



Learning Lexical Clusters in Children's Books

Edmond Lau
6.xxx Presentation
May 12, 2004



Vision

- Children learn word patterns from repetition
 - Cinderella's "glass slippers"
-> Barbie's "plastic slippers"
- How far can statistical regularity explain children's language learning?

Children demonstrate an amazing ability to learn the meanings and correct usages of many new vocabulary words every day. Intuitively, the more times a child sees and hears a particular phrase, such as Dr. Seuss's "green eggs and ham," the more likely the child will be able to correctly use that word pattern in everyday conversation. Moreover, from experiencing a phrase like "Cinderella's glass slippers," a child might also reinforce her confidence to use a related phrase such as "Barbie's plastic slippers." These observations suggest that regularity in word patterns may play a critical role in language learning mechanisms.

Yip and Sussman, in developing a computational model for understanding phonological knowledge, have examined the roles of sparse representations, near misses, and negative examples as mechanisms that enable children to learn language from experience. If we are to understand how human beings acquire and use knowledge about language, however, we also need to examine the role of statistical regularity in language learning; in particular, we need to determine whether the statistical frequency with which children experience certain word patterns also contributes significantly to a child's ability to learn language.

As a first step toward answering this question, I propose a new idea called *lexical clusters*, based on Deniz Yuret's lexical attraction model and Steven Larson's clustering of statistically similar data, to investigate the impact that statistical regularity may have on language learning mechanisms in children. Using the Java implementation of a lexical attraction parser as the starting point, I implemented a system that discovers lexical clusters; the purpose of my project is to explore the extent to which statistical regularities can explain how children learn related words and phrases.



Powerful Ideas

Yuret's Lexical Attraction Model

+

Larson's Clustering Maps



"Lexical Clusters"

The core of my project centers around a method for integrating Deniz Yuret and Steven Larson's two powerful ideas for exploiting statistical regularity with unsupervised learning algorithms. Yuret demonstrated that by simply using the likelihood of pairwise relations between words, lexical attraction models of language can correctly identify 35.5% of the word links in the 20 million words of the Penn Treebank corpus. Larson, on the other hand, demonstrated that by clustering together collections of statistically similar information vectors, a system can develop symbolic class definitions grounded on the statistical processing of experience. These two ideas are in fact mutually compatible; the frequency table of word links constructed by Yuret's parser can provide the statistical information necessary to automatically generate class definitions.

Combining Yuret and Larson's ideas, I present a concept called *lexical clustering* to group together related words based on subsymbolic descriptions acquired from a lexical attraction parser. A *lexical cluster* is a collection of related words that possesses statistically similar linkage structures with other words. Related words, such as *gold*, *silver*, and *siladium*, exhibit the property that they can all be used in similar phrases. For instance, the words can all interchangeably modify nouns such as *rings*, *alloy*, and *coins*; however, none of them would be used to describe words such as *dog*, *cat*, or *mouse*. In terms of Yuret's lexical attraction model, related words therefore possess similar linkage structures with other words, and the statistical frequency with which related words appear in related contexts serves as an indicator of word similarity. The statistical frequency of word links, however, must be kept distinct from the frequency of the individual words; the word *siladium* appears less frequently in everyday conversation than the words *gold* and *silver* even though all three words may belong to the same lexical cluster.

Clustering Algorithm

“the city mouse and the country mouse”

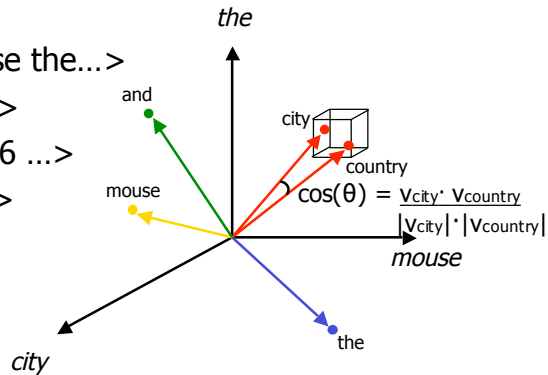
<city country mouse the...>

- city: <0 0 2 3 ...>

- country: <0 0 3 6 ...>

- the: <3 6 2 0 ...>


- ...



