Collecting Semantic Data from Amazon’s Mechanical Turk for a Lexical Knowledge Resource in a Text to Picture Generating System

Abstract

WordsEye is a system for converting from English text into three-dimensional graphical scenes that represent that text. At the core of WordsEye is the Scenario-Based Lexical Knowledge Resource (SBLR), a unified knowledge base and representational system for expressing lexical and real-world knowledge needed to depict scenes from text. This paper explores information collection methods for building the SBLR, using Amazon’s Mechanical Turk (AMT) and manual normalization of raw AMT data. The paper follows with a review of existing relations in the SBLR and classification of the AMT data SBLR semantic relations. Since manual annotation is a time-consuming and expensive approach, we also explored the use of automatic normalization of AMT data through WordNet similarity measures and log-odds and log-likelihood ratios extracted from large corpora.

1. Introduction

The text-to-scene conversion system WordsEye [1], [2] seeks to bridge the gap between language, graphics, and knowledge by developing new theoretical models and technology to enable the automatic conversion of text into a new type of semantic representation – a virtual 3D scene. 3D scenes provide an intuitive representation of meaning in an extended sense by making explicit the contextual elements implicit in our mental models. The natural language component in the current incarnation of WordsEye is built in part on several already existing components, including a stochastic part of speech tagger, a statistical parser and the WordNet (WN) semantic hierarchy [3]. The parsed sentence is first converted into a dependency representation. Then lexical semantic rules are applied to this dependency representation to derive the components of the scene description. The depiction module of WordsEye interprets the scene description to produce a set of low-level depictors representing poses, spatial relations, color attributes, etc. The resulting depictors are then used (while maintaining constraints) to manipulate the 3D objects that constitute the final, renderable scene. Figure 1 indicates the general architecture of the WordsEye system.

The text-to-scene conversion mechanism centers on a new type of lexical knowledge representation, which we call a Scenario-Based Lexical Knowledge Resource (SBLR) [4]. The SBLR is a unified knowledge base and representational system for expressing lexical and real-world knowledge needed to depict scenes from text. It is used in conjunction with the WordsEye system to semantically interpret input text. The SBLR will ultimately include information on the semantic categories of words; the semantic relations between predicates (verbs, nouns, adjectives, and prepositions) and their arguments; the types of arguments different predicates typically take; additional contextual knowledge about the visual scenes various events and activities occur in; and the relationship between this linguistic information and the 3D objects in our objects library.

Alternative methods for building the SBLR have included mining information from external semantic resources such as WordNet, FrameNet, and PropBank, as well as the use of information extraction techniques on other corpora. This paper explores information collection methods using
reviewing some prior work in text to scene conversion, the paper follows with manual review of existing relations in the SBLR and classification of the AMT data into existing and new semantic relations. Then we compare these manual results with automatic normalization of the data through WN similarity measures as well as log-odds and log-likelihood ratios from the Google web corpus and English Gigaword.

2. Prior work

Some systems exist for producing 3D graphics from natural language sources including [5] one of the first text-to-picture conversion systems, the Put system [6], as well as CarSim [7] and AVis [8], which are domain-specific systems to create animations from natural language descriptions of accident reports. Systems also include a system for transforming text sourced from fiction into corresponding 3D animations [9], 3SVD [10] a 3D scene creation system using story-based descriptions, an ontology-driven generation of 3D animations for training and maintenance [11] and CONFUCIUS [12] which is a multi-modal text-to-animation system that generates animations of virtual humans from single sentences containing an action verb. In these systems the referenced objects, attributes, and actions are typically relatively small in number or targeted to specific pre-existing domains.

3. Data collection from AMT

Amazon’s Mechanical Turk is an online marketplace that provides a way to pay people small amounts of money to perform tasks that are simple for humans but difficult for computers. Examples of these Human Intelligence Tasks (HITs) range from labeling images to moderating blog comments to providing feedback on the relevance of results for a search query. The highly accurate, cheap and efficient results of several NLP tasks [13] have encouraged us to explore using AMT.

We collected information about several hundred objects in WordsEye’s database, including information about their typical parts, typical location and typical objects around them. We designed three separate tasks for collecting such information about each target object. Each of the three tasks was performed on more than 300 nouns from our object library with 2 assignments per HIT. In all tasks, the workers had to be inside the US and have a HIT approval rate greater than or equal to 99%.

