Corpus-based acquisition of head noun countability features

Lane O.B. Schwartz
Churchill College
Cambridge CB3 0DS

los20@cam.ac.uk

A thesis submitted
to the University of Cambridge
in fulfilment of the requirements for
the degree of Master of Philosophy
in Computer Speech, Text and Internet Technology.

July 2002

Abstract

In recent years, significant advances have been made in the use of corpora as tools in language processing. Lexical acquisiton techinques have been somewhat successful in learning verb subcategorization information. Yet much of the other information available from corpora has not been harnessed. The countability property of nouns is one property that would be useful to acquire. Such information could help in word sense disambiguation, in determining appropriate determiners during generation (especially in the case of machine translation), and as a lexicographic resource during dictionary construction.

Existing lexical resources which include countability features of nouns have been created largely by hand. Manual tagging of noun countability is expensive in terms of time and labor. It is difficult to extend such resources as new terminology emerges.

This thesis presents a method of automatically acquiring countability properties of head nouns. This information is gathered from a part-of-speech tagged corpus, specifically the British National Corpus (BNC). Basic noun phrase chunking is performed on the corpus to obtain head nouns and their accompanying determiner, if any. High-reliability grammatical cues are used to automatically tag head noun tokens as either count or non-count. This method relies heavily on the grammatical role determiners play in the countability of head nouns.

This thesis demonstrates that the method used is both grammatically sound and successful, showing an improvement over the baseline. The automatic countability tagger can correctly tag nouns with countability in up to 87% of noun phrases.

Acknowledgements

I wish to extend my sincere thanks to my advisor, Dr. Ann Copestake, for help and guidance during this project and throughout my time at Cambridge.

Thank you to all those who have helped to develop free software. I am especially grateful to the authors of the GNU Text Utilities. These tools have proved to be an invaluable aid over the course of this project.

And most of all, thank you to my wife, Sarah, for crossing the Atlantic and for putting up with me as I researched and prepared this thesis.

This thesis was written on GNU Emacs, typeset with LaTeX and rendered for print with Radical Eye Software's dvips.

Declaration of original content

The work presented in this thesis is, to the best of my knowledge and belief, original and my own work, except as acknowledged in the text. The material has not been submitted, either in whole or in part, for a degree at this or any other university.

Lane O.B. Schwartz

Copyright

Contents

Chapter 1

Introduction

In recent years, significant advances have been made in the use of corpora as tools in language processing. Lexical acquisiton techinques have been somewhat successful in learning verb subcategorization information. Yet much of the other information available from corpora has not been harnessed.

The countability property of nouns is one property that would be useful to acquire.

Such information could help in word sense disambiguation, in determining appropriate determiners during generation (especially in the case of machine translation), and as a lexicographic resource during dictionary construction. Accurate countability information could help improve word sense disambiguation in certain cases. Such information can also assist lexicographers in dictionary construction. Such an automatic countability tagger (ACT) could greatly reduce the need for manual entry of countability features into lexicons. Learners of English as a second language may find a lexicon where nouns are marked with countability attributes extremely valuable. But, perhaps the most important use for countability information is in machine translation.

This thesis presents a novel tagger capable of automatically tagging the nouns in a part-of-speech tagged corpus with countability tags. I show that the automatic countability tagger presented here is both grammatically sound and useful, showing improvements over the baseline of always choosing the most common countability feature. I show that the automatic countability tagger can correctly tag nouns with countability in up to 87% of noun phrases.

This work is presented as follows. Chapter 2 examines the nature of countability and number in English. Chapter 3 looks at the lexical resources used in tagging and in evaluation. In chapter 4 the algorithm for basic automatic countability tagging is presented. Chapter 5 evaluates the results of the automatic countability tagger against the countability data in the COMLEX and NTT lexicons. Finally, chapter 6 explores several extensions to the basic tagger, including an expanded algorithm.

Chapter 2

Background

A primary aim of this project is the production of an automatic countability tagger for English. It is therefore crucial to establish a working definition of countability in English.

The chapter will be presented as follows: Section 2.1 presents the basic features of countability; the most important metonymic conversion rules that affect countability are also considered. Section 2.2 discusses the two most important factors that influence countability: determiners and number. Specifically, this section explores the cues given by determiners and by a head noun's number feature which can disclose the noun's countability feature. Section 2.3 continues in this vein, exploring additional factors which can reveal more information about a head noun's countability.

2.1 Basics of countability in English

Countability is an important feature of nouns in English. A given instance of a noun may be either countable, non-countable, or ambiguous with respect to countability. A singular noun that occurs in a countable context refers to an individual bounded entity. Likewise, plural countable nouns refer to a set of individual entities, each of which is conceptually bounded. On the other hand, non-count nouns refer to concepts or unbounded substances that cannot (by default) be individuated and counted.

[1] i. The *disc* contained vital information.
ii. *Helium* escaped from the balloon.

Some nouns, such as those above, occur almost exclusively as either countable or non-countable. It is difficult to find examples where *disc* functions in a non-countable capacity; likewise, *helium* can not be easily used in a countable context. Many nouns, however, can act in either a countable or a non-countable capacity.

The nouns in example [2] below are by no means exceptional in having both countable and non-countable senses. Huddleston and Pullum (2002) propose that this systematic dual usage of nouns as both countable and non-countable is usually the result of polysemy rather than homonymy. That is, the uses of *stout* in [2.iii] and [2.iv] represent

[2] i. The wind blew a *paper* away from the newsstand.
ii. Processed wood is used to manufacture *paper*.
iii. I'll have one *stout*, please.
iv. Guinness is the most well known brand of *stout*.
v. The Bill of Rights guarantees all citizens certain *freedoms*.
vi. The slave managed to buy his *freedom*.

two distinct senses of the same lexical item rather than two distinct lexical items. This systematic usage can often be predicted. Nouns which ordinarily are non-countable may become countable through metonymic sense shifts that Quirk et al. (1985) call 'reclassification.' Countable nouns also may become non-countable through this process. Huddleston and Pullum (2002) offer six regular types of reclassification:

[3] i. Substance [non-count] ⇒ serving of the substance [count]
[non-count] — *There is water in the basement.*
[count] — *I need to buy another water.*

ii. Substance [non-count] ⇒ kind/variety of the substance [count]
[non-count] — *I enjoy baking bread.*
[count] — *My favorite breads are sourdough and rye.*

iii. Animal [count] ⇒ food made from the animal [non-count]
[count] — *We saw a moose on that hill yesterday.*
[non-count] — *The only meat in the freezer is moose.*

iv. Abstract concept [non-count] ⇒ event instantiating the concept [count]
[non-count] — *Keep an eye out for procedural irregularity.*
[count] — *Irregularities in Enron's accounting have shocked the world.*

v. Abstract concept [non-count] ⇒ result of the concept [count]
[non-count] — *Creation is better than destruction.*
[count] — *He was proud of his creation.*

vi. Item [count] ⇒ (ground) substance made from the item [non-count]
[count] — *A deer raced across the field.*
[non-count] — *There was deer on the road and the car after the accident.*

The above metonymic rules are generally productive. Huddleston and Pullum (2002) demonstrate the produtivity of [3.i] by noting that the sense conversion from a drink to a serving of a drink holds even in the case of commercial drinks with brand names, such as Ovaltine.

However, the rules of reclassification are not without exception. Briscoe and Copestake (1991), in their consideration of sense extensions, note that certain metonymic extensions are blocked, specifically when synonymous lexemes already exist. [3.iii] would suggest that the count noun *cow* (the animal) may be reclassified into the non-count noun *cow* (the meat from the animal); however, the existance of a separate lexeme, *beef*, that is synonymous with the proposed extension, usually blocks the usage of *cow* as non-count.

Likewise, Huddleston and Pullum (2002) note that [3.iv], while generally produc-

tive, is not entirely predictable. They contrast the usage of *injustice*, which follows the pattern described in [3.iv], with *harm*, which does not.

2.2 Factors influencing countability

Bond (2001) observed: "Countability in English involves two phenomena: whether a noun can be both singular and plural, and what kind of dependents it can take."

This observation is worth careful consideration. While native speakers of English may have little difficulty differentiating between count and non-count instances of nouns merely on the basis of their intuition on boundedness and individuation (see 2.1), computer programs without the advantage of world knowledge, and even non-native speakers with a different conception of boundedness often cannot make the judgement of what English noun tokens are countable and which are not. For this reason it is worthwhile to examine the syntax of noun phrases (NPs) to determine the extent to which noun countability can be emperically determined.

2.2.1 Determiners

Nouns do not normally appear in isolation, but rather within the context of a noun phrase in a sentence. It can be shown that the determiner of a noun phrase is closely linked to the countability of the head noun of the NP. The determiner can thus be of primary significance in automatically establishing the countability feature of the head noun.

The ties between the determiner of an NP and the countability of the NP's head noun discussed below apply primarily (but not exclusively) to determiners which modify singular common nouns. This especially applies to ϕ, which denotes the null determiner. While plural common nouns with no determiner commonly are countable, singular common nouns with no determiner are nearly always non-countable.

Huddleston and Pullum (2002), Quirk et al. (1985), Huddleston (1988) and numerous other grammarians have all noted the relationship between the presence of certain determiners in an NP and the countability feature of the corresponding head nouns. Specifically, there are certain determiners which always or nearly always occur in NPs with countable head nouns. Likewise, other determiners are found almost exclusively in NPs with non-countable head nouns.

[4] i. Countable — *a, an, another, each, every, either, neither, one*
ii. Non-countable — *enough, insufficient, less, little, more, most, much, overmuch, such, sufficient,* ϕ

Other determiners are of significantly less use in determining a head noun's countability. Several of these less indicative determiners are listed below:

[5] iii. Ambiguous — *any, his, her, its, my, no, that, the, this, your*

2.2.2 Number

The second point of consideration is the number feature of the head noun in question. In English, number is a binary feature — possible values are singular and plural.[1] Unlike countability, the number feature of nouns is generally well documented in machine readable dictionaries (MRDs). Additionally, many parsers mark nouns with a number feature.

Examination reveals that a noun's number feature directly relates to its countability. With few exceptions, plural nouns occur only in countable contexts. However, Huddleston (1988) notes that in cases where a plural noun does not have an established singular form (ex. *earnings*, *dregs*), the plural noun is in fact non-countable.[2]

So, the existance of plural and singular forms of a noun help to indicate the noun's countability feature.

2.3 Additional considerations

Of the surface factors that can be used to predict a head noun's countability feature, the two described above are the most important. The determiners listed in example [4] of 2.2.1 can be used to readily predict countability in noun phrases where they occur. Likewise, the value(s) that a head noun's number feature may take is a good indicator of its countability feature. Specifically, plural nouns that can also be singular are usually countable; singular nouns with no plural form, and plural nouns with no singular form, tend to be non-countable.

However, using the cues given above, countability remains ambiguous for numerous nouns that will be encountered in real text. This section considers several of the more complex situations encountered when trying to disambiguate in these situations.

2.3.1 Bare nouns

Bare head nouns in English are defined as the head nouns of NPs which have the null determiner (ϕ). The majority of these bare nouns are either plural or proper. The countability feature of bare plurals can generally be discerned using the criteria summarized above. The concept of countability does not easily apply to proper nouns; this class of nouns thus falls outside the scope of this work. However, bare head nouns which are common singular nouns require special consideration.

The behavior of bare nouns is not consistent across languages. Schmitt and Munn (1998) note that most Romance languages generally do not allow bare plurals and bare mass nouns to appear in argument position. On the other hand, the authors observe that Brazilian Portuguese not only permits bare plurals and bare mass nouns, but also allows bare singular count nouns. Like Brazilian Portuguese, English generally allows bare plurals and bare mass nouns in argument position. Bare count singular nouns in English will now be examined:

[1] Ambiguity in this feature is also possible for a given noun token.
Example of ambiguity in number: *The fish swam up the stream.*

[2] See Appendix A for a discussion of bipartite nouns in this context.

According to [4.ii], the presence of the null determiner (ϕ) in an NP headed by a common singular nouns should act as a reliable indicator that the head noun is non-countable. Indeed, Quirk et al. (1985) note that "Singular count nouns cannot in general head an NP without a determiner." Below are typical examples of (non-countable) bare common nouns:

[6] i. *Gold* is very valuable.
ii. The majority of the Earth's surface is covered in *water*.
iii. In today's world, *information* has become an important commodity.

