Concept Learning and Categorization from the Web

Abdulrahman Almuhareb (aalmuh@essex.ac.uk)
Massimo Poesio (poesio@essex.ac.uk)
Department of Computer Science / Language and Computation Group
University of Essex
Colchester, CO4 3SQ, United Kingdom

Abstract

In previous work, we found that a great deal of information about noun attributes can be extracted from the Web using simple text patterns, and that enriching vector-based models of concepts with this information about attributes led to drastic improvements in noun categorization. We extend this previous work in two ways: (i) by comparing concept descriptions extracted using patterns with descriptions extracted with a parser, and (ii) by developing an improved dataset balanced with respect to ambiguity, frequency, and WordNet unique beginners.

Introduction

The goal of our research is to develop fully automatic methods to learn from text the associations between a concept and its **attributes**[1]–*e.g.*, to learn that **flights**, unlike **enzymes** or **trials**, have **departure times** and **destinations**. Although this information is considered central for concept definition both in knowledge representation work based on description logics (Baader et al, 2003) and in psychological research on concepts (Murphy, 2004), this information is not present in WordNet (Fellbaum, 1998) (except for information about parts) and is not used in current NLP work on learning concept hierarchies (Curran and Moens, 2002; Lin, 1998; Pantel and Ravichandran, 2004). In previous work (Almuhareb and Poesio, 2004) we demonstrated (i) that a great deal of information about noun attributes[2] can be extracted from the Web, and (ii) that enriching vector-based lexical representations of nouns by including automatically extracted information about attributes leads to drastic improvements in noun clustering. However, our earlier work was limited in two ways. First of all, we only used simple text patterns to identify noun modifiers and noun attributes, whereas parsers are used in most work of this kind. Secondly, an analysis of the relatively few misclassified nouns indicated that many such cases were ambiguous or infrequent nouns, but our original dataset was not designed to fully analyze these cases. The experiments discussed in this paper were designed to remedy these shortcomings. We briefly review the literature and our own previous work. We then discuss a new dataset balanced with respect to ambiguity and frequency, and the methodology we used to build concept descriptions including information about syntactic relations. A new clustering experiment is then discussed. The analysis of the results indicates that using simple text patterns is an efficient method to collect data from the Web. Also, the results indicate that class type and frequency significantly affect the quality of the clustering, while ambiguity has no such effect.

[1] We'll use the term 'attribute' to refer to the notion also referred to in the literature as 'feature' or 'role'.

[2] For the moment our system does not attempt wordsense disambiguation, hence the talk of 'nouns' instead of 'concepts'. These two terms (noun and concept) are used interchangeably in this paper.

Background

Lexical Acquisition with Vectorial Representations

Much of the original work in the acquisition of lexical resources and domain ontologies in NLP used vector-based word representations derived from work in information retrieval (Schuetze, 1992), in which only word associations are recorded. These kinds of representations are still in use, particularly in work on concept acquisition in computational psycholinguistics (Landauer, Foltz, and Laham, 1998; Lund and Burgess, 1996) but most current work in NLP exploits information about grammatical relations extracted using a parser (Curran and Moens 2002; Grefenstette, 1993; Lin, 1998; Maedche and Staab, 2002; Pantel and Ravichandran, 2004). For example, Lin (1998) would represents the noun *dog* as a vector of <syntactic relation, term> pairs such as <adj-mod, brown>. Such vectors are used as the input to clustering. Both hierarchical and non-hierarchical algorithms have been tested, using soft-clustering as well as hard-clustering (e.g., EM), but non-hierarchical hard-clustering is prevalent (for a good discussion, see (Maedche and Staab 2002; Manning and Schuetze, 1999)).[3] The best of the clustering algorithms in (Pantel and Lin, 2002) achieves an F of about 60%.

While the vectorial representations used in this work do capture relational information, the relations in question are purely syntactic—subject, object, adjunct, noun modifier—and even though terms such as **brown** specify values of attributes, no attempt is done to identify terms that specify different values of the same attribute—i.e., to generalize the representation of a concept to include the attribute **color**.

