Finding Attributes in the Web Using a Parser

Abdulrahman Almuhareb & Massimo Poesio
Department of Computer Science
Language and Computation Group
University of Essex
aalmuh@essex.ac.uk & poesio@essex.ac.uk

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 by comparing concept descriptions extracted using patterns with descriptions extracted with a parser. Our results show that it is computationally more efficient to use simple text patterns than parsing text.

1 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 natural language processing (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 partially limited because we only used simple text patterns to identify noun modifiers and noun attributes, whereas parsers are used in most work of this kind. The experiments discussed in this paper were designed to remedy this shortcoming. We briefly review the literature and our own previous work. We then discuss 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 patterns is an efficient method to collect data from the Web.

2 Background

2.1 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

[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 word sense discrimination, hence the talk of 'nouns' instead of 'concepts'.

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 et al, 1998; Lund and Burgess, 1996) but most current work in NLP exploits information about grammatical relations (GRs) 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 hard-clustering as well as soft-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**.

2.2 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 attempt had been made to learn about the more general form of attributes used here (including also qualities), 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. In previous work (Almuhareb and Poesio, 2004) we developed a first method to do this, building noun descriptions by extracting relational information from the Web via simple text 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[4]. 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

[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).

[4] We developed an attribute classifier in (Poesio and Almuhareb, 2005) that classifies candidate attributes into: parts & related objects, qualities, related agents, activities, and non-attributes.

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 *t*-test weighting function from Manning and Schuetze (1999) were used:

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

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):

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

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 (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 of 85.51%, *F* of 74.41%—but a total of 31 nouns were miss-clustered.

A first question raised by this preliminary work is how many of the mistakes made by our clustering algorithm were due to class type, ambiguity, and data sparsity. This issue has been addressed in (Almuhareb and Poesio, 2005). Briefly, we designed a new dataset of 402 concepts (nouns). The dataset is balanced in terms of class type, ambiguity, and frequency. The concepts belong to 21 different classes. These classes are chosen to be representative classes of all of the unique beginners of WordNet. The dataset contains an equal number of concepts that are very frequent, medium frequent, and low frequent; the same applies for ambiguity. Our experiment revealed that class type and frequency have significant effects on accuracy of clustering, while ambiguity has no such effect.

A second question raised by this early work is whether clustering errors could be further reduced by using a parser to extract information about a greater range of syntactic relations. We would also expect that using the parser would allow us to extract a variety of information about modifiers and attributes that we couldn't collect with our rigid patterns. For example, we could relax constructions such as "`the size of the window is`" to allow for modifiers, as in "… `the` **`modified`** `size of the` **`large`** `window` …". The experiments discussed in this paper were designed to investigate this question.

3 Extracting Relations Using a Parser

For this experiment, we collected Web documents and parsed them to extract grammatical relations (GRs) related to the concepts in our balanced dataset, using then this information to cluster the concepts. We discuss our data collection methods in this section, the clustering methods in the next.

3.1 The Web as a Corpus

What makes working with the Web different from working with other corpora is that the Web is a *huge* source of information that offers a large variety of *dynamic* content, not all of which is reliable.

The advantage of working with a huge amount of data is that the chances of finding enough information are greatly increased. On the other hand, dealing with so much data requires an enormous amount of processing power. As we will see below, the issue of efficiency is crucial when comparing using text patterns with using parsing.

A second important difference between the Web and other corpora is that the Web is dynamic. Things that didn't exist in the Web yesterday may be in today. On the contrary, things that were in yesterday may not be in today.

A third issue with using the Web is that it contains a lot of false information. False content can be of two types: intended and unintended. By unintended false content we mean erroneously formatted data, that when processed by computer produce incorrect information. Intended false information, on the other hand, is included on purpose. One example of false content that we have encountered during our work is what is called invisible or hidden text. Here, the designer of the Web page includes a large amount of invisible text that can't be seen directly when browsing the page. The purpose of this text is to deceive search engines to make the page relevant to certain search requests and hence increase Website traffic.

3.2 Web Document Processing

We process Web documents in two main steps. In the first step, we download the document from the Web and extract from them relevant blocks of text. The documents for this experiment were selected using a subset of the URLs gathered in our previous work using text patterns (Almuhareb and Poesio, 2005). These URLs were collected from the result instances of search requests built using our attribute and value text patterns, and returned by the Google API.