For task 1, we asked the workers to name 10 common objects that they might typically find around or near a given object. We also requested that the workers not name any items inside the given object. For task 2, we asked the workers to name 10 locations in which they might typically find a given object and in task 3, we asked the workers to list 10 parts of a given object. Given that some objects might not consist of 10 parts, (i.e., they are very simple objects), we wanted the worker to name as many parts as possible. We collected 17,200 responses from the AMT tasks and paid $106.90 overall for completion of the three tasks. Table 1 shows a summary of the AMT tasks, payments, and completion time.

The data that we collected in this step was in raw format. The next step was normalizing the data; that is, mapping data entered by the workers into entities and relations contained within the SBLR. In the next sections we discuss our methodologies for normalizing the raw data. We began by manually normalizing the AMT data and then classifying the data via new and pre-defined semantic relations. Due to the time-consuming nature of the manual annotation task, we also explored automatically normalizing AMT data.

4. Manual normalization of the data

Data collected from AMT tasks was manually normalized via removal of uninformative target-response pairs and definition of the relations between the remaining target-response pairs. Response items given in their plural form were lemmatized to the singular form of the word. A total of 34 relations were defined for the complete sets of Mechanical Turk data. Defining relations was completed manually and determined by pragmatic and/or cultural cues about the relationship held

<table>
<thead>
<tr>
<th>Task</th>
<th>Number of target words</th>
<th>Number of user inputs</th>
<th>Ave. time per assignment</th>
<th>Reward per assignment</th>
<th>Effective Hourly Rate</th>
<th>Completion time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objects</td>
<td>342</td>
<td>6850</td>
<td>2’</td>
<td>$0.05</td>
<td>$1.54</td>
<td>5 days</td>
</tr>
<tr>
<td>Location</td>
<td>342</td>
<td>6850</td>
<td>2’</td>
<td>$0.05</td>
<td>$1.26</td>
<td>5 days</td>
</tr>
<tr>
<td>Parts</td>
<td>245</td>
<td>3500</td>
<td>1’</td>
<td>$0.07</td>
<td>$2.29</td>
<td>5 days</td>
</tr>
</tbody>
</table>
between the target-response pair. Screening Mechanical Turk workers to confirm that they were from the United States ensured that actions or items which might differ in their typically found location by cultural or geographical context (e.g., eat breakfast) were restricted to the location(s) generally agreed upon by English speakers within the United States. Relation definition focused primarily on defining concrete, graphically depictable relationships.

Generic, widely applicable relations were used in the general case for all sets of Mechanical Turk data (e.g., the containment relation containing.r was used for generic instances of containment; the next-to.r relation was used for target-response pairs for which the orientation of the items with respect to one another was not a defining characteristic of their relationship). Finer distinctions were made within these generic relations, e.g. habitat.r and residence.r within the overarching containment relation, which specified that the relation held between two items was that of habitat or residence, respectively. More semantically explicit relations were used for target-response pairs that tended to occur in more specific relations. Specific relations of this type included those spatial relations from the following target item-response item-relation triples: javelin – dirt – embedded-in.r, mobile – ceiling – attached-to.r and binoculars – case – true-containing.r. Another subsection of relations included functional relations such as harmonica – hand – human-grip.r, earmuffs – head – wearing.r and owl – perch – support-for.r. Relation labels for meronymic (part-whole) relations were based on already defined part-whole classifications [14].

4.1. Data and results for each AMT task

Target-response pairs that misinterpreted ambiguous target item (e.g., misinterpreting mobile as a cell phone rather than as a decorative hanging structure, prompting mobile - ear as an object-nearby object pair) were manually rejected. Target-response pairs were also discarded if the interpretation of the target item, though viable, was not contained within the SBLR library. This was especially prevalent in instances where the target item was a plant or animal (e.g. crawfish) that could be interpreted as either a live plant/animal or as food. Since the SBLR does not contain the edible interpretation of some nouns, pairs such as crawfish – plate were discarded in the nearby objects task.

In the object-location task, the most common relation labels were derivatives of the generic spatial containment relation. The containing.r relation accounted for 38.01% of all labeled target-response pairs; habitat.r accounted for 11.02%, and on-surface.r accounted for 10.6%.