Mining the Web for Attributes

The starting point of this research is previous work attempting to identify particular semantic relations: e.g., **part-of** relations (Berland and Charniak, 1999; Poesio et al, 2002) and **is-a** relations (Caraballo, 1999; Hearst, 1998; Pantel and Ravichandran, 2004). To our knowledge, no

[3] Many researchers attempt to extract **is-a** links directly from text instead of using hierarchical clustering—e.g., Caraballo (1999), Pantel and Ravichandran (2004).

attempt had been made to learn about attributes, nor to use these 'semantic' relations in the vector representation of concepts in replacement of, or addition to, grammatical relations such as those discussed above. We did this in previous work (Almuhareb and Poesio, 2004), building noun descriptions by extracting relational information from the Web via simple patterns used to express queries for the Google API. In addition to a pattern to extract noun modifiers, we also used a pattern to extract (candidate) nominal attributes. This pattern for attributes was based on a linguistic test for attributes first proposed by Woods (1975):

- **A** is an attribute of **C** if we can say [**V** is a/the **A** of **C**],

for example: ***brown*** *is a* ***color*** *of* ***dogs***. The pattern used to identify noun modifiers is shown in (1), that for nominal attributes in (2):

(1) "[a|an|the] * C [is|was]"

(e.g., "… an **inexpensive** car is …")

(2) "the * of the C [is|was]"

(e.g., "… the **price** of the car was ... ")

A variety of ways of using the information extracted from the Web to build vectorial lexical representations were tested. We tested both vectors using only modifiers and only attributes, and vectors using both. Both Boolean vectors and weighted vectors were tried; both raw frequencies and normalized weights obtained using the version of the t-test proposed by Curran and Moens (2002) (shown in (3)) were used as weights:

(3)
$$t_{i,j} \approx \frac{\frac{C(concept_i, attribute_j)}{N} - \left(\frac{C(concept_i) \times C(attribute_j)}{\mathrm{N}^2}\right)}{\sqrt{\frac{C(concept_i, attribute_j)}{N^2}}}$$

where N is the total number of relations, and C is a count function. The modifiers formula is similar. We also tested a variety of clustering algorithms; the best results were obtained using CLUTO's hard-clustering algorithm (Karypis, 2002) and extended Jaccard as vector similarity measure, consistently with what suggested, e.g., by Curran and Moens (2002):

(4)
$$sim(concept_m, concept_n) = \frac{\sum_i (t_{m,i} \times t_{n,i})}{\sum_i (t_{m,i} + t_{n,i})}$$

where $t_{m,i}$ and $t_{n,i}$ are the weighted co-occurrence values between concept **m** and concept n with attribute/modifier i, and computed as in equation (3).

Two evaluations were tried: with the dataset of 34 concepts from 3 classes (animals, body parts, and geographical locations) used by Lund and Burgess (1996) and with a larger set of 214 nouns from 13 different classes in WordNet (Fellbaum, 1998) (buildings, diseases, vehicles, feelings, body parts, fruits, creators, publications, animals, furniture, cloth, family relation, time). The worse results were obtained using vector representations containing only modifiers; better results were obtained using just attributes; but the best results were obtained using both types of information –i.e., combining the 'definitional' information provided by attributes with the 'concordance'-like information provided by modifiers: e.g., although both **cars** and **buildings** have a color, **red** is a much more likely color for cars than for buildings. In fact, using both attributes and modifiers we obtained perfect clustering for the Lund / Burgess dataset. Even with the larger dataset, we obtained very good results: Accuracy 85.51%, F=74.41%—but a total of 31 nouns were misclassified.

The first question raised by this early work is whether classification errors could be further reduced by using a parser to extract information about a greater range of syntactic relations. A second question is how many of these mistakes were due to ambiguity or to data sparsity. Our analysis of the results of the previous experiment did reveal, first of all, that many of the misclassified nouns were ambiguous: e.g., *cancer* has both a **feeling** and a **disease** sense in WordNet; *lounge* can be used to describe both a **building** and a piece of **furniture**. Secondly, we found that many of these misclassified nouns were relatively rare: examples include *abattoir* and *zebra*. The experiments discussed in this paper were designed to address these issues.

The New Experiment

For this new experiment, we designed a balanced dataset containing 402 nouns from 21 WordNet classes and used the RASP parser (Briscoe and Carroll, 2002) to extract grammatical relations (**GRs**).

A Balanced Dataset for Noun Clustering

Our goal was to create a dataset balanced with respect to three factors: *class type*, *frequency*, and *ambiguity*.

