Collecting Semantic Data by Mechanical Turk for the Lexical Knowledge Resource of a Text-to-Picture Generating System

Masoud Rouhizadeh[*] Margit Bowler[*] Richard Sproat[*] Bob Coyne[**]

[*]Center for Spoken Language Understanding, Oregon Health and Science University
[**]Department of Computer Science, Columbia University

Abstract

WordsEye is a system for automatically converting natural language text into 3D scenes representing the meaning of 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. To enrich a portion of the SBLR, we need to fill out some contextual information about its objects, including information about their typical parts, typical locations and typical objects located near them. This paper explores our proposed methodology to achieve this goal. First we try to collect some semantic information by using Amazon's Mechanical Turk (AMT). Then, we manually filter and classify the collected data and finally, we compare the manual results with the output of some automatic filtration techniques which use several WordNet similarity and corpus association measures.

1 Introduction

WordsEye (Coyne and Sproat, 2001), (Coyne et al., 2010) is a system for automatically converting natural language text into 3D scenes representing the meaning of that text. A version of WordsEye has been tested online (www.wordseye.com) with several thousand real-world users. The system works by first parsing each input sentence into a dependency structure. These dependency structures are then processed to resolve anaphora and other coreferences. The lexical items and dependency links are then converted to semantic nodes and roles drawing on lexical valence patterns and other information in the Scenario-Based Lexical Knowledge Resource (SBLR) (Coyne et al., 2010). The resulting semantic relations are then converted to a final set of graphical constraints representing the position, orientation, size, color, texture, and poses of objects in the scene. Finally, the scene is composed from these constraints and rendered in OpenGL (http://www.opengl.org).

The SBLR is the core of the text-to-scene conversion mechanism. It is a unified knowledge base and representational system for expressing lexical and real-world knowledge needed to depict scenes from 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.

To enrich a portion of the SBLR we need to fill out some contextual information about several hundred objects in WordsEye's database, including information about their typical parts, typical location and typical objects nearby them. Such information can in principle be extracted from online corpora (e.g. Sproat (2001)), but such data is invariably noisy and requires hand editing. Furthermore, precisely because much of the information is *common sense* it is rarely explicitly stated in text. Ontologies of common sense information such as Cyc are effectively useless for extracting such information.

This paper explores our proposed methodology to achieve this goal. First we try to collect some semantic information by Amazon's Mechanical Turk (AMT). Then, we manually filter and classify the collected data and finally, we compare the manual results with the output of some automatic filtration techniques which use WN similarity and corpus association measures.

2 Data collection from Amazon's Mechanical Turk

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 (Callison-Burch and Dredze, 2010) have encouraged us to explore using AMT.

We designed three separate tasks to collect information about typical nearby objects, typical location and typical parts of the objects of our library. For **task 1**, we asked the workers to name 10 common objects that they might typically find around or near a 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.

Task	TW	UI	AA	RPA	EHR	ACT
Objects	342	6850	2´	$0.05	$1.54	5
Locations	342	6850	2´	$0.05	$1.26	5
Parts	245	3500	1´	$0.07	$2.29	5

TW: Number of Target Words; **UI**: Number of User Inputs; **AA**: Average Time Per Assignment;
RPA: Reward Per Assignment; **EHR**: Effective Hourly Rate; **ACT**: Approximate Completion Days

Table 1: Summary of AMT tasks, payments and the completion time

The data that we collected in this step was in raw format. The next step was filtering out undesirable data entered by the workers and mapping it into entities and relations contained within the SBLR.

3 Manual filtering and classifying the data

Data collected from AMT tasks was manually filtered via removal of undesirable target item-response item pairs and classified via definition of the relations between the remaining target item-response item pairs. Response items given in their plural form were lemmatized to the singular form of the word. A total of 34 relations were defined within the Amazon Mechanical Turk data. Defining relations was completed manually and determined by pragmatic cues about the relationship held between the target item-response item pair. Restricting AMT workers to those within the United States ensured that actions or items which might differ in their typically found location by cultural or geographical context were restricted to the location(s) generally agreed upon by English speakers within the United States.

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 item-response item 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 item-response item 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
binoculars - case - true-containing.r

Another subsection of relations included functional relations such as those within the following triples:

harmonica - hand - human-grip.r

earmuffs - head - wearing.r

Relation labels for meronymic (part-whole) relations were based off of already defined part-whole classifications (Priss, 1996).

3.1 Data and results for each AMT task

Target item-response item pairs were usually rejected for misinterpretation of the potentially 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). Target item-response item 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. With the exception of mushroom, the SBLR does not contain the edible interpretation of these nouns; in the object-nearby object task, target item-response item pairs such as *crawfish - plate* were discarded.

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 the object-nearby object 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

4 Automatic filtering undesirable data

Manual processing of the data is a time-consuming and expensive approach. As a result, we are investigating different automatic techniques to filter out the undesirable responses from AMT, using current manually annotated data as a gold standard for evaluation of automatic approaches.

4.1 WordNet Similarity measures

In the first approach, we computed some lexical similarity scores for the target and the response items based on the following WN similarity measures. (It should be noted that not all of the target and responses were present in WN. For such words, we used their nearest hypernyms).

WN Path Distance Similarity between each target word and each received response for that target word. This score denotes how similar two word senses are, based on the shortest path that connects the senses in the is-a (hypernym/hypnoym) taxonomy. We selected the maximum similarity score of different senses of the target and the respond words.

