submitted to *International Joint Conference on Artificial Intelligence (IJCAI-03)*, Acapulco, August 12-15, 2003

Inducing criteria for lexicalization parts of speech using the Cyc KB, and its extension to WordNet

Tom O'Hara
Computer Science Department
New Mexico State University
Las Cruces, NM 88001
tomohara@cs.nmsu.edu

Stefano Bertolo and **Bjørn Aldag** and **Nancy Salay** and **Jon Curtis** and **Michael Witbrock** and **Kathy Panton**
Cycorp, Inc.
Austin, TX 78731
{bertolo,aldag,nancy,jonc,witbrock,payton}@cyc.com

Abstract

We present an approach for learning criteria for part-of-speech classification by induction over the lexicon contained within the Cyc knowledge base. This produces good results (73.3%) using a decision tree that incorporates semantic features (e.g., Cyc ontological types), as well as syntactic features (e.g., headword morphology). Accurate results (90.5%) are achieved for the special case of deciding whether lexical mappings should use count noun or mass noun headwords. For this special case, comparable results are also obtained using OpenCyc (86.9%), the publicly available version of Cyc, and the Cyc-to-WordNet translation of the semantic speech part criteria (86.3%).

1 Introduction

We use the term *lexical mapping* to describe the relation between a word and its syntactic and semantic features in a semantic lexicon. The term *lexicalize* will refer to the process of producing these mappings, which are referred to as *lexicalizations*.[1] Selecting the part of speech for the lexical mapping is required so that proper inflectional variations can be recognized and generated for the term. Although often a straightforward task, there are special cases that can pose problems, especially when fine-grained speech part categories are used.

In particular, deciding whether the headword in a phrase should be lexicalized as a mass noun is not as straightforward as it might seem. There are guidelines available in traditional grammar texts, as well as the more technical linguistics literature. But these mainly cover high level categories, such as substances, the prototypical category for mass nouns, and concrete objects, the prototypical category for count nouns. However, for lower-level categories the distinctions are not so clear, especially when the same headword occurs in different types of contexts. For example, "source code" is a mass noun usage, whereas "postal code" is a count noun usage.

In addition, sometimes the same word will be a mass noun in some contexts and a count noun in others, depending on the underlying concept. For example, "anthrax" will be a mass noun when referring to the bacteria, but it will be a count noun when referring to the resulting skin lesions (e.g., "anthraces"). There has been much work on the coercion of count nouns into mass nouns (and vice versa), such as the 'grinding rule' [Briscoe *et al.*, 1995], a special case of which covers animal terms becoming mass nouns when referring to the food (e.g., "Let's have pig tonight."). However, there has been little work on determining whether terms should be lexicalized via mass nouns or count nouns. The work here illustrates how this can be done by learning a decision tree based on the ontological types of the underlying concept. Thus, it relies only upon semantic criteria.

Motivation for this comes in the context of building a large-scale knowledge base (KB), namely Cyc [Lenat, 1995]. Traditionally at Cycorp, there has been a split in the knowledge engineering, with the domain knowledge being entered separately from the lexical knowledge. The reason for this is that the knowledge engineers might not be familiar with the linguistic considerations necessary for performing the mappings accurately. They also might not be familiar with all the lexicalization conventions to allow for consistent lexical knowledge entry. To alleviate this bottleneck, the linguistic criteria can be inferred from the knowledge base, exploiting the large number of previous decisions made by lexical knowledge engineers regarding speech part selection.

After an overview of the Cyc knowledge base in the next section, Section 3 discusses the approach taken to inferring the part of speech for lexicalizations, along with the classification results. This section also includes an extension in which the semantic criteria are mapped into WordNet. This is followed by a comparison to related work in Section 4.

2 Cyc knowledge base

In development since 1984, the Cyc knowledge base is the world's largest (120,000 concepts, one million-plus axioms)[2] formalized representation of commonsense knowledge [Lenat, 1995]. The Cyc KB divides roughly into three layers. The upper ontology contains a formalization of only

[1] The term *lexicalization* is used in a broader sense than that traditionally used in grammatical literature: "fossilized" words (i.e., no longer morphologically decomposable [Huddleston and Pullum, 2002]).