First of all, we aimed to include one class of nouns for each of the 21 unique beginners of the WordNet noun hierarchy[4]. We chose subclasses for each of these 21 beginners that would represent a reasonably natural cluster: e.g., the hyponym *social occasion* for the unique beginner *event*. From each such class, we selected between 13 and 21 nouns to be representative concepts for the class (e.g., *ceremony*, *feast*, and *graduation* for the class *social occasion*).

Secondly, we aimed to include about 1/3 high frequency nouns, 1/3 medium frequency, and 1/3 low frequency. Noun frequencies where estimated using the British National Corpus. We considered as highly frequent those nouns with frequency 1,000 or more; as medium frequent the nouns with between 1,000 and 100 occurrences; and those between 100 and 5 as low frequent.

Thirdly, we wanted the dataset to be balanced as to ambiguity, estimated on the basis of the number of senses in WordNet. Nouns with 4 or more senses were considered

[4] WordNet has 25 unique beginners; four of them (body, food, communication, and process) are actually hyponyms of other unique beginners.

highly ambiguous; nouns with 2 or 3 senses medium ambiguous; and nouns with a single sense as not ambiguous.

The final set contains 402 nouns, and each level of frequency and ambiguity is equally represented in the set. The set contains 46 nouns that can be assigned to more than one class belonging to the dataset itself, e.g., *bull* is both an *animal* and a *legal document*.

Vector Descriptions

In the previous experiment, we used three different lexical representations for nouns. In each model, nouns were described using a vector of features; the three models differ in the type of features. The features in the **attribute** model are noun attributes extracted using the pattern in (2), such as *color* and *size* for the noun *car*. The features of the **values** model are nominal modifiers extracted using pattern (1) (we simply call them values as many of them are values of attributes, e.g., *red* for the attribute *color*). The third model, **both**, contains features of the first and the second models.

In this new experiment, we introduced a new model that is based on parsed text. Features of this model are all types of grammatical relations (GRs) produced by RASP. These include nominal modifiers, verb subjects, and conjunctions. For example, the *bear* vector includes:

(Modifier, polar)	832
(Modifier, of, paw)	374
(Conjunction, lion)	517
(Subject, eat)	191

The numbers above are the frequency of encountering each relation with the noun bear in the text.

Web Data Collection

For each noun, we aimed to collect up to 10,000 attributes and values. Concept attributes and attribute values were collected from the Web using the text patterns and the Google search engine as discussed in the background section (Almuhareb and Poesio, 2004). However, we relaxed the text patterns used to collect attributes and values (*"the * of the C is"* and *"the * C is"*) to collect more data for the low frequency concepts. A new pattern for attributes based on the possessive construction was added, *"the C's * is"*. Also, the list of restriction words used to make sure that *C* is a noun (i.e., *is* and *was*) was expanded to include other words, for example: *are*, *were*, *for*, and *will*.

The URLs of the collected pattern instances were used to retrieve documents from the Web. A maximum of 10,000 documents were retrieved for each noun; depending on the number of the collected URLs. Only sentences that contain the targeted nouns were extracted from these documents and parsed to collect grammatical relations that are related these nouns.

Filtering and Weighting Features

A moment's thought will suggest that not all 'attributes' found by means of our patterns correspond to actual attributes: this is because the (quasi) possessive construction, just like any other syntactic constructions, can be used to express a variety of semantic relations. So, for example, "The car's *gone*" compared to "The car's *window*". We are currently developing a classifier to identify actual attributes (Almuhareb and Poesio, in preparation);[5] however, some of these problematic cases can already be identified by means of morphological information and weighting.

Examples like the one above can be identified simply by checking whether the candidate attribute is a noun. This could be done using a POS tagger; we did it by checking if the 'attribute' is in WordNet's noun database. This method also helps in identifying misspelled words.

The t test weighting function (3) was used to select the best features. We only consider positive values produced from the t test weighting function as it shown to produce better results for similar tasks (Almuhareb and Poesio, 2004; Curran and Moens, 2002). We found that increasing the t test threshold from 0 to a higher value does not improve the clustering accuracy. Curran and Moens (2002) found that introducing cutoffs on frequencies improves the accuracy. We achieved the best results using a minimum cutoff at 2 and a maximum cutoff at 5,000 on the accumulated frequency of the attributes/values over all of the concepts. The cutoffs are used to remove very rare and general features.