Resnik Similarity between each target word and each of the received responses for that target word. This score denotes how similar the two word senses are, based on the Information Content (IC) of the Least Common Subsumer (most specific ancestor node) (Resnik, 1999).

The Average Pairwise Similarity Score which is computed based on WN path distance similarity score. If we assume $W_1, W_2 ... W_n$ to be *n* responses for target word *T*; and S_{ij} to be the WN path distance similarity between W_i and W_j, then the average pairwise similarity score for W_i will be $\frac{S_{i1}+S_{i2}+...+S_{in}}{n}$. This will provide us the average similarity of each response (i.e W_i) with the other responses (i.e. W_j

so that *i*$\neq$ *j*). In this way we will reward the responses that are more semantically related to each other (regardless of their similarity to the target word).

The WN Matrix Similarity which is a bag of words similarity matrix based on WN path distance similarities. For target word T we have the following similarity matrix:

$$\begin{array}{l} 1 + S_{12} + ... + S_{1n} \\ S_{21} + 1 + ... + S_{2n} \\ \vdots \\ S_{n1} + S_{n2} + ... + S_{nn} \end{array}$$

In this matrix row i is the similarity vector of W_i represented as $\vec{V_i} = [S_{i1} + S_{i2} + ... + S_{in}]$. We use cosine similarity to calculate the similarity measure of two words. So, the similarity measure of W_i and W_j is the cosine of $\vec{V_i}$ and $\vec{V_j}$ and is computed by $CS_{ij} = cos(\theta) = \frac{V_i.V_j}{||V_i||.||V_j||}$. Then the WN matrix similarity score of W_i will be $\frac{CS_{i1}+CS_{i2}+...+CS_{in}}{n}$. The more two words are semantically related to similar set of words, the higher cosine similarity they will have. If a word is related to many different words in the set, it will obtain higher WN matrix similarity score.

4.2 Corpus association measures

The next approach for filtering 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 or 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 (Dunning, 1993) to compute the association between the two words.

4.3 Discussion and evaluation of automatic filtaration techniques

The collected responses of each AMT task were ranked separately by each of the above similarity and association measures. We classify the ranked responses into "keep" (higher-scoring) and "reject" (lower-scoring) classes by defining a specific threshold for each list. Then we evaluated the accuracy of each filtration approach by computing their precision and recall on correct "keep" items (see table 2). In this table the baseline score shows the accuracy of the responses of each AMT task before using automatic filtration techniques. It should be added that collecting data by using AMT is rather cheap and fast, so we are more interested in higher precision (achieving highly accurate data) than higher recall. Lower recall means we lose some data, which is not too expensive to collect.

	Baseline		**Log-likelihood**		**Log-odds**		**WN Path Dist sim**		**Resnik sim**		**WN Pairwise sim**		**WN Matrix sim**	
	Pre	Rec	Pre	Rec	Pre	Rec	Pre	Rec	Pre	Rec	Pre	Rec	Pre	Rec
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
PAR	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
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

Table 2: The accuracy of automatic filtering approaches

As can be seen in table 2, 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.

In the part-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).

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 pairwise similarity (0.9855) by removing lower-scoring part of AMT responses (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.

5 Conclusions

In this paper, we investigated the use Amazon’s Mechanical Turk for collecting semantic information for a portion of our lexical knowledge resource. Manual evaluation of the AMT responses (baseline results in table 2) confirms that we can collect highly accurate data in a cheap and efficient way by using AMT. The accuracy of automatic filtration techniques sounds promising as we were able to filter out some undesirable data, most of the time without loosing so much of collected responses.

Overall, we have shown a method which is very good in collecting semantic information and some other methods which are very good at filtering out word pairs that are undesirable in this particular context (i.e locations, nearby object and parts of our object library). This approach seems to have the potential to be extended for more contexts. For the future work, we are planning to apply this methodology to collect semantic information about *action verbs*, such as information about the locations of the action, the participants, their relation to each other, the background objects and so on.

References

Callison-Burch, C. and M. Dredze (2010). Creating speech and language data with amazon’s mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk*, Los Angeles, CA, USA, pp. 1–12.

Coyne, B., O. Rambow, J. Hirschberg, and R. Sproat (2010). Frame semantics in text-to-scene generation. In R. Setchi, I. Jordanov, R. Howlett, and L. Jain (Eds.), *Knowledge-Based and Intelligent Information and Engineering Systems*, Volume 6279 of *Lecture Notes in Computer Science*, pp. 375–384. Springer Berlin / Heidelberg.

Coyne, B. and R. Sproat (2001). Wordseye: An automatic text-to-scene conversion system. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, Los Angeles, CA, USA, pp. 487– 496.

Coyne, B., R. Sproat, and J. Hirschberg (2010). Spatial relations in text-to-scene conversion. In *Computational Models of Spatial Language Interpretation, Workshop at Spatial Cognition 2010*, Mt. Hood, OR, USA, pp. 9–16.

Dunning, T. E. (1993). Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics 19*(1), 61–74.

Priss, U. (1996). Classification of meronymy by methods of relational concept analysis. In *Online proceedings of the 1996 Midwest Artificial Intelligence Conference*, Bloomington, IN, USA.

Resnik, P. (1999). Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, 95–130.

Sproat, R. (2001). Inferring the environment in a text-to-scene conversion system. In *Proceedings of The First International Conference on Knowledge Capture*, Victoria, BC, Canada, pp. 147–154.