[2] These figures and the results discussed later are based on Cyc KB version 576 and system version 1.2577. See www.cyc.com/publications.html for detailed documentation on the KB.

the most general and fundamental of distinctions (e.g., tangibility versus intangibility; objects versus stuff; being an instance of a class versus a specialization, or subclass). In sharp contrast, the lower ontology is a grab bag of facts useful for particular applications, such as web searching, but not necessarily representative of commonsense reasoning (e.g., that "Dubya" refers to *President George W. Bush*). Between the two layers lies the middle ontology, where consensus, or commonsense knowledge about the world is encoded (e.g., *once something dies, it stays dead* or *open containers lose their contents when turned upside-down*). In addition, the KB includes a broad-coverage English lexicon mapping words and phrases to terms throughout the KB.

2.1 Ontology

Central to the Cyc ontology is the concept *collection*, which corresponds to the familiar notion of a set, but with membership intensionally defined (so distinct collections can have identical members, which is impossible for sets). Every object in the Cyc ontology[3] is a member (or *instance*, in Cyc parlance) of a collection. Collection membership is expressed using the predicate *isa*, whereas collection subsumption is expressed using the transitive predicate *genls* (shorthand for "generalization"). These predicates correspond to the set-theoretic notions *element of* and *subset of* respectively and thus are used to form a partially ordered hierarchy of concepts. For the purposes of this discussion, the *isa* (and for collections, *genls*) assertions on a Cyc term constitute its *type definition*.[4]

Figure 1 shows the type definition for *PhysicalDevice*,[5] a prototypical denotatum term for count nouns.[6] The type definition of *PhysicalDevice* indicates that it is a collection that is a specialization of *Artifact*, etc. As is typical for terms referred to by count nouns, it is an instance of the collection *ExistingObjectType*. Note that the 'Mt' labels refer to microtheories, which is the way that knowledge is compartmentalized in Cyc. (*G-Mt* indicates a general microtheory.)

Figure 2 shows the type definition for *Water*, a prototypical denotation for mass nouns. Although the *asserted* type information for *Water* does not convey any properties that would suggest a mass noun lexicalization, the *genls* hierarchy of collections does. In particular, the collection *ChemicalCompoundTypeByChemicalSpecies*, is known to be a specialization of the collection *ExistingStuffType*, via the transitive properties of *genls*. See Figure 3. Thus, by virtue of being an instance of *ChemicalCompoundTypeByChemicalSpecies*, *Water* is known to be an instance of *ExistingStuffType*. This illustrates that the decision tree for the mass noun distinction needs to consider not only *asserted*, but *inherited* collection membership.

[3]Atomic terms in the KB are called *constants*; there are also non-atomic terms (e.g., *(LeftFn Brain)*), for which definitional information are inferred automatically.

[4]As a formalization of commonsense knowledge, the Cyc KB contains hundreds of other, non-hierarchical predicates, used to express facts, and rules for concluding to facts, about objects in ontology.

[5]Unless otherwise noted, all examples are taken from OpenCyc version 0.7 (KB version 567 and system version 1.2594). An online version is available at www.opencyc.org/public_servers.

[6]Concept names in Cyc generally are self-explanatory, so descriptions are not included unless relevant to the discussion.

Collection: **PhysicalDevice**

Mt: ArtifactGVocabularyMt
isa: ExistingObjectType
genls: Artifact ComplexPhysicalObject
SolidTangibleProduct

Mt: ProductGMt
isa: ProductType

Figure 1: Type definition for *PhysicalDevice*, a prototypical denotatum term for count noun mappings.

Collection: **Water**

Mt: UniversalVocabularyMt
isa: ChemicalCompoundTypeByChemicalSpecies

Mt: UniversalVocabularyMt
genls: Individual

Mt: NaivePhysicsVocabularyMt
genls: Oxide

Figure 2: Type definition for *Water*, a prototypical denotatum term for mass noun mappings.

2.2 English lexicon

Natural language lexicons are integrated directly into the Cyc KB [Burns and Davis, 1999]. Though several lexicons have been partially integrated, the English lexicon is the only one to have been integrated to any notable level of completeness. The mapping from nouns to concepts is done using one of two general strategies, depending on whether the mapping is from a name or a common noun phrases. Binary predicates are used to effect name-to-term mappings, with the name represented as a string. For example,

(nameString HEBCompany "HEB")

A *denotational assertion* maps a phrase into a concept, usually a collection. The phrase is specified via a lexical word unit (i.e., lexeme concept) with optional string modifiers. The part of speech is specified via the one of Cyc's *SpeechPart* constants. Syntactic information, such as the wordform variants and their speech parts, is stored with the Cyc constant

ExistingStuffType
$\Uparrow_{genls}$
ChemicalSubstanceType
$\Uparrow_{genls}$
ChemicalCompoundTypeByChemicalSpecies

Figure 3: Generalization relations showing that *ChemicalCompoundTypeByChemicalSpecies* in a specialization of *ExistingStuffType*.