In the part-whole task, AMT workers provided responses that were predominantly labeled by part-whole relations. When AMT responses were not relevant for part-whole relations, they tended to fall under the generic containment relation. The object-part.r relation accounted for 79.12% of all labeled target-response pairs; stuff-object.r accounted for 16.33%, and containing.r accounted for 1.48%.

As with the part-whole task, responses in the nearby objects task that were not relevant for the next-to.r relation usually fell under the generic spatial containment relation. In this task, the next-to.r relation was the most frequently utilized relation label, accounting for 75.66% of all target-response pairs labeled. The on-surface.r relation was the second most common relation, with 5.69%, and containing.r accounted for 4.44% of all labeled target-response pairs.

5. Automatic normalization of the data

As stated before, manual normalization of the data is a time-consuming and expensive approach. As a result, we are investigating different automatic techniques to normalize the raw data and filter out the uninformative outputs from AMT, using current manually annotated data as a gold standard for evaluation of the outcomes of automatic approaches.

5.1. WordNet Similarity measures

In the first approach, we computed semantic similarity scores between target-response pairs (i.e., each target word in our object library and the received responses of the AMT tasks) based on WN similarity measures (expanding the target words which were not present in WN to their nearest hypernyms).

The similarity measures included: 1) The WN path similarities between each target word in our object library and the received outputs of the AMT tasks. Here, we selected the maximum similarity score of different senses of target and respond words. 2) The (maximum of ) Resnik similarity between target-response pairs, which returns a score denoting how similar the two word senses are, based on the Information Content (IC) of the Least Common Subsumer (most specific ancestor node)
3) The average pair-wise similarity score based on WN path similarity score. To illustrate this further, if we assume $W_1, W_2, ..., W_n$ as $n$ number of AMT outputs for target word $T$ and $S_{ij}$ as the maximum WN path similarity score between $W_i$ and $W_j$, then the average pair-wise similarity score for $W_i$ will be $(S_{i1} + S_{i2} + ... + S_{in})/n$. 4) The WN matrix similarity which is a bag of words similarity matrix based on WN path similarities. For target word $T$ we have the following similarity matrix where $S_{ij}$ is the maximum WN path similarity score between $W_i$ and $W_j$:

\[
\begin{bmatrix}
1, S_{i1}, S_{i2}, ..., S_{in} \\
S_{j1}, S_{j2}, 1, ..., S_{jn} \\
\vdots \\
S_{m1}, S_{m2}, ..., 1
\end{bmatrix}
\]

Each row of the matrix is the similarity vector of the word in its first column. For instance, $V_i$ is the similarity vector of $W_i$ and represented as $[S_{i1}, S_{i2}, ..., S_{in}]$. We use cosine similarity to calculate the similarity measure of two words. For example, the similarity measure of $W_i$ and $W_j$ is the cosine of $V_i$ and $V_j$ ($CS_{ij}$) and is computed by $CS_{ij} = \frac{V_i \cdot V_j}{||V_i|| \cdot ||V_j||}$ so the WN matrix similarity of $W_i$ will be equal to $(CS_{i1} + CS_{i2} + ... + CS_{in})/n$ (where $n$ is the number of inputs).

5.2. Corpus association measures

The next approach for normalizing the raw data was finding association measures of target-response pairs using Google's 1-trillion 5-gram web corpus (LDC2006T13), by counting the frequency of each target and response word in unigram and bigram portions of the corpus and then the number of times the two words co-occur within a +/- 4-word window in the 5-gram portion of the corpus. We also computed the sentential co-occurrences of each target-response pair (i.e. the number of sentences in which the target and the response words appear and the number of sentences in which both words occur together) on the English Gigaword corpus (LDC2007T07) which is a ~1 billion word corpus of articles marked up from English press texts (mainly the New York Times). Based on these counts, we used log-likelihood and log-odds ratio [16] to compute the association between the two words.

5.3. Discussion and evaluation of automatic normalization techniques

Data collected from each Mechanical Turk test was classified into both a higher-scoring and lower-scoring set of target-response pairs by log-likelihood and log-odds ratio, WN path similarity, Resnik similarity, WN average pair-wise similarity and WN matrix similarity. The higher-scoring pairs were predicted to be relevant AMT outputs; conversely, the lower-scoring set of target-response pairs were predicted to be uninformative AMT outputs. We evaluated the accuracy of each automatic normalization approach by computing the precision and recall against the manually normalized data (table 2). Since collecting data by using AMT is rather cheap and fast, we are more interested in achieving higher precision than high recall. In other words, higher precision means we achieved high accurate data by our automatic normalization and lower recall means we lose some data, which is not expensive to collect.