We only process documents formatted in HTML. (Other types of documents could also be processed if accessed though their HTML versions generated automatically by Google.) A block of text is a piece of text that ends with an HTML tag that breaks the flow of the text such as *new line tags* and *end of paragraph tags*. A block can be as small as a single word, and as big as a whole paragraph. Only blocks that contain one or more occurrences of the targeted concepts are extracted. This is to insure that we only process appropriate contents. The HTML parsing was done using the HTML parsing package from the Java 2 APIs

In the second step, we parse the collected data using the RASP parser (Briscoe and Carroll, 2002). We start by splitting the collected blocks into sentences. Irrelevant sentences get filtered out. Next, we tag the remaining sentences with part of speech tags, and again remove sentences in which the targeted concepts are not tagged as nouns. In the final part, we parse and generate grammatical relations for the remaining text.

3.3 Vector Descriptions

In our previous experiments, 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 **attributes** 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 is a combined model that contains features of the first and the second models. In this new experiment, we introduced a fourth model that is based on parsed text. Features of this model are all types of grammatical relations (GRs) produced by RASP. Table 1 shows the most frequent GRs of the concept (noun) *strawberry* found in the collected data.

(detmod, strawberry, the)	(ncmod, fruit, strawberry)
(detmod, strawberry, a)	(ncmod, strawberry, wood)
(ncsubj, be, strawberry)	(ncsubj, grow, strawberry)
(ncmod, strawberry, fresh)	(dobj, eat, strawberry)
(ncmod, strawberry, wild)	(ncsubj, have, strawberry)
(ncmod, plant, strawberry)	(ncmod, strawberry, frozen)
(xcomp, be, strawberry)	(ncmod, strawberry, cup)
(ncmod, strawberry, whole)	(ncmod, strawberry, red)
(ncmod, strawberry, ripe)	(dobj, have, strawberry)
(ncmod, strawberry, cultivated)	(ncmod, of, variety, strawberry)
(iobj, with, garnish, strawberry)	(ncmod, strawberry, modern)
(detmod, strawberry, an)	(ncmod, cultivar, strawberry)
(ncmod, variety, strawberry)	(ncsubj, grow, strawberry, obj)
(dobj, slice, strawberry)	(ncmod, strawberry, big)
(ncmod, strawberry, large)	…
(dobj, like, strawberry)	

Table 1: Most frequent GRs of the noun *strawberry*.

3.4 Data Collection

We collected about 2.8GB of data containing 6,300,559 blocks of text for the entire set of 402 concepts. The average number of blocks per Web document is about 7 blocks. Using this average we can estimate the number of documents that has been accessed to be about 900,000 Web documents. The average document download and blocks extraction time is about 2.4 seconds per document on a Pentium IV with 1GB memory.

The text blocks that have been collected contain 7,118,491 relevant sentences. These sentences include 16,943,977 relevant GRs. The average preprocessing and parsing speed is about 3.2 sentences per second. The total number of collected attributes and values GRs is about 4.5M out of about 17M total GRs. This is about the same number of attributes and values instances that have been collected using text patterns.

The first result of this work is that collecting attributes and values using simple text patterns is a much simpler and faster than collecting them using parsed text from the Web. The average time for collecting such relations using patterns is about 0.60 second/relation, while the average time when using the parser is about 0.97 second/relation assuming that document URLs are given. Therefore, using the parser to collect relations from the Web that can be captured using text patterns is only justified if the text patterns method fails to collect enough information.

4 Clustering with Parsed Data

4.1 Clustering Algorithm and Evaluation Measures

We clustered nouns using the CLUTO clustering toolkit as in our previous studies. 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).

Description	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	$Entropy = \sum_{r=1}^{k}\frac{n_r}{n}E(S_r)$	$Purity = \sum_{r=1}^{k}\frac{n_r}{n}P(S_r)$

Table 2: Entropy and Purity in CLUTO.
S_r is a cluster, n_r is the size of the cluster, q is the number of classes,
n^i_r is the number of concepts from the ith class that were assigned to the rth cluster,
n is the number of concepts, and 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 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 2.

4.2 Comparing Text Patterns to Parsed Text

Table 3 compares the clustering accuracy for the three vector construction methods using text patterns with that of the vector construction method using parsing. The experiment was performed in several steps due to the extensive time required for parsing data from the Web. Roughly, it took about five months to complete this experiment. We did the clustering once the data became available. This also allowed us to examine the performance using sets of different sizes. The results show that there is no advantage from using a model that is based on a parsed text when using all types of grammatical relations (see next section). Models built using text patterns produced better results in all of the cases. In each case, either the attributes model or the attributes/values combined model has the best accuracy. For example: the purity for the complete dataset produced from the combined model built using text patterns is more accurate than the parsed model (0.677 compared to 0.614).