	Usage	
Predicate	OpenCyc	Cyc
denotation	3218	16589
compoundString	330	1958
multiWordString	1252	23670
headMedialString	192	871
total	4992	43088

Table 1: Denotational predicate usage in the Cyc English lexicon. This excludes microtheories for non-standard lexicalizations (e.g., *ComputereseLexicalMt*).

for the word unit. For example, *Device-TheWord*, the Cyc constant for the word 'device,' has a single syntactic mapping since the plural form is inferable:

Constant: Device-TheWord
Mt: GeneralEnglishMt
isa: EnglishWord
posForms: CountNoun
singular: "device"

The simplest type of denotational mapping associates a particular sense of a word with a concept via the *denotation* predicate (i.e., relation type). For example,

(denotation Device-Word CountNoun 0 PhysicalDevice)

This indicates that sense 0 of the count noun 'device' refers to *PhysicalDevice* via the associated wordforms "device" and "devices".

To account for phrasal mappings, three additional predicates are used, depending on the location of the headword in the phrase. These are *compoundString*, *headMedialString*, and *multiWordString* for phrases with the headword at the beginning, the middle, and the end, respectively. For example,

(compoundString Buy-TheWord (”down”) Verb Buy-Down)

This states that "buy down" refers to *BuyDown*, as well as "buys down, "buying down", and "bought down" as determined from the verb 'buy'.

Table 1 shows the frequency of the various predicates used in the denotational assertions, excluding lexicalizations that involve technical, informal or slang terms.[7] Of these, 9,739 have a *MassNoun* part of speech for the headword, compared to 20,936 for *CountNoun*. This subset of the denotational assertions forms the basis of the training data used in the mass versus count noun classifier, as discussed later.

3 Inference of lexicalization part of speech

3.1 General approach

Our method of inferring the part of speech for noun lexicalizations is to apply machine learning techniques over the lexical mappings from English words or phrases to Cyc terms. For each target denotatum term, the corresponding types and generalizations are extracted from the ontology. This includes terms for which the denotatum term is an instance or specialization, either explicitly asserted or inferable via transitivity. For simplicity, these are referred to as *ancestor terms*. The association between the lexicalization parts of speech and the common ancestor terms forms the basis for the criteria used in the lexicalization speech part classifier and the special case for the mass-count classifier.

There are several possibilities in mapping this information into a feature vector for use in machine learning algorithms. The most direct method is to have a binary feature for each possible ancestor term, but this requires thousands of features. To prune the list of potential features, frequency considerations can be applied, such as taking the most frequent terms that occur in type definition assertions. Alternatively, the training data can be analyzed to see which reference terms are most correlated with the classifications.

For simplicity, the frequency approach is used here. The most-frequent 256 atomic terms are selected, excluding internal constants flagged with the *quotedCollection* predicate (e.g., *PublicConstant*); half of these terms are taked from the *isa* assertions, and the other half from the *genls* assertions. These are referred to as the *reference terms*. For instance, *ObjectType* is a type for 21,042 of the denotation terms (out of 43,088 cases), compared to 19,643 for *StuffType*. These occur at ranks 10 and 11, so they are both included. In contrast, *HandTool* occurs only 226 times as a generalization term at rank 443, so it is pruned.

Given a training instance, such as a denotation from a word unit into a specific Cyc concept using a particular *SpeechPart* (e.g., *MassNoun* or a *CountNoun*), the feature specification is derived by determining all the ancestor terms of the denotatum term and converting this into a vector of occurrence indicators, one indicator per reference term. The part of speech serves as the classification variable. For example, consider the mapping of "heat production" to *HeatProductionProcess*.

(multiWordString ("heat") Produce-TheWord *MassNoun* *HeatProductionProcess*)

The type definition follows along with some of the ancestor terms inferred via transitivity (as given in the Cyc KB Browser).