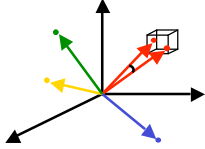
My project implementation consists of two main components: a lexical attraction parser and a clustering algorithm. Professor Winston supplied the Java code for a parser based on Yuret's lexical attraction model, which I integrated into my system with a few minor modifications. With each input sentence, the parser updates a table of word frequencies and a table of linkage frequencies between each pair of words. The second major component is the clustering algorithm, also implemented in Java, used to build groups of related words from the data accumulated by the parser. This slide illustrates the key ideas behind how the clustering algorithm determines groups of related words. As a motivating example, I suppose that the parser has analyzed the phrase “the city mouse and the country mouse” and trace how the clustering algorithm would determine that the words *city* and *country*, which modify mouse in exactly the same way, belong to the same lexical cluster.

The clustering algorithm first creates an N-dimensional feature space, where N is the total number of unique words parsed, and associates a distinct word to each dimension. Thus, the algorithm might assign the first dimension to *city*, the second to *country*, the third to *mouse*, etc. For each word *i*, the algorithm then constructs a length-N linkage vector, where the *j*th term in the vector denotes the number of links that the parser has ever assigned to word *i* and the word associated with the *j*th dimension. The left portion of the slide illustrates that the linkage vector for the word *city* might show that the parser has linked *city* to itself 0 times, to *country* 0 times, to *mouse* 2 times, and to *the* 3 times; similarly, *country*'s linkage vector might show that the parser has linked the word to *city* and *country* 0 times, to *mouse* 3 times, etc. Because only three dimensions can be illustrated graphically, the picture on the right only shows a projection of the N-dimensional space onto the three dimensions specified by *mouse*, *city*, and *country*.

From the N resulting linkage vectors, the algorithm then builds a similarity matrix by calculating the similarity between each pair of words *a* and *b* from the cosine of the angle between the two linkage vectors. This similarity metric essentially determines the extent to which a pair of words links to all other words in the same proportions; it assumes a value ranging from 0 (very dissimilar) to 1 (very similar). The algorithm then determines whether two words belong to the same cluster by comparing their similarity value to a threshold. Continuing with the example, the graph on the right shows that the linkage vectors for *city* and *country* as being closely aligned; the cosine of the angle between the two vectors would exceed the specified threshold parameter, and the algorithm would consequently group the two words together into a lexical cluster. All other words shown are too different to be clustered together. The final step in the clustering algorithm involves merging together clusters that share common words, again based on a parameter specifying the degree of overlap required for two clusters to be merged. The clustering algorithm runs in $O(N^2)$ time and uses $O(N^2)$ space, where N is the number of unique words parsed.




Experiment



- Train on 10 children's short stories and fables (> 20,000 words)
- Iterate:
 - Tweak parameters
 - Execute clustering algorithm

To examine the effectiveness of the clustering algorithm and to illustrate the concept of lexical clusters, I conducted an experiment to find lexical clusters in children's books. I chose to run the parser on children's books rather than on the Penn Treebank corpus because my vision involved determining the role that lexical clusters may play in language learning mechanisms in children. In particular, I executed the parser on ten different children's short stories and fables, including some of my childhood favorites such as *Cinderella*, *Jack and the Beanstalk*, and six chapters from *Alice in Wonderland*. The textual database totaled 20,663 words with 2200 unique words.

Using the link frequencies accumulated by the parser, I performed several iterations of 1) tweaking the similarity and merging thresholds and 2) running the clustering algorithm to find lexical clusters. The nature of this experiment has two implications. First, because the standard for rating the correctness of lexical clusters is inherently subjective, an objective statistical analysis of the experimental results is not possible. For example, the words *hurried* and *died* might be clustered together because they are both verbs, but they might also be separated into two separate clusters due to their semantic disparity. Second, because the size of the textual database pales in comparison to the 20 million words used by Yuret, the accuracy of the parser's results should also be considerably lower; this limitation, however, is mitigated by the consideration that children have a significantly smaller word bank than an adult reading the Wall Street Journal.



Lexical Cluster Results


- Sample Clusters:
 - country, trap, little, city
 - knight, gentleman, lived, person
 - handsomer, five, four, seven
 - hasted, run, hurried
 - watched, let, fed, dragged, carried, hid, killed, entirely, drive

This slide presents a flavor for some of the lexical clusters returned by the clustering algorithm. Though imperfect, each of the clusters presented above convey a general class definition grounded on statistical regularity. The first cluster contains words related to *mouse*, most likely due to a parse of the story “The City Mouse and the Country Mouse.” The second cluster illustrates some words related to people while the third shows a cluster of numbers. The fourth cluster contains words specifically related to fast movement, and the fifth cluster consists of words that are primarily past tense verbs.

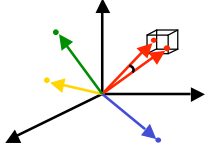
A post-experimental analysis of the data, particularly the first cluster example shown in the slide, reveals a surprising discovery. Yuret contends in his paper that from the standpoint of maximizing mutual information, the direction of word links has no effect on the probability of a particular linkage structure and thus lexical attraction between two words is symmetric. However, when dealing with a probabilistic framework that involves semantics, the results illustrate that the direction of word links matters significantly. The story “The City Mouse and the Country Mouse” contains numerous uses of the terms *city mouse*, *country mouse*, *little mouse*, and *mouse trap*. The lexical attraction model of language merely encodes the linkage frequencies of these phrases and ignores the direction of the word links, even though the word *mouse* functions in a different role as a modifier in the last phrase. Had the parser taken into account the direction of word links, the unrelated word *trap* would most likely not have been categorized into the first cluster.

The lexical clustering results presented in the slide are not representative of the result set. The vast majority of lexical clusters actually consisted of unrelated words, and the results illustrated in the slide were selectively chosen. Nonetheless, the fact that some empirical support exists for the unsupervised learning of lexical clusters is still quite amazing. The inaccuracy of the results can primarily be attributed to the smaller word bank used by the parser combined with lack of direction in the word links.

Overall, I conclude from the results that even though some statistical regularity does indeed exist for the unsupervised learning of lexical clusters, the regularity is insufficient to contribute to a major theory of language learning in children. The lexical clustering results exhibit too much noise to account for the relative ease with which children can correctly use word phrases that they see and hear even just once or twice.



Bootstrap Learning




- Larson's Bootstrapping Idea
 - Learn, cluster, then learn some more
- Design Extension
 - Incorporate clustering algorithm into unsupervised learning

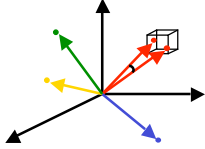
In addition to the conducting the experiment, I also explored a possible way in which lexical clusters may help to bootstrap the language learning process. In his paper, Larson also presents the idea that classification based on statistical similarities enables symbols to be bootstrapped from experience. His Intrinsic Representation system received hand and eye sensory inputs and then constructed clusters and associations across the two sensory maps; each subsequent trigger of a hand or eye cluster would also trigger the associated cluster in the other sensory map.

Using the clustering algorithm developed for this project, a similar idea can also be incorporated into Yuret's lexical attraction parser. An experiment seeking to explore the bootstrapping of words from experience would parse some fraction of the input sentences and then run the clustering algorithm to find lexical clusters; for each subsequent word link identified, the parser would not only increase the linkage frequency of the word pair, but also increase the linkage frequencies with words that belong to the same lexical clusters as the word pair.

This bootstrapping concept opens the door for a possible improvement in the accuracy of the Yuret's lexical attraction parser. Moreover, unlike the experiment performed for this project, this proposed experiment, if executed on the Penn Treebank corpus, could generate objective and measurable accuracy ratings that can be compared to Yuret's numbers to determine the viability of bootstrapping language learning using lexical clusters.



Contributions



- Combined two powerful ideas that exploit regularity
- Developed a clustering algorithm to find lexical clusters
- Experimented with lexical clusters on children's books

In my project, I have:

1. Combined Yuret and Larson's powerful ideas for exploiting regularity to formulate the concept of lexical clusters as a mechanism for identifying similar words.
2. Developed and implemented a clustering algorithm to compare word similarity based on the frequency tables constructed by Yuret's lexical attraction parser.
3. Experimented with the algorithm on ten children's books and short stories in order to illustrate lexical clustering.
4. Discovered that even though some statistical regularity exists for forming lexical clusters, the amount of regularity is insufficient to generate accurate clusters for the majority of data.
5. Applied Larson's bootstrapping idea to the learning of similar words as a possibility for improving the identification accuracy of Yuret's lexical attraction parser.