Description	Attributes	Values	Attributes & Values	All GRs
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
21 Classes (the complete dataset)				
Purity	0.657	0.567	**0.677**	0.614
Entropy	0.335	0.384	**0.296**	0.360
Vector Size	24,178	94,989	119,167	276,501*

* Only GRs that occurred more than once. Actual vector size is 535,901.

Table 3: Clustering accuracy for the four models using different number of classes

4.3 Clustering Using Selected Types of Grammatical Relations

In work discussed elsewhere, we tested whether better results could be obtained by using a subset of all grammatical relations: only attributes, attributes and grammatical objects, etc. We found that reducing the number of GRs considered does result in an improved model—in fact, in a model which is better than the model constructed using text patterns (*purity* = 0.701 compared to *purity* = 0.677) although the improvement in accuracy was not statistically significant.

4.4 Analyzing the Clustering Solution of the Best Case

Table 4 shows the confusion matrix of the clustering solution of the best case, the reduced GRs model. The complete solution is shown in Appendix A.

Class	Cluster																				
	00	**01**	**02**	**03**	**04**	**05**	**06**	**07**	**08**	**09**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
animal						18				1		1									
assets			14								1			1		2			1		2
atmospheric phenomenon										1			1				1	13	1	2	2
chemical element									18			3									1
creator							16												1		
district														18							3
edible fruit												17									1
feeling				1													11			3	
game					12													2	2	2	2
illness	5			13																	
legal document		11	5													1		2	1		
monetary unit		1											1		17						2
motivation			1														6	2	2	7	
pain	1			6									1							10	
physical property		1														12		2	1	4	1
social occasion								2					1						17		
social unit								10		1	8			1							1
solid													12		1	2		1		4	
time			1	1				1			4						3	1	6		
tree												1			1	1					16
vehicle										14											

Table 4: Confusion matrix for the best clustering solution.

The results show that clustering accuracy depends significantly on class types (Almuhareb and Poesio, 2005). Concepts of certain classes seem to be very similar to each other and have very distinct features that make them very different from concepts from other classes. As a result, most of the concepts of such classes occur in almost separate clusters. The classes: *animal*, *creator*, *chemical element*, *district*, *edible fruit*, *monetary unit*, and *vehicle* are among these classes. Most of these classes are physical classes. Concepts from another type of classes appear to be very similar but they share some features that make them also similar to concepts from some other classes. For example: *tree*, *social occasion*, *feeling*, and *illness*. The last three mentioned classes are abstract classes. Classes such as: *time* and *motivation* found to be heterogonous classes. Concepts of such classes are very dissimilar; they occur in several clusters. Concepts from these two classes have not been a majority in any cluster. As a result, and because the number of clusters is fixed to 21, concepts of two other classes, *illness* and *social unit*, become the majority of four different clusters, two clusters for each class.

Cluster assignments of some of the miss-clustered concepts (nouns) are not completely incorrect. These miss-clustered nouns have been clustered according to a valid sense that

doesn't exist in our gold standard (WordNet). For example, the following miss-clustered nouns with their valid senses: *sciatica* (illness/pain), *party* (social occasion/social unit), *coco* (tree/edible fruit), *ball* (solid shape/social occasion), *capital* (assets/district), *nation* (social unit/district), *chill* (physical property/atmospheric phenomenon), and *kiwi* (tree/edible fruit).

5 Conclusions

Most recent work on lexical clustering (Lin, 1998; Pantel and Ravichandran, 2004) uses parsers to build the vector descriptions of concepts. The main advantage over simple text patterns one can expect from using a parser is that working off the output of a syntactic parser makes it possible 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. 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 simple patterns. This result is consistent with much recent work on using the Web as a corpus suggesting that this can alleviate data sparsity problems.

Acknowledgments

Abdulrahman Almuhareb is supported by King Abdulaziz City for Science and Technology (KACST), Riyadh, Saudi Arabia.

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). Concept Learning and Categorization from the Web. In *Proc. of CogSci*. Italy, July.

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. In *Proc. of NAACL*.

Poesio, M. and Almuhareb, A. (2005). Identifying Concept Attributes Using a Classifier. In *Proc. ACL-SIGLEX Workshop on Deep Lexical Acquisition.* Ann Arbor, USA, June.

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).

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.

Appendix A: Best Case Clustering Solution for the 402 Concepts

Majority Class	Concepts*
animal	bear, bull, camel, cat, cow, deer, dog, elephant, horse, kitten, lion, monkey, puppy, rat, sheep, tiger, turtle, zebra
assets	allocation, allotment, credit, fund, gain, income, interest, investment, mortgage, profit, quota, share, taxation, wager, (*bond, check, cheque, licence, obligation, incentive, nonce*)
atmospheric phenomenon	airstream, blast, cloudburst, cyclone, hurricane, lightning, rainstorm, sandstorm, thunderstorm, tornado, twister, typhoon, wind, (*nap, whist, floater, straddle, mania, superego, chill, snap, jag, quaternary*)
chemical element	aluminium, bismuth, cadmium, calcium, carbon, copper, germanium, helium, hydrogen, iron, lithium, magnesium, nitrogen, oxygen, platinum, potassium, titanium, zinc
creator	architect, artist, builder, craftsman, designer, developer, farmer, inventor, maker, manufacturer, musician, originator, painter, photographer, producer, tailor
district	borderland, borough, caliphate, canton, city, country, county, kingdom, land, metropolis, parish, prefecture, shire, state, suburb, sultanate, town, village, (*nation, capital*)
edible fruit	apple, banana, berry, cherry, fig, grape, lemon, lime, mango, melon, olive, orange, peach, pear, pineapple, strawberry, watermelon, (coco, oyster, charcoal, gold, silver)
feeling	conscience, desire, fear, happiness, impulse, joy, love, passion, pleasure, suffering, wonder, (*deterrence, ethics, life, morality, obsession, possession, future, hereafter, moment, clemency*)
game	baccarat, basketball, bowling, chess, football, golf, keno, lotto, rugby, soccer, tennis, volleyball
illness	anthrax, cholera, flu, plague, smallpox, (*burn*)
illness	acne, arthritis, asthma, cancer, cirrhosis, diabetes, eczema, glaucoma, hepatitis, leukemia, malnutrition, meningitis, rheumatism, (*earache, headache, lumbago, migraine, neuralgia, sciatica, menopause, pain*)
legal document	acceptance, assignment, bill, constitution, convention, decree, draft, law, opinion, statute, treaty, (*mark, extension*)
monetary unit	cordoba, dinar, dirham, dollar, drachma, escudo, franc, guilder, lira, peso, pound, riel, rouble, rupee, shilling, yuan, zloty, (*ovoid, cinchona*)
pain	ache, backache, bellyache, soreness, sting, stinging, tenderness, throb, toothache, torment, (*coolness, heaviness, reflexion, shortness, concavity, crinkle, droop, fluting, crosswind, drizzle, anger, sadness, shame, curling, handball, compulsion, disincentive, incitement, inducement, motivator, urge, wanderlust*)
physical property	deflection, diameter, length, mass, momentum, plasticity, poundage, radius, sensitivity, temperature, visibility, weight, (*indentation, taper, casuarina, margin, payoff, sequestration*)
social occasion	celebration, ceremony, commemoration, commencement, coronation, enthronement, feast, fete, fiesta, fundraiser, funeral, graduation, inaugural, pageantry, prom, rededication, wedding, (*date, day, epoch, gestation, period, yesteryear, dispensation, snowfall, constructor, beano, raffle, rescript, dynamic, occasion, stretch*)
social unit	brigade, confederacy, family, household, league, legion, platoon, team, tribe, troop, (*aeon, dance, party*)
social unit	agency, branch, bureau, club, committee, company, department, office, (*today, tomorrow, tonight, yesterday, venture*)
solid	corner, cube, cuboid, cylinder, dodecahedron, dome, icosahedron, knob, octahedron, ring, salient, tetrahedron, (*cloud, penny, stitch, ball*)
tree	acacia, chestnut, conifer, hornbeam, jacaranda, mandarin, mangrove, oak, palm, pine, pistachio, rowan, samba, sapling, sycamore, walnut, (*hoard, trove, aurora, fog, neon, anchorage, riverside, seafront, kiwi, faro, softball, cent, fen, glow, divan*)
vehicle	aircraft, airplane, automobile, bicycle, boat, car, cruiser, helicopter, motorcycle, pickup, rocket, ship, truck, van, (*mouse, shower, house*)

∗ Concepts that don't belong to the majority class are between parentheses and in italic.