Collection: *HeatProductionProcess*
Mt: NaivePhysicsVocabularyMt
isa: TemporalStuffType DefaultDisjointScriptType
genls: Emission

(isa HeatProductionProcess ?ARG2)
32 answers for ?ARG2 :
Collection ... StuffType ... TemporalStuffType Thing

(genls HeatProductionProcess ?ARG2)
22 answers for ?ARG2 :
Emission EnergyTransferEvent Event Event-Localized GeneralizedTransfer ... Thing TransferOut Translocation

It turns out that all of these except for *EnergyTransferEvent* are in the reference list. Therefore, the corresponding feature vector would have 1's in the 49 slots corresponding to the unique reference terms and 0's in the other 207 slots, along with *MassNoun* for the classification value.

[7] Cyc was adapted for use in the Hotbot web search engine and thus recognizes many colorful mass terms (e.g., "farm sex").

The example illustrates that some of the reference terms are not very relevant to the classification at hand (e.g., *Thing*). Advanced techniques could be used to address this, such as that used for collocation selection in word-sense disambiguation based on conditional probability. This is not done here, as it complicates the training process without significantly improving performance. The result is a table containing 30,675 feature vectors that forms the training data. Standard machine learning algorithms can then be used to induce the mass noun lexicalization criteria.

3.2 Sample criteria

We use decision trees for this classification. Part of the motivation is that the result is readily interpretable and can be incorporated directly by knowledge-based applications. Decision trees are induced in a process that recursively splits the training examples based on the feature that partitions the current set of examples to maximize the information gain [Witten and Frank, 1999]. This is commonly done by selecting the feature that minimizes the entropy of the distribution (i.e., yields least uniform distribution). Because the complete decision tree is over 300 lines long, just a few fragments are shown to give an idea of the criteria being considered in the count-mass classification.

```
if ObjectType and Event and CreationEvent then
    if AnimalActivity then
        CountNoun
    else
        MassNoun
```

This fragment indicates that creation events are generally lexicalized via count noun mappings when they represent animal activities. Otherwise, mass noun lexicalizations are used. An example of a concept inheriting from *AnimalActivity* is *MakingSomething*, with the count term “creation”. One not inheriting from *AnimalActivity* is *PhysicalSynthesis*, with the mass term “physical synthesis.”

```
if (not ObjectType) and (not Relation) and Agent-
Generic then
    MassNoun

if (not ObjectType) and Relation then
    CountNoun
```

The second rule fragment indicates that if both *ObjectType* and *Relation* are not ancestor terms for a concept, then the reference will use mass nouns for concepts that inherit from *Agent-Generic*. An example of this is *Dissatisfied*, referred to as “dissatisfaction”. The notion of generic agents might seem odd here, but emotional states in Cyc are restricted to agents. For concepts that are not typed as *ObjectType* but are typed as *Relation*, the reference will use count nouns. For example, any *UnitOfMeasure*, a specialization of *Relation*, is lexicalized using a count noun (e.g., “meter”).

3.3 Results for mass-count distinction

Table 2 shows the results of 10-fold cross validation for the mass-count classification. This was produced using the J48 algorithm in the Weka machine learning package [Witten and Frank, 1999]. This shows that the system achieves an accuracy of 90.5%, an improvement of 22.3 percentage points over the baseline of always selecting the most frequent case. Figure 4 for more details on the classification results, in particular the confusion matrix. The OpenCyc version of the classifier also performs well. This suggests that sufficient data is already available in OpenCyc to allow for good approximations for such classifications.

	OpenCyc	Cyc
Instances	2607	30676
Entropy	0.76	0.90
Baseline	78.3	68.2
Accuracy	87.5	*90.5*

Table 2: Mass-count classification over Cyc lexical mappings and using Cyc reference terms as features. *Instances* refers to size of the training data. *Baseline* selects most frequent case. *Accuracy* is average in the 10-fold cross validation.

	OpenCyc	Cyc
# Instances	3721	43089
# Classes	16	33
Entropy	1.95	2.11
Baseline	54.9	49.0
Accuracy	71.9	*73.3*

Table 3: General speech part classification using Cyc. *# Instances* is size of the training data. *Baseline* selects most frequent case. Instances refer to size of the training data. The classes are Cyc'S speech part category values. *Accuracy* is the average from the 10-fold cross validation.

