pound, powder, power, praise, prance, preach, print, proceed, promise, propel, protect, prune, pry, puff, punish, ramble, recite, recline, recognize, record, reflect, refute, relax, release, remain, remind, repair, repel, replace, respond, ring, rip, roast, row, ruin, sand, saunter, saute, save, screw, select, sew, shave, shift, shiver, shock, shove, shovel, shuffle, sign, sip, skip, sled, slice, slip, slither, smack, smash, smell, smirk, smoke, smudge, soar, spiral, sprint, squat, squirm, squirt, stab, stagger, stamp, stare, start, steal, sting, stoop, store, stray, streak, stroll, strut, study, stumble, stun, suck, suffer, suggest, sulk, surf, surprise, surrender, suspend, sway, switch, tape, taste, teach, thank, threaten, transfer, transform, trap, traverse, tremble, trot, tumble, type, view, visualize, wander, water, wiggle, win, wince, wonder, work, worry, zoom

Level 4: allow, attend, capture, contact, decrease, dim, exclude, expand, finish, flap, flare, group, haul, huddle, hunch over, increase, magnify, mean, multiply, narrow, plan, prohibit, question, quit, recycle, retire, sharpen, shear, shrink, sift, speed, stuff, surge, widen

Level 5: abandon, abduct, abhor, absorb, abuse, accelerate, access, accompany, accuse, achieve, add, adjust, administer, adore, advance, advise, advocate, affect, alert, allay, allude, amaze, amuse, anger, annihilate, anoint, answer, apply, apprehend, approve, arrest, assist, avoid, awake, awaken, babble, base, belch, bellow, belt, bicker, bicycle, bid, bind, bomb, bombard, bore, borrow, bother, broil, bunt, burp, bus, butt, button, camp, cancel, canoe, chance, check, chuckle, clamp, clash, clasp, clutch, comb, comfort, command, compare, compute, conceal, concentrate, conclude, condemn, condone, conduct, confess, confirm, confuse, construct, convict, cool, correspond, crackle, crank, criticize, cure, curl, curtsy, dare, daydream, debate, deceive, decelerate, declare, deduce, defeat, defect, define, defy, delay, denounce, dent, depict, depose, derail, deserve, design, desire, detect, determine, dice, digest, discontinue, disgust, disperse, displease, dissect, dissolve, dock, dodge, donate, doubt, doze, drain, dread, drown, dry, dunk, dwindle, ease, eliminate, elude, embark, embarrass, encourage, endanger, endorse, engrave, enquire, enrage, entertain, envelop, envy, etch, evacuate, evade, exchange, expect, experience, explore, faint, fatten, feast, fit, flicker, flirt, fool, frame, free, fret, gain, gallop, gargle, grip, gripe, groan, grunt, guage, guide, guillotine, gulp, gush, handcuff, handle, harvest, hesitate, hiccup, highlight, hiss, hoard, honk, hoodwink, host, humiliate, hurtle, hush, identify, ignite, imitate, impact, implore, impose, impress, imprison, improve, incant, incarcerate, indicate, infect, infer,

inflame, ingest, inhabit, inherit, inject, inquire, insert, insist, insult, intend, interfere, invade, investigate, invite, jab, knead, knit, learn, loiter, loom, manipulate, marvel, memorise, mention, morph, mug, mutilate, navigate, neglect, obey, ooze, orbit, overlook, owe, pacify, parachute, perspire, persuade, peruse, pilot, plaster, plod, plummet, poison, portray, position, pout, probe, prove, pump, puncture, quiz, radiate, raft, raid, rattle, recoil, refer, register, relay, rely, render, report, require, resemble, reside, resign, retort, reveal, reverberate, review, reward, ridicule, risk, sacrifice, scoop, score, scramble, scribble, shred, shun, sidle, signal, silence, sketch, slay, sling, smoothe, smother, snort, sock, solve, soothe, sort, spare, spark, sparkle, spend, spring, spy, stabilize, starve, strain, string, subtract, summon, suppose, survey, swallow, swear, sweat, tame, terrify, test, thwack, tingle, toast, torture, trace, translate, transport, trim, trudge, trust, tunnel, underline, unearth, urge, vacuum, vandalize, vary, veer, videotape, vote, wake, wallop, warble, warm, waste, waver, wax, weave, weigh, weld, whack, whimper, whine, whip, whiz, wind, wring, yodel