As can be seen in table 2, the baseline accuracy of the nearby objects task is quite high (precision=0.8934, recall=1.0), and we gain the best precision by using WN average pair-wise similarity (0.9855) by removing lower-scoring part of AMT outputs (recall=0.3215). The high precision in all automatic techniques is due primarily to the fact that the open-ended nature of the task resulted in a large number of target-response pairs that, while not pertinent to the next-to-r relation, could be labeled by other relations. Again, the open-ended nature of the nearby objects task resulted in the lowest percentage of rejected high-scoring pairs (high recall in most of the measures). In the part-

| Table 2: The accuracy of automatic normalization approaches on semantically expanges target-response s |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
|                | **BL**          | **LL**          | **LO**          | **WS**          | **RS**          | **WP**          | **WM**          |
|                | **Per** | **Rec** | **Per** | **Rec** | **Per** | **Rec** | **Per** | **Rec** | **Per** | **Rec** | **Per** | **Rec** | **Per** | **Rec** | **Per** | **Rec** | **Per** | **Rec** |
| **OBJ**        | 0.8934 | 1.0     | 0.9015 | 1.0     | 0.9286 | 0.9144 | 0.9123 | 1.0     | 0.9185 | 1.0     | 0.9855 | 0.3215 | 0.8925 | 1.0     |
| **PRT**        | 0.7887 | 1.0     | 0.7921 | 0.4523 | 0.8321 | 0.5022 | 0.8073 | 1.0     | 0.8234 | 1.0     | 0.9045 | 0.2859 | 0.9010 | 0.2516 |
| **LOC**        | 0.5527 | 1.0     | 0.7832 | 0.6690 | 0.7851 | 0.6684 | 0.5624 | 0.9724 | 0.5674 | 0.9784 | 0.6115 | 0.3657 | 0.4832 | 1.0     |

**OBJ**: Object task; **PRT**: Part task; **LOC**: Location task; **Per**: Precision; **Rec**: Recall; **BL**: Baseline; **LL**: Log-odds; **LO**: WN Path sim.; **RS**: Resnik sim.; **WP**: WN Ave. pair wise sim.; **WM**: WN Matrix sim.
whole task, the best precision (0.9010) was achieved by using WN matrix similarities but again we lost a noticeable portion of data (recall= 0.2516). Rejected target-response pairs from the higher-scoring part-whole set were often due to responses that named attributes, rather than parts, of the target item (e.g. croissant – flaky). Many responses were too general (e.g. gong – material). Many target-response pairs would have fallen under the next-to-r relation rather than any of the meronymic relations. The majority of the approved target-response pairs from the lower-scoring part-whole set were due to obvious, “common sense” responses that would usually be inferred rather than explicitly stated, particularly body parts (e.g. bunny – brain).

Within the object-location data set, we gained the best precision (0.7832) by using log-odds with relatively high recall (0.6690). Target-response pairs that were approved or rejected contrary to automatic predictions were due primarily to the specificity of the response location. Within the higher-scoring set, responses that were too generic (e.g. turntable – store) were rejected. Within the lower-scoring set, extremely specific locations that were unlikely to occur within a corpus or that were not present in WN synsets were accepted (e.g. caliper – architect’s briefcase).

6. Conclusions

In this paper, we investigated the use of information collection methods for building our SBLR, using AMT. Manual evaluation of the AMT outputs (baseline results) confirms that we can collect highly accurate data in a cheap and efficient way by using AMT. Comparison of manually normalized target-response pairs collected from all three AMT tests with the automatic normalization approaches—based on the corpus association measures and WN similarities—reveals that in order to achieve more accurate data (high precision) we will lose a portion of out AMT outputs (low recall). We also obtained better results by semantically expanding some target words and computing the association measures on larger corpora. For the future work, we plan to optimize our automatic normalization techniques by word sense disambiguation of each target and response pair and use Latent Semantic Analysis of to analyze the data based on Wikipedia documents.

References