3.4 Results for general speech part classification

The mass/count noun distinction can be viewed as a special case of speech part classification. Running the same classifier using the full set of speech part classes yields the results shown in Table 3. Here the overall result is not as high, but there is a similar improvement over the baseline. In terms of absolute accuracy it might seem that the system based on OpenCyc is doing nearly as well as the system based on full Cyc. This is somewhat misleading since the distribution of parts of speech is simpler in OpenCyc, as shown by the lower entropy value [Jurafsky and Martin, 2000]. Nonetheless, it suggests that sufficient data is already available in OpenCyc to allow for good approximations for part of speech inference. As usual, more data leads to better performance.

3.5 Extension to WordNet

The mass noun criteria based on the full Cyc KB requires access to the KB to be useful for incorporation in applications. The full KB is proprietary except for certain research purposes, so access to it might be difficult. However, the criteria induced over the Cyc KB can be carried over into WordNet by taking advantage of the WordNet mapping in the KB (covering a subset of WordNet version 1.6). In effect, this augments the WordNet lexicon with mass noun indicators, making it

```
=== Stratified cross-validation ===

Correctly Classified Instances       27763               90.504  %
Incorrectly Classified Instances      2913                9.496  %
Kappa statistic                          0.7737
K&B Relative Info Score            2108455.9597 %
K&B Information Score                19011.1535 bits      0.6197 bits/instance
Class complexity | order 0           27659.1802 bits      0.9017 bits/instance
Class complexity | scheme           186136.8792 bits      6.0678 bits/instance
Complexity improvement     (Sf)    -158477.699  bits     -5.1662 bits/instance
Mean absolute error                      0.142
Root mean squared error                  0.2778
Relative absolute error                 32.7544 %
Root relative squared error             59.666  %
Total Number of Instances            30676

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.958     0.209      0.908     0.958     0.932    CountNoun
  0.791     0.042      0.897     0.791     0.841    MassNoun

=== Confusion Matrix ===

     a     b   <-- classified as
 20055   881 |     a = CountNoun
  2032  7708 |     b = MassNoun
```

Figure 4: Detailed results for Cyc mass-count classifier using Weka's J48 system.

easier for applications such as *Yahoo!* to account for the distinction.

The Cyc-to-WordNet mapping includes over 8,000 of the synsets, with emphasis on the higher-level Cyc concepts. The mapping could be applied either to the final decision tree or to the feature table prior to classification. The latter is preferable, because the decision tree induction can then account for overly general mappings along with gaps in the mappings.

A separate classifier based on WordNet synsets is produced as follows: Each of the Cyc reference term features is replaced by a feature for the corresponding reference synset. Each of these binary features indicates whether the target denotatum synset is a specialization of the reference synset:

⟨target-synset, *has-ancestor-hypernym*, reference-synset⟩

Correspondence is established by first checking for an assertion directly linking the Cyc reference term to a WordNet synset. If that fails, there is a check for a linkage from one of the reference term's generalizations into WordNet. In cases where there are no such synsets, the feature will not be used. In cases where several reference terms correspond to the same synset, the features will be conflated.

Given the 256 reference terms used for the Cyc-based results (shown in Table 2), the process to establish correspondences yields 70 distinct features (due to 62 deletions and 124 conflations). Table 4 shows the results, indicating an accuracy of 86.3% in mass-noun classification, which is close to that when using the original features.

	OpenCyc	Cyc
Instances	3395	30675
Entropy	0.74	0.90
Unmapped accuracy	86.9	89.5
Baseline	79.2	68.2
Mapped accuracy	85.3	*86.3*

Table 4: Mass-count classification over Cyc lexical mappings using reference term features mapped into WordNet. *Baseline* selects most frequent case. *Unmapped accuracy* refers to results shown earlier. *Mapped accuracy* incorporates the WordNet mappings prior to training and classification (average of 10 trials).

The following is a simple fragment from the resulting decision tree:

```
if N03875475 then          {color, coloring}
   if N04496504 then       {kind, sort, form, variety}
      CountNoun
   else
      MassNoun
```

This shows that color terms are generally mass nouns unless referring to kinds of colors (e.g., different pigments). In terms of WordNet, since the corresponding synsets are disjoint (i.e., not related via a common hypernym), this entails that the mass noun lexicalization will always be preferred. In Cyc, the count noun usage only applies when concepts are lexicalized via multi-word phrases headed by "color" (e.g., *HumanSkinColor* as "skin color"). These concepts are not represented in WordNet, so this does not produce any conflicts.