The situation in real usage is not so simple. In certain constrained situations, English does permit bare count singular nouns. Huddleston and Pullum (2002) observe that this unusual phenomenon is permissible when certain nouns act as a predicative complement. Singular count nouns still cannot occur bare in subject or object positions.

[7] i. During the operation, Dick Cheney was *president*.
ii. **President* \ *The president* called for a cabinet meeting.
iii. I saw **president* \ *the president* of China visiting the pyramids.

Thus, any system which directly implements the cues in [4.ii] will incorrectly tag situations such as [7.i] as non-countable.

2.3.2 Mass nouns as plurals

The number feature of nouns can be quite useful in determining the appropriate countability feature (see section 2.2.2). Observation shows that singular nouns with no plural form are usually non-countable, as are plural nouns with no singular form. On the other hand, nouns that have both singular and plural forms tend to be countable. But, as is so often the case, the true situation has exceptions.

The countability reclassification rule in [3.i] allows nouns which normally denote non-countable substances to refer to a serving of the substance. Likewise, [3.ii] allows substance nouns to denote a kind or a variety of the substance. In either case, the conversion forces a noun that is usually non-countable to act as a countable noun.

This conversion by itself does not interfere with the claims about number and countability made above. However, Kay (1999) notes that once this conversion has occurred, nouns (which are normally non-countable) may be used in plural form.

[8] i. [non-countable] *Milk* is rich in calcium.
ii. [countable] One *milk* that I can't tolerate is goat's milk.
iii. [countable plural] She is allergic to all non-soy-based *milks*.

Both [3.i] and [3.ii] are highly productive rules; non-countable substance nouns can often be coerced into countable readings denoting servings, portions, kinds, or varieties of the substance. In a large corpus, the coerced plural form of many non-countable nouns may well occur. If the above cues are used as guidelines (and nouns that have both a singular and a plural form are considered countable), then the existance of a plural form for a non-countable noun will trigger some errors (such that the noun is misclassified as countable).

2.3.3 Complex determiners

So far, most attention has been paid to simple determiners. Some of these determiners clearly signal a countable head noun (*a, an, each, either...*). Other simple determiners mark a non-countable head noun (*enough, less, most, much...*). Many simple determiners, however, allow both countable and non-countable head nouns (*her, his, its, the...*).

Complex determiners also may have a part to play in countability. Complex determiners are multi-word expressions which act in a similar role to simple, single word determiners. Specifically, complex determiners which denote measurement can affect head noun countability; van Eijck (1991) notes that these complex determiners can occur with countable and non-countable head nouns:

[9] i. This recipe calls for two cups of *flour*.
ii. This recipe calls for two cups of *raisins*.

The above author rightly notes that complex determiners of the form *# MEASURE of* cannot accurately predict head noun countability. However, observation reveals that when these complex determiners occur with a plural head noun, that head noun is nearly always countable; conversely, when the head noun in these cases is singular, that noun tends to be non-countable.

Chapter 3

Resources

This chapter considers the various lexical resources used in this project. For each resource, I examine the format of the resource and the extent to which countability attributes are marked in the resource.

Section 3.1 looks at the corpus used in the experiments — the British National Corpus. Section 3.2 considers COMLEX, the first lexicon used to evaluate the automatic countability tagger's results. The second evaluation resource, NTT's countability-tagged lexicon, is discussed in section 3.3. Section 3.4 examines the usefulness of these lexicons as evaluation standards by evaluating COMLEX against the NTT lexicon.

3.1 Corpus: BNC

The British National Corpus (BNC) is an important collection of spoken and written examples of British English. The corpus consists of more than 100 million words. The material in the BNC includes excerpts from fiction and non-fiction, as well as transcriptions of spoken dialogue. See (Warwick, 1997) for more details about content of the corpus, as well as details of its construction.

The BNC is divided into sentences. This works divides the corpus into two parts. Ten percent of the sentences are reserved for use as a test corpus. The remaining 90% of the sentences from the corpus are used in the experiments that follow. Each word (including punctuation) in the corpus is tagged with a part-of-speech (POS) marker. No countability information is included in the BNC. A typical sentence is listed below in [10]. See (Leech, 1997) for details on the tagset used in the corpus.

3.1.1 Noun phrase chunker

A basic noun phrase chunker was used in this project to extract noun phrases from the BNC sentences. The chunker extracts noun phrases based on the following formula: an NP consists of a determiner (or ϕ) followed by optional adverb(s), followed by optional

[10] ^^
This DT0
virus NN1
affects VVZ
the AT0
body NN1
's POS
defence NN1
system NN1
so<blank>that CJS
it PNP
can VM0
not XX0
fight VVI
infection NN1
. .

adjective(s), followed by optional noun(s), ending with a single noun. See Appendix A for the NP chunker source code.

In some cases, the tagging process used to create of the BNC was unable to fully disambiguate a word in the corpus. These words are marked with multiple part-of-speech tags to denote the ambiguity. For simplicity, the chunker disgards all noun phrases that contain any ambiguously tagged words.

It should also be noted that the NP chunker makes a significant simplifying assumption. Namely, the chunker only allows one determiner in each noun phrase. Huddleston (1988) describes three seperate classes of determiners; a noun phrase may have up to one determiner of each type. The chunker used here ignores all but the final determiner in cases where an NP has more than one determiner.

Additionally, the chunker disgards any NP where the head noun is not tagged as a singular common noun. Plurals and proper nouns are thus excluded from direct consideration in this work. However, due to tagging errors in the BNC, a number of plural and proper nouns are incorrectly tagged as singular common nouns. The presence of these grammatical tagging errors is a potential source of error during automatic countability tagging, as the automatic countability tagger is designed to work with the expected output of the NP chunker (a determiner followed by a singular common head noun — see 3.1.2 below).

3.1.2 Head nouns and determiners

Using the NP chunker, slightly over 16.27 million noun phrases were extracted from the British National Corpus (BNC). The head noun and the determiner (or ϕ, for NPs with a null determiner) for each NP were then made available to the automatic countability tagger. No words in an NP other than the head noun and determiner are used by the automatic countability tagger; hence, these extra words are discarded by the chunker.

Below are the determiners and head nouns extracted from [10] by the NP chunker. Note that — signifies a null determiner (ϕ). (For the remainder of this work, references

to the BNC data refer to the format below —
DETERMINER HEAD_NOUN

[11] This virus
the body
— system
— infection

As evident above, this method of NP chunking is not without error. A more advanced chunker or parser would realize that *the body 's defense system* is a single NP, rather than two; a more appropriate chunking result follows:

[12] This virus
the system
— infection

Although not error-free, the method used here to NP chunk the BNC is fast and efficient, providing a good first step towards countability tagging by extracting over 16.27 million head nouns and the corresponding determiners from the BNC.

3.2 Lexicon: COMLEX

The Common Lexicon (COMLEX), developed at New York University and presented in (Grisham et al., 1994), is a lexicon for English comprising approximately 38,000 word entries. Of these, 21,125 entries are nouns. The resource was constructed largely by manually entering features to the word entries in the lexicon. Numerous features are marked for each lexical entry, including countability. These features are described entensively in (Wolff et al., 1994).

The possible values of the countability feature, marked only in nouns, are of particular interest. COMLEX encodes the concepts of countable and non-countable nouns through three possible tags: COUNTABLE, NCOLLECTIVE, and AGGREGATE. Two additional values are also used below. Nouns tagged as both COUNTABLE and NCOLLECTIVE are marked below as COUNT_NCOLL. Finally, a fifth possible value, AMBIGUOUS, is inferred if none of the other tags are present.

3.2.1 COUNTABLE

Nouns marked as COUNTABLE in COMLEX are considered to be unambiguously countable. According to Wolff et al. (1994), a noun is tagged as COUNTABLE if it must be preceded by a determiner; nouns such as *chicken* and *fish* are excluded from the COUNTABLE category. Although these nouns require a determiner when used in their countable sense, they can too easily be used in a non-countable sense to be marked with this tag.

COMLEX additionally considers two proper subsets of COUNTABLE nouns. Nouns marked PREPNOUN can occur without a preceding determiner when preceded by a specific preposition (or prepositions). Likewise, nouns marked PREDNOUN can occur without a preceding determiner when they occur as the object of a certain class

of verbs. See 2.3.1 and Wolff et al. (1994) pp. 11-12 for more information on these two subclasses. For purposes of evaluation against COMLEX, this thesis considers any nouns with the features PREDNOUN or PREPNOUN to simply be COUNTABLE nouns, without taking into account the special circumstances that allow nouns in these categories occur without a preceding determiner.

[13] [COUNTABLE] She cut her *ankle* shaving this morning.
[COUNTABLE] I would like an *apple*.
[PREPNOUN] She dislikes travelling by *plane*.
[PREDNOUN] He became *governor* on January 1.

3.2.2 NCOLLECTIVE and COUNT_NCOLL

Nouns marked NCOLLECTIVE indicate substances or concepts that are by default non-countable. Note that this category does not include nouns denoting groups of people. Additionally, a small (27) set of nouns in COMLEX are marked as both COUNTABLE and NCOLLECTIVE; in the results that follow these nouns are noted as COUNT_NCOLL. Nouns in this category are commonly used in both countable and non-countable contexts.

[14] [NCOLLECTIVE] Growing *evidence* pointed to his guilt.
[NCOLLECTIVE] The *ice* extended out from the shore for several miles.
[COUNT_NCOLL] *Liquid* flowed from the side of the punctured barrel.
[COUNT_NCOLL] A dark green *liquid* was visible inside the lava lamp.

3.2.3 AGGREGATE

A small (155) set of nouns in COMLEX are marked as AGGREGATE. AGGREGATE nouns are collective — that is, when the subject of a sentence, these nouns can act as either singular or plural. See Copestake (1995) and Wolff et al. (1994) for more information on this class of nouns.

In general, collective nouns of this type are thought to be countable. If this were true, then AGGREGATE nouns should be grouped with the COUNTABLE nouns. Indeed, a majority (109) of these nouns are tagged both as AGGREGATE and as COUNTABLE.[1]

[15] The *union* is petitioning for better pay.
The *union* are petitioning for better pay.

However, nearly 30% of the listed AGGREGATE nouns are not marked as being COUNTABLE. A review of these nouns presents two main reasons why these remaining nouns are not considered to be COUNTABLE.

Many of the nouns are in fact grammatically countable in most cases, but still are not listed as COUNTABLE. In some cases (*buffalo*, *fish*) this may be because the plural form of the noun is lexically identical as the singular, and so can occur without

[1] As mentioned above, COMLEX allows multiple countability features to be set for each noun.

a determiner in the plural form. (For a noun to be COUNTABLE, it must require a determiner.) In other cases (*family, parliament*) I can only postulate that certain nouns were not considered COUNTABLE because they can be used to mean an (non-countable) conceptual institution as well as a (countable) group of people.

Other nouns listed as AGGREGATE are in fact grammatically non-countable. These nouns can take both singular or plural agreement, but are actually non-countable (*acoustics, artillery*).

While many AGGREGATE nouns are also COUNTABLE, no distinction is made in COMLEX to distinguish between the two types of (non COUNTABLE) AGGREGATE nouns given above. The AGGREGATE feature in COMLEX therefore does not offer any disambiguation of noun countability beyond that provided by the COUNTABLE and NCOLLECTIVE features.

3.2.4 AMBIGUOUS

A large number of noun entries in COMLEX are tagged as neither as COUNTABLE nor as NCOLLECTIVE. COMLEX provides no solid basis for establishing the countability of this set of nouns. In the experiments and results that follow, this set of nouns is labelled AMBIGUOUS.

3.3 Lexicon: NTT

An additional resource used in evaluation was a list of nouns marked with countablity features, created by the Nippon Telegraph and Telephone (NTT) Communication Science Laboratiories. This resource provided countability features for approximately 35,000 distinct nouns.

The NTT data contains a richer set of possible countability features than COMLEX does. These values do not describe a noun's absolute countablity feature; rather, they designate a countability preferenece, recognizing that most nouns can be used in both countable or non-countable contexts. Fully countable (CO) nouns nearly always appear in countable contexts. Fully uncountable (UC) nouns, on the other hand, are nearly always non-countable. Strongly countable (CB) nouns tend to function as countable, while weakly countable (UB) nouns are non-countable by default. A small number of nouns in the data set are uncountably plural (UP); that is, these nouns are plural with no singular form. No countability preference is recorded for nouns marked (NN). For more detailed discussion of noun countability preferences see (Bond, 2001).

3.4 Resource evaluation: COMLEX vs NTT

Before any results of an automatic countability tagger are tested against the evaluation lexicons, it is useful to consider the lexicons themselves. Specifically, how well do the countability attributes for nouns in one resource compare to the countability attributes for the same nouns in the other resource?

This section considers how closely the countability features of nouns in COMLEX correspond to the same nouns in the NTT data. To perform this evaluation, each noun entry in COMLEX was looked up in the NTT lexicon. Of the 21,225 nouns in COMLEX, only 9,587 had a corresponding entry in the NTT lexicon. The comparison below considers only those nouns common to both sets of data.

3.4.1 Comparison methods

Entries in COMLEX were compared to the NTT data using three different methods. This was nescessary because of the different structure of the two resources. COMLEX was created as a general computational linguistic resource for English. The vast majority of nouns in COMLEX have only one countability feature: COUNTABLE, NCOLLECTIVE, or AMBIGUOUS.[2] A small number of nouns have two features listed (COUNTABLE and NCOLLECTIVE); this can, however, be treated as a single (joint) feature – COUNT_NCOLL. An entry from COMLEX would have the following form for English words *X*, *Y*, and *Z*:

[16] X COUNTABLE
Y NCOLLECTIVE
Z AMBIGUOUS

The NTT lexicon, on the other hand, was created for use in NTT's ALT-J/E Japanese-to-English machine translation system (Ikehara et al, 1991), (Bond, 2001). Hence, each noun may have multiple entries in the lexicon, corresponding to different Japanese translations of the same English word. Each entry lists one countability value for the word. For example, English word *X* may have five entries in the NTT lexicon: one entry with countability CO, three with countability CB, and one entry with countability UB.

[17] X CO
X CB
X CB
X CB
X UB

In order to compare the countability of COMLEX noun entries against the countability of corresponding entries in the NTT lexicon, a method is needed to map each noun from COMLEX to one of the entries for that noun in the NTT lexicon. In the above example, this would determine whether word *X* with countability COUNTABLE is paired with the entry in the NTT lexicon with countability CO, or with CB, or with UB. Three methods for performing this task are used.

3.4.2 Method 1

This method may be considered the baseline. Each noun entry from COMLEX is looked up in the NTT lexicon. All countability values listed in the NTT lexicon for a

[2]The AGGREGATE feature in COMLEX provides no additionally useful countability information, and so is not considered here.

given noun are considered to be correct. Duplicate entries in the NTT lexicon (for a given countability value) were removed, but nouns were still allowed to have multiple entries corresponding to multiple countability values. So, using the example above, word *X* from COMLEX would match three entries from the NTT lexicon:

[18] X CO COUNTABLE
X CB COUNTABLE
X UB COUNTABLE

The results of this evaluation method are listed below. The first column lists the countability values that nouns in COMLEX may have (COUNT = COUNTABLE, NCOLL = NCOLLECTIVE, AMBIG = AMBIGUOUS, N_C = both COUNTABLE and NCOLLECTIVE). Column 2 lists the number of times a word from COMLEX (that had a particular countability value) matched with an entry from the NTT lexicon. The remaining columns list the percentage of times these matches occured with each of the countability values from the NTT data.

Method 1	Matches	CO	CB	UB	UC	UP	NN
COUNT	5825	87.74%	1.72%	1.60%	6.18%	0.55%	2.21%
NCOLL	393	23.66%	4.07%	13.74%	54.20%	0.51%	3.82%
C_N	15	53.33%	6.67%	6.67%	26.67%	0.00%	6.67%
AMBIG	5359	33.87%	7.43%	7.71%	48.65%	0.69%	1.66%

So, interpreting the above results, words marked as COUNTABLE in COMLEX matched with a corresponding entry in the NTT lexicon 5825 times. (Note that because of the nature of this method, an individual noun from COMLEX may match up to 6 entries in the NTT lexicon[3]; one match will occur for each countability value that the noun occurs with in the NTT lexicon.) Of the total matches of COUNTABLE COMLEX nouns against entries in the NTT lexicon, 87.74% were with entries marked CO in the NTT lexicon.

The above results show widespread (though by no mean total) agreement between the two lexicons. The intersections of COUNT with UC and NCOLL with CO are the most crucial. These show the extent to which the resources directly contradict each other. These values are reasonably low. Likewise, the direct agreement values are encouraging. Entries for COUNTABLE nouns in COMLEX matched some sort of countable noun entry (either CO or CB) 89.46% of the time. Agreement for total non-countable entries (NCOLLECTIVE compared to UB and UC) was lower, at 67.95%.

3.4.3 Method 2

Method 2 contrasts with method 1 by requiring that each noun from COMLEX should match at most one entry in the NTT lexicon. When a noun in the NTT lexicon has multiple entries, this method requires that the countability feature with the majority (or plurality) of entries is selected as the only match for the corresponding COMLEX entry.

[3]This would happen if the NTT lexicon lists entries for a word with all 6 possible countability values: CO, CB, UB, UC, UP, and NN

Using this method, the 5 NTT lexicon entries for word *X* (see [17]) would be reduced to a single entry that would match the entry for *X* in COMLEX:

[19] X CB COUNTABLE

The results of evaluating all nouns from COMLEX against the NTT lexicon using this method are listed below:

Method 2	Matches	CO	CB	UB	UC	UP	NN
COUNT	5433	92.67%	1.69%	0.75%	3.59%	0.26%	1.03%
NCOLL	335	20.00%	4.48%	14.33%	58.21%	0.00%	2.99%
C_N	13	61.54%	7.69%	7.69%	23.08%	0.00%	0.00%
AMBIG	4141	29.58%	7.94%	7.46%	53.54%	0.60%	0.87%

Note that the number of word matches listed in the table above is less than the corresponding values from the results for method 1. This is because method 1 allows a single entry from COMLEX to find multiple matches in the NTT lexicon; method 2 only allows a single match per entry in COMLEX — hence the lower values in the Matches column.

Agreement between the lexicons was better using this method than method 1. Total countability agreement (COUNTABLE matches CO or CB) rose from 89.46% to 94.36%. The corresponding values for non-countable nouns (NCOLLECTIVE matches UB or UC) also rose, from 67.95% to 72.54%. Direct contradiction between resources fell — from 6.18% to 3.59% (COUNTABLE matches UC) and from 23.66% to 20.00% (NCOLLECTIVE matches COUNTABLE).

3.4.4 Method 3

Method 3 carries the insights from method 2 one step further. This method collapses the NTT countability categories of CO and CB together into a single “countable” category. Similarly, UB and UC are collapsed into a single “non-countable” category. After the categories are folded together, the same majority method as described in refMethod2 is used.

This method seeks to correct situations where, for example, a noun Y in the NTT lexicon has a plurality of entries marked as UB, but the total entries of CO and CB outnumber the UB entries. The hypothetical entries from COMLEX and the NTT lexicon are listed below:

[20] Y COUNTABLE

Under method 2, the UB entry would be selected as the match, leading to a contradiction between resources. In this method however, the combined CO-CB entry is choosen, leading to agreement.

The results for method 3 are listed below:

[21] Y CO
Y CO
Y CB
Y CB
Y UB
Y UB
Y UB

Method 3	Matches	CO,CB	UC,UB	UP	NN
COUNT	5433	93.91%	4.77%	0.29%	1.03%
NCOLL	335	23.88%	74.63%	0.00%	1.49%
COUNT_NCOLL	13	69.23%	30.77%	0.00%	0.00%
AMBIG	4141	35.47%	63.34%	0.63%	0.56%

Direct agreement (NCOLLECTIVE matches UC,UB) rose from the method 2 level of 72.54% to 74.63%. However, direct agreement (COUNTABLE matches CO,CB) fell slightly from the method 2 level of 94.36% to 93.91%. Both measures, however, are still well above the baseline of method 1.

Direct contradiction can not be measured in this method, as there is no longer a way to compare NCOLLECTIVE matches to just CO or COUNTABLE matches to just UC. So, the general contradiction levels will be examined. Using method 3 the level of (COUNTABLE matches UC,UB) stands at 4.77%; this is above the 4.34% level from method 2 but below the 7.78% level from method 1. General contradiction levels for (NCOLLECTIVE matches CO,CB) are 23.88% for method 3, a fall from the 24.48% of method 2 and the 27.73% of method 1.

3.4.5 Evaluation summary

In order to make a meaningful evaluation of any automatic system against (largely) hand-constructed lexicons such as COMLEX and the NTT lexicon, it is necessary to establish an upper evaluation bound for comparison. The results in 3.4.2, 3.4.3, and 3.4.4 give just such a bound. These results show the level at which two manually constructed lexicons agree with each other on countability of nouns in the lexicons.

It is surprising that such a large number of nouns from COMLEX do not appear in the NTT lexicon. COMLEX lists 21,225 nouns. The NTT lexicon contains over 53,000 distinct nouns. Yet only 9,587 noun entries from COMLEX (45%) had a corresponding entry in the NTT lexicon.

For those nouns which did occur in both lexicons, the results demonstrate that methods 2 and 3 show a higher level of inter-lexicon agreement than method 1. However, there is no clear case for asserting that either method 2 or method 3 reached an overall higher level of agreement than the other.

Chapter 4

Automatic Countability Tagger

This chapter presents the heart of this project: an automatic countability tagger (ACT). The grammatical framework for the ACT is based heavily on the topics covered in Chapter 2. Details on the corpus and lexical resources used by ACT in this work are listed in Chapter 3.

This chapter is presented as follows. First, I will explore the background behind the ACT, paying particular attention to the inspiration provided by previous work in automatic acquisition of verb subcategorization frames, especially Brent (1993). Next, I will detail the general level of pre-processing applied to the British National Corpus. Finally, I will consider the primary algorithm used to automatically tag nouns from the BNC with countability features.

4.1 Inspiration

Acquisition of lexical features from corpora is not a new idea. The techniques presented in this chapter were inspired in large part by the success of lexical acquisition techniques in learning verb subcategorization frames. The countability research of Francis Bond and others involved with NTT's ALT-J/E system also shaped the direction of this work. This section briefly details the research that inspired the shape of ACT.

4.1.1 Verb subcategorization

In recent years, significant advances have been made in the use of corpora as tools in language processing. Much research has focused on the automatic acquisition of verb subcategorization frames; this research is in part a response to research by Briscoe and Carroll (1993) showing that up to half of all parse failures are a result of incorrect subcategorization information in the parser's reference lexicon. Briscoe and Carroll (1997) also note that the lexicalist nature of most current syntactic theories make subcategorization information extremely useful in lexicon-oriented parsers and generators. Such parsers and generators can use information that an (accurately) annotated lexicon

provides to constrain the number of parses or output strings it produces.

The first and most obvious method for acquiring lexical features is manual tagging. Manual feature tagging generally ensures that trained linguists and/or lexicographers properly annotate a resource with the desired feature(s). COMLEX is one resource that was annotated largely through manual means.

However, manual tagging is extremely expensive in terms of time and annotation effort. Additionlly, annotators are unlikely to have the time and resources to continually update an annotated lexicon to keep up with new terminology and usage. Manning (1993) also cites the difficulty of tracking specialized terminology; Manning also notes that annotated lexicons may be desired for many languages — a significant challenge with manual tagging. Automatic acquisition of lexical feature from corpora promises a solution to these problems (albeit a partial solution in many cases).

Brent (1993) presents one of the first approaches to automatic acquisition of verb subcategorization frames from unannotated corpora. Brent notes that certain cues in text signal the occurance of specific features with high reliablity. He proposes that these cues be harnessed to acquire features automatically from corpora.

Brent considers two specific cues, one to identify verbs and one to identify argument phrases. The first cue notes that verbs tend to appear in a corpus in both inflected and uninflected forms — specifically in base form and with an -ing suffix. He argues that since few other words share this behavior, he can identify verbs in the corpus using nothing but this cue. Simililarly, Brent uses the occurance of the phrase "that the" to automatically identify the start of argument phrases. Using these cues as a base, Brent constructes a system capable of learning a number of verb subcategorization frames.

The remaining details of Brent's system will not be considered here. Numerous systems built since 1993 have surpassed Brent's initial attempts at automatic acquisition of verb subcategorization frames from corpora. Manning (1993) presents one such system that uses a stochastic part-of-speech tagger in combination with a finite-state parser to acquire subcategorization frames. Other research, including Briscoe and Carroll (1997), has helped to expand the coverage and accuracy of automatic verb subcategorization systems.

But most important to this work is Brent's initial observation about cues. Manning (1993) claims that Brent's use of cues is overly simplistic, ignoring a large number of verbs and possible frames. However, in the realm of noun countability, numerous high-reliability cues exist (see Chapter 2). Like Brent's verb cues, these countability cues do not cover every case. But these cues, based on observations in Chapter 2, can reliably predict a noun's countability feature in a significant number of situations.

4.1.2 Why study countability?

The work of Brent (1993) was important in its use of cues to identify lexical attributes for extraction. However, the focus of this work is noun countability, not verb subcategorization frames. It is therefore essential to establish a motivation for extracting countability features from corpora.

There are numerous reasons for wanting a machine-readable lexicon with accurate countability information. Accurate countability information could help improve word

sense disambiguation in certain cases. Such information can also assist lexicographers in dictionary construction. Learners of English as a second language may find a lexicon where nouns are marked with countability attributes extremely valuable.

But, perhaps the most important use for countability information is in machine translation. One need only skim the literature surronding countability research to observe a wide variety in how noun countability affects different languages.

Schmitt and Munn (1998), Schmitt and Munn (1999), and Schmitt and Munn (2000) consider countability in Brazilian Portuguese (specifically the nature of bare singular countable nouns); the authors contrast the usage of bare singular countable nouns in Brazilian Portuguese with the usage of such nouns in other Romance languages and in English. Singh (1992) considers how different treatment of countability in Hindi and English can make a significant difference in another area — perfective verb usage. Kay (1999) even explores the countability of color nouns in ancient Greek and Latin, questioning how close the usage of these terms in the ancient languages compares to their usage in English.

Knowledge of noun countability values is an important aspect in the creation of robust machine translation systems. This is best illustrated in a series of papers by Francis Bond countability in NTT's Japanese-to-English machine translation system ALT-J/E:

Bond et al (1994) describe the lack of explicit countability and number marking in most Japanese nouns; the paper explores the difficulty this creates in generating appropriate English translations of Japanese. Bond et al (1996) examine classifiers in Japanese and their relationship with countability in English. Bond and Ikehara (1996) look further into the challenge of disambiguating Japanese nouns with respect to countability when translating into English. Bond (2001) presents a thorough examination of number, countability, and determiners in English and Japanese; Bond also presents an algorithm for determining noun phrase countability and number in Japanese. Finally, Bond and Vatikiotis-Bateson (2002) show that countability in English can be predicted (to a certain extent) using a semantic ontology.

It should be noted that Bond (2001) also gives an overview of the lexicons for English and Japanese used by NTT in ALT-J/E. One feature marked in noun entries of the English lexicon is countability. The NTT countability lexicon described in Chapter 3 is derived from the English lexicon used in ALT-J/E.

A primary motive for automatic acquisition of noun countability features is therefore to assist in the creation of countability-marked lexicons for use in accurate machine translation. Countability information for the target language is crucial when translating from a language with little or no explicit marking of countability or number (such as Japanese and Chinese) into a language that requires some marking of countability and number (such as English and German). Additional reasons for creating countability-marked lexicons are to potentially improve some word sense disambiguation, to provide use as a lexicographic resource in dictionary construction, and to assist native speakers of, for example, Chinese and Japanese in learning English.

4.2 Automatic tagging

The above section explores the reason why countability is important and the rationale for creating an automatic countability tagger. The remainder of this chapter describes how such a tagger is implemented here. The tagger is implemented in Perl. See the appendices for code. The tagger is then run over the British National Corpus, producing a list of nouns extracted from the BNC, each marked with a countability feature value.

The British National Corpus contains over 100 million words of spoken and written text. Sction 3.1 explores the format of the corpus, as well as the pre-processing steps applied to it to prepare it for countability tagging. In short, a noun phrase chunker is run over each sentence in the corpus.

The chunker returns the head noun and determiner for each noun phrase that it detects in the corpus. See [10] and [11] for an example of chunker input and output. When run over the BNC, the noun phrase chunker returns over 16.27 million head nouns with corresponding determiners. This collection of head nouns and determiners serves as input to the main part of the automatic countability tagger.

The following base tagging algorithm uses the determiner of a noun phrase to predict the countability property of the head noun of the NP. The algorithm assumes that all nouns to be tagged are common singular nouns.

[22]

- If the determiner is *a*, *an*, *another*, *each*, *either*, *every*, *neither*, or *one*, tag countability of head noun as COUNT.
- Else if the determiner is *enough*, *insufficient*, *less*, *little*, *more*, *most*, *much*, *overmuch*, *such*, *sufficient*, or null (ϕ), tag countability of head noun as NON-COUNT.
- Else tag countability of head noun as UNSURE.

Processing the 16.27 million noun tokens from the BNC with the above algorithm results in the following distribution of tagged results:

	Noun tokens	Percent of total
COUNT	1.90 million	11.70%
NON-COUNT	4.80 million	29.52%
UNSURE	9.56 million	58.78%

This tagging algorithm is based on the factors influencing countability, specifically those mentioned in 2.2.1. However, the more complex factors given in section 2.3 are excluded from the above algorithm. This decision was made in order to simplify the tagging process. It is recognized that in making this choice, a certain level of error is introduced to the tagging results. For example, the tagger will always tag bare singular nouns as non-countable; cases such as [7.i] where a bare singluar noun is actually countable are always ignored. For a discussion of extensions to the base algorithm, see chapter 6.

Chapter 5

Results

The cues used to construct the tagging algorithm in [22] are believed to be highly reliable. These cues were compiled by closely examining countability in the grammars of (Huddleston and Pullum, 2002), (Quirk et al., 1985), and (Huddleston, 1988).

But, in order to be verified, the results of the ACT should be compared against other resources to gain some estimate of the level of error in automatic tagging. The ACT results are evaluated against COMLEX and the NTT lexicon. These resources are described in chapter 3. A baseline resource evaluation of COMLEX against the NTT lexicon is also given, in section 3.4.

5.1 COMLEX

For evaluation against COMLEX, each of the 16.27 million head noun tokens extracted from the BNC was examined to determine if that noun token was listed in COMLEX; 7.78 million of the noun tokens were listed in COMLEX. [1] This section first establishes a baseline for evalutating the BNC extracted head nouns against COMLEX. Once a baseline is clear, this section describes the results of evaluating the tagged BNC head nouns against COMLEX.

5.1.1 Baseline

Before attempting a direct evaluation of the countability-tagged BNC head nouns against COMLEX, it is important to establish a baseline against which to set the evaluation results. The following considers only those 7.78 million nouns extracted from the BNC which are found in COMLEX.[2]

The following results were obtained by looking up each of the 7.78 million head noun tokens in COMLEX and extracting the countability feature from COMLEX.

[1] For reference, this percentage of coverage (47.82%) is slightly higher than the level of coverage for COMLEX noun entries evaluated against the NTT lexicon (45.17%).

[2] An additional 8.48 million nouns were extracted from the BNC, but not included for consideration in the figures below because COMLEX did not include entries for these additional nouns.

	COUNT	NCOLL	AMBIG	COUNT_NCOLL
COMLEX-assigned countability	50.90%	3.55%	45.49%	0.05%

Based on the above data, a baseline of 50.90% may be established for all experiments ranking my results against COMLEX. An automatic countability tagger could correctly (according to COMLEX) tag 50.90% of the 7.78 million BNC noun tokens by tagging all noun tokens as COUNTABLE.

As a second point of reference, the distribution of countability for the 21,215 nouns listed in COMLEX is given below:

	COUNT	NCOLL	AMBIG	COUNT_NCOLL
Nouns in COMLEX	62.00%	2.93%	34.94%	0.13%

5.1.2 Results

The following details the level of agreement between this project's automatic countability tagger and the noun countability features in COMLEX.

	COUNT	NCOLL	AMBIG	COUNT_NCOLL
COUNT	68.45%	1.63%	29.81%	0.11%
NON	32.20%	5.96%	61.81%	0.03%
UNSURE	57.64%	2.54%	39.77%	0.05%

The above figures represent a significant improvement over the baseline. Of those nouns that the automatic countability tagger marked as countable (COUNT), 68.45% were marked as only countable (COUNT) in COMLEX. Only 1.63% of the tagger's results in this category were marked as definitely incorrect.

Results for nouns that the tagger considered non-countable (NON) are less clear. Only 5.96% of these nouns are directly confirmed as correct by COMLEX. However, this is still higher than the baseline, which does not correctly identify any non-countable noun instances.[3] Additionally, 61.81% of these nouns are considered by COMLEX to be ambiguous (AMBIG) with respect to countability. Thus, given the rather limited nature of COMLEX when it comes to non-countable nouns (only 2.93% of all COMLEX entries are marked as only non-countable), the tagger does rather well on this point.

[3] See 5.1.1. In the COMLEX baseline, all nouns are tagged as COUNTABLE.

5.2 NTT

Evaluation of the ACT results against the NTT lexicon proceeded in the same manner as the evaluation of COMLEX against the NTT lexicon (see section 3.4). Each of the 16.27 million head noun tokens extracted from the BNC was examined to determine if that noun token was listed in the NTT lexicon. Of the 16.27 million noun tokens extracted from the BNC, approximately 10.02 million were listed in the NTT lexicon.

Throughout this section, reference is made to evaluation methods 1, 2, and 3. A brief review is provided for each method in the appopriate subsection below. However, these evaluation methods are fully described in section 3.4. Please refer to that section for examples and more complete descriptions of each method.

5.2.1 Baseline

Like the baseline for COMLEX in section 5.1.1,

The baseline figures below apply to the 10.02 million noun tokens extracted from the BNC that also occur in the NTT lexicon.

for these matching

The baseline for the testing extracted BNC head nouns against NTT lexicon is listed below. The first column indicates the method used; the second column notes how many matches occurred using that method. The remaining columns show how the countability features are distributed. These figures are derived by looking up each of the 10.02 million noun tokens in the NTT lexicon and taking the countability feature suggested by the NTT lexicon.

	Matches	CO	CB	UB	UC	UP	NN
Method 1	13.72 million	58.19%	8.99%	9.05%	17.75%	4.11%	1.92%
Method 2	10.02 million	62.92%	9.28%	8.73%	14.06%	4.35%	0.67%

	Matches	CO/CB	UC/UB	UP	NN
Method 3	10.02 million	68.41%	25.78%	5.25%	0.56%

Using the above data, a baseline level may be established with respect to each evaluation method. If a tagger were to tag all input as CO, it would by correct 58.19% of the time under method 1 and 62.92% of the time under method 2. If the tagger were to tag all input as CO/CB, it would be correct 68.41%. The tagger would be incorrect in all cases where the noun's true countability is not CO (or CO/CB under method 3).

For reference, the distribution of countability within the NTT data itself is provided below. Here, distribution 1 is similar to method 1 — multiple entries per word are possible (see [18]. Distribution 2 is similar to method 2 — only only one entry is allowed per word, namely the entry corresponding to the most frequent sense of the word (see [19]).

	CO	CB	UB	UC	UP	NN
Distribution 1	67.51%	2.51%	3.44%	22.52%	3.06%	0.96%
Distribution 2	68.92%	2.45%	3.15%	21.77%	3.02%	0.69%

5.2.2 Method 1

Each of the noun of the 10.02 million noun tokens from the BNC that also exist in the NTT lexicon are looked up in the NTT lexicon. Each lexical item may have multiple entries in the NTT lexicon (see [17]). In method 1, a word X from the BNC scores one match for each distict countability sense that the NTT lexicon has for word X. See 3.4.2 for additional details.

The Distribution line reveals the breakdown of countability values within the NTT lexicon itself. The following lines give the countability matches (within the NTT lexicon) for nouns in the BNC that the ACT considered COUNTABLE, NON-COUNTABLE, and UNSURE.

	CO	CB	UB	UC	UP	NN
Distribution	67.51%	2.51%	3.44%	22.52%	3.06%	0.96%
COUNT	71.68%	9.31%	6.91%	9.67%	0.44%	1.99%
NON	48.44%	10.20%	11.63%	26.34%	1.65%	1.74%
UNSURE	60.