Clustering Algorithm and Evaluation Measures

Noun clustering was done using the CLUTO clustering toolkit as in the previous study. We used CLUTO's default clustering algorithm, Repeated Bisections, which produces hard globular clusters. Nouns that have more than one possible class are judged to be correctly clustered if they were clustered with any of the possible classes. The pairwise similarities between nouns were computed using the extended Jaccard similarity function as in (4).

Table 1: Entropy and Purity in CLUTO

	Entropy	Purity
Single Cluster	$E(S_r) = -\frac{1}{\log q}\sum_{i=1}^{q}\frac{n_r^i}{n_r}\log\frac{n_r^i}{n_r}$	$P(S_r) = \frac{1}{n_r}\max_i(n_r^i)$
Overall	$Entropy = \sum_{r=1}^{k}\frac{n_r}{n}E(S_r)$	$Purity = \sum_{r=1}^{k}\frac{n_r}{n}P(S_r)$

$\boldsymbol{S_r}$ is a cluster, $\boldsymbol{n_r}$ is the size of the cluster, $\boldsymbol{q}$ is the number of classes, $\boldsymbol{n^i_r}$ is the number of concepts from the ***i*th** class that were assigned to the ***r*th** cluster, and $\boldsymbol{n}$ is the number of concepts and $\boldsymbol{k}$ is the number of clusters.

The clusters were evaluated using CLUTO's *purity* and *entropy* functions. Cluster purity indicates the degree to which a cluster contains concepts from one class only (perfect purity would be 1). Cluster entropy indicates whether concepts of different classes are represented in the

[5] Briefly, we are developing a classifier to automatically classify candidate attributes into parts (e.g., "the *window* of the car"), qualities (e.g., "the *color* of the car"), activities (e.g., "the *selling* of the car"), and related-objects (e.g., "the *driver* of the car").

cluster (perfect entropy would be 0). Overall purity and entropy are the weighted sum of all individual cluster purity and entropy, respectively. The equations for entropy and purity are shown in Table 1.

Results

Comparing Text Patterns to Parsed Text

Table 2 shows the clustering accuracy of different sub-datasets using unfiltered data for the four models. The three sub-datasets (3, 6, and 9 classes) contain 62, 121, and 170 concepts, respectively. The largest sub-dataset contains about 42 % of the total concepts. The results show that there is no advantage from using a model that is based on a parsed text. In fact, using simple text patterns to build a definition model from the Web produces slightly better results. For example: the purity of the largest sub-dataset produced from the combined model of attributes and values built using text patterns is somewhat more accurate than the parsed model (0.882 compared to 0.871).

Table 2: Clustering accuracy for the four models using different number of classes

Description	Attributes	Values	Both	All GRs from Parsed Text
3 Classes				
Purity	0.984	0.823	0.968	0.919
Entropy	0.060	0.465	0.118	0.253
Vector Size	9,586	24,180	33,766	184,610
6 Classes				
Purity	0.959	0.810	0.934	0.934
Entropy	0.093	0.293	0.112	0.134
Vector Size	14,285	40,020	54,305	282,863
9 Classes				
Purity	0.859	0.876	0.882	0.871
Entropy	0.211	0.201	0.180	0.188
Vector Size	15,824	49,584	65,408	332,747

The superiority of the text patterns models is much clearer when looking to the sizes of the vectors in these models. The pattern models are much simpler than the model based on parsed text. For example, in the largest sub-dataset, the vector size of the combined model is about 1/5 the vector size of the parsed model.

An additional advantage of the pattern models over the parsed model is has to do with the complexity of the data collection procedure. Data collection for the pattern models requires only sending the query to the Google search engine, and extracting features from the return results. While the parsed model requires: finding related Web documents, downloading them, preprocessing them, parsing relative sentences, and extracting GRs.

Clustering the Whole Dataset

Table 3 shows the clustering accuracy of the text pattern models using filtered data for the whole dataset. The attribute model produced the best results (e.g., purity=0.709) compared to the value and the combined models (0.627 and 0.664, respectively). The vector size of the attribute models is about ¼ the size of the vector of the value model.

Table 3: Clustering accuracy using filtered models

Description	Filtered Attributes	Filtered Values	Both
Purity	0.709	0.627	0.664
Entropy	0.283	0.338	0.302
Vector Size	12,345	51,345	63,690

Effect of Class, Frequency, and Ambiguity on Clustering Accuracy

The effect of class, ambiguity and frequency on clustering accuracy was measured using the purity of the cluster to which a concept was assigned as evaluation measure. A one-way ANOVA with cluster purity as the dependent variable was computed for each factor. The calculation matrix used for these one-way ANOVAs is a 402 × 2 matrix, with one row for each concept: the first column specifies the value of the factor (e.g. ambiguity level, or class ID), and the second column is the purity of the cluster to which the noun is assigned. We found that the class factor has a significant main effect on the clustering accuracy ($F(20,381) = 46.045$, $P < 0.0005$) as does frequency ($F(2,399) = 3.554$, $P < 0.05$), but we found no significant main effect for ambiguity. The average purity of the low-ambiguity concepts was higher than that of the high-ambiguous and medium-ambiguous concepts (0.724 compared to 0.695 and 0.708), but the test showed no significant main effect of this.

Uniqueness of Common Attributes Among Concepts of Different Classes

Different classes vary widely on the degree of uniqueness of the common attributes among their concepts. Table 4 shows the top 5 common attributes that are shared between concepts of each class in the dataset. Also, it shows the average of the degree of uniqueness of these common attributes. The degree of uniqueness of $attribute_j$ of $class_i$ is computed as the following:

$$Uniqueness_{i,j} = \frac{C(class_i, attribute_j)^2}{n_i \times C(attribute_j)}$$

where $\boldsymbol{n_i}$ is the number of concepts in $class_i$. $\boldsymbol{C}$ is a count function that counts concepts that are associated with the given attribute. Uniqueness ranges from 0 to 1.

Certain classes such as *animal* and *vehicle* have more unique attributes than the other classes. This results in more accurate clustering for their concepts. For example, the purity of the cluster with animal majority is 1.000, and the purity of the vehicle majority cluster is 0.875. On the other hand, some classes such as *game* and *pain* do not have such useful attributes which results in having less accurate clusters. For example, the purity of the game majority cluster is 0.636 and the purity of the cluster with pain majority is 0.524. This result is particularly intriguing at the light of Wittgenstein's use of the concept 'game' as an example of concept whose instances share few or no attributes (Wittgenstein, 1953).

Table 4: Top common attributes of each class in the dataset

Class	**Top 5 Common Attributes** (Average Uniqueness)
animal	liver, intestines, stomach, skull, fur (0.81)
vehicle	tires, windshield, backseat, motor, brakes (0.75)
creator	ingenuity, initials, expertise, imagination, widow (0.64)
edible fruit	pulp, ripeness, juice, peel, tartness (0.55)
monetary unit	devaluation, depreciation, pegging, overvaluation, convertibility (0.52)
social occasion	venue, eve, attendees, evening, occasion (0.43)
district	citizens, geology, topography, landscape, mayor (0.41)
social unit	founder, membership, leadership, chief, missions (0.41)
legal documents	signing, negotiation, issuance, amendment, wording (0.39)
chemical element	combustion, corrosion, bioavailability, solubility, absorption (0.36)
solid	vertices, symmetries, vertexes, surfaces, triangles (0.28)
time	fashions, trends, weather, artists, dictator (0.28)
assets	quantum, payment, maximisation, allocation, proceeds (0.27)
illness	pathogenesis, diagnosis, etiology, outbreak, complications (0.27)
physical property	derivative, measuring, scaling, logarithm, reciprocal (0.26)
feeling	ardour, reawakening, listener, incarnation, spent (0.25)
atmospheric phenomenon	winds, brunt, roar, rumbling, swath (0.23)
tree	leaves, bark, foliage, trunk, wood (0.22)
motivation	embodiment, quickening, insanity, promptings, reproach (0.13)
pain	pain, worst, pathophysiology, severity, cure (0.10)
game	finals, final, winners, game, stands (0.09)

Conclusions

The main expected advantage of using a parser over simple text patterns is that working off the output of a syntactic parser allows to generalize across patterns instantiations: e.g., the instances: '*the color of C*', '*the final color of C*', and '*the surprisingly rich color of C*' can all be used to identify color as a possible attribute of C. Much recent work on using the Web as a corpus suggests that this usage can alleviate data sparsity problems. The work discussed here indicates that with enough data, there may be less need to generalize across syntactic patterns; we can find enough information using just the simplest patterns.

We also hope the dataset proposed here can be a first step towards developing common evaluation criteria for lexical acquisition. (The dataset is discussed in greater detail in (Almuhareb and Poesio, Submitted.)

Acknowledgments

Abdulrahman Almuhareb is supported by King Abdulaziz City for Science and Technology (KACST), Riyadh, Saudi Arabia. We want to thank Google for making their Web API available to the research community, George Karypis for the CLUTO clustering toolkit, and Ted Briscoe & John Carroll for the RASP parser.

References

Almuhareb, A. and Poesio, M. (2004). Attribute-Based and Value-Based Clustering: An Evaluation. In *Proc. of EMNLP*. Barcelona, July.

Almuhareb, A. and Poesio, M. (2005). Evaluation of Attribute-Based and Value-Based Clustering Using a Balanced Dataset. Submitted.

Baader, F., Calvanese, D., McGuinness, D., Nardi, D. and Patel-Schneider, P. (Editors). (2003). *The Description Logic Handbook*. Cambridge University Press.

Berland, M. and Charniak, E. (1999). Finding parts in very large corpora. In *Proc. of the 37th ACL* (pp. 57–64). University of Maryland.

Briscoe, E. and Carroll, J. (2002). Robust accurate statistical annotation of general text. In *Proceedings of the Third International Conference on Language Resources and Evaluation* (pp. 1499-1504). Las Palmas, Gran Canaria.

Caraballo, S. A. (1999). Automatic construction of a hypernym-labeled noun hierarchy from text. In *Proc. of the 37th ACL*.

Curran, J. R. and Moens, M. (2002). Improvements in automatic thesaurus extraction. In *Proc. of the ACL Workshop on Unsupervised Lexical Acquisition* (pp. 59–66).

Fellbaum, C. (Editor). (1998). *WordNet: An electronic lexical database*. The MIT Press.

Grefenstette, G. (1993). SEXTANT: Extracting semantics from raw text implementation details. *Heuristics: The Journal of Knowledge Engineering*.

Hearst, M. A. (1998). Automated discovery of WordNet relations. In Fellbaum, C. (Editor). *WordNet: An Electronic Lexical Database*. MIT Press.

Karypis, G. (2002). *CLUTO: A clustering toolkit. Technical Report 02-017*. University of Minnesota. At http://www-users.cs.umn.edu/~karypis/cluto/.

Landauer, T. K., Foltz, P. and Laham, D. (1998). Introduction to Latent Semantic Analysis. *Discourse Processes*, 25, (pp. 259-284).

Lin, D. (1998). Automatic retrieval and clustering of similar words. In *Proc. of COLING-ACL* (pp. 768-774).

Lund, K. and Burgess, C. (1996). Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instrumentation, and Computers*, 28, (pp. 203-208).

Maedche, A. and Staab, S. (2002). Measuring Similarity between Ontologies. In *Proc. Of the European Conference on Knowledge Acquisition and Management - EKAW-2002*. Madrid, Spain, October 1-4.

Manning, C. D. and Schuetze, H. (1999). *Foundations of Statistical NLP*. MIT Press.

Murphy, G. L. (2004). *The Big Book of Concepts*. MIT Press.

Pantel, P. and Lin, D. (2002). Discovering Word Senses from Text. In *Proceedings of KDD-02* (pp. 613-619). Edmonton, Canada.

Pantel, P. and Ravichandran, D. (2004). Automatic Labeling of Semantic Classes. Proc. NAACL.

Poesio, M., Ishikawa, T., Walde, S. and Vieira, R. (2002). Acquiring lexical knowledge for anaphora resolution. In *Proc. of LREC*. Las Palmas. June.

Schuetze, H. (1992). Dimensions of meaning. In *Proceedings of Supercomputing '92* (pp. 787-796).

Wittgenstein, L. (1953). *Philosophical Investigations*. Blackwell.

Woods, W. A. (1975). What's in a link: Foundations for semantic networks. In Bobrow, D. G. and Collins, A. M. (Editors). *Representation and Understanding: Studies in Cognitive Science* (pp. 35-82). Academic Press, New York.