4 Related work

We are unaware of other approaches to the automatic determination of the preferred lexicalization part of speech, using either statistical or traditional knowledge-based frameworks, nor for special case of the mass-count distinction (see [O'Hara *et al.*, 2003]). There has been much work in part-of-speech tagging [Church, 1988; Brill, 1994], which concentrates on the sequences of speech tags rather than the default tags. There has also been much work on the coercion of speech tags from one type to the other, especially with respect to mass and count noun conversions [Briscoe *et al.*, 1995]. And there has been some work addressing contextual interpretation, again with emphasis on mass term [Bunt, 1985]. In contrast, we address the creation of lexical mappings of mass terms into concepts, which can be viewed as precompiling mass noun preferences into the lexicon. In fact, this could serve as input into Bunt's process for mass noun interpretation.

Quirk et al. [1985] provide rough guidelines for whether nouns will be mass nouns or count nouns based on the type of the denotation term. For example, count nouns refer to individual countable entities whereas mass nouns refer to undifferentiated masses (or continua). They also provide information about which suffixes are indicative of the traditional grammatical categories. Huddleston and Pullum [2002] provide similar guidelines but do provide more details on the differences involved.

5 Conclusion

This paper shows that an accurate decision procedure (90.5%) for determining mass-count distinction in lexicalizations can be induced from the lexical mappings in the Cyc KB. The general case yields a classifier with promising performance (73.3%), considering that it is a much harder task, with over 30 categories to choose from. This relies upon semantic information, in particular Cyc's ontological types, in additional to syntactic information (e.g., headword morphology). Although the main approach incorporates Cyc's conceptual distinctions, the approach can be extended to non-Cycapplications via the WordNet mapping.

Future work will investigate how the classifiers can be generalized to for classifying word usages in context, rather than isolated words. This could complement existing part-of-speech taggers by allowing for more detailed tag types. We will also consider how to integrate the system with formal approaches to mass noun interpretation such as by Bunt [1985]. This will require work on deciding when pragmatic influences based on contextual clues should override the lexicon preferences.

This work has just scratched the surface of what can be learned from the massive amount of data in Cyc. Given the recent release of OpenCyc, we encourage others to investigate how information in the Cyc knowledge base can be exploited for use in intelligent applications.

Acknowledgements

TODO: This will be added after the blind review.

References

[Brill, 1994] Eric Brill. Transformational based part of speech tagging. In *Proc. XXX*, 1994.

[Briscoe *et al.*, 1995] Ted Briscoe, Ann Copestake, and Alex Lascarides. Blocking. In P. Saint-Dizier and E. Viegas, editors, *Computational Lexical Semantics*, pages 273–302. Cambridge University Press, Cambridge, 1995.

[Bunt, 1985] Harry C. Bunt. *Mass terms and model-theoretic semantics*. Cambridge University Press, Cambridge, 1985.

[Burns and Davis, 1999] Kathy J. Burns and Anthony B. Davis. Building and maintaining a semantically adequate lexicon using Cyc. In Evelyn Viegas, editor, *The Breadth and Depth of Semantic Lexicons*, pages 121–143. Kluwer, 1999.

[Church, 1988] Ken Church. The parts tagger. In *Proc. XXX*, 1988.

[Huddleston and Pullum, 2002] Rodney Huddleston and Geoffrey K. Pullum. *The Cambridge Grammar of the English Language*. Cambridge University Press, Cambridge, 2002.

[Jurafsky and Martin, 2000] Daniel Jurafsky and James H. Martin. *Speech and Language Processing*. Prentice Hall, Upper Saddle River, New Jersey, 2000.

[Lenat, 1995] D. B. Lenat. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11), 1995.

[O'Hara *et al.*, 2003] Tom O'Hara, Nancy Salay, Michael Witbrock, Dave Schneider, Bjoern Aldag, Stefano Bertolo, Kathy Panton, Fritz Lehmann, Matt Smith, David Baxter, Jon Curtis, and Peter Wagner. Inducing criteria for mass noun lexical mappings using the Cyc KB, and its extension to WordNet. In *Proc. Fifth International Workshop on Computational Semantics (IWCS-5)*, 2003.

[Quirk *et al.*, 1985] R. Quirk, S. Greenbaum, G. Leech, and J. Svartvick. *A Comprehensive Grammar of the English Language*. Longman, 1985.

[Witten and Frank, 1999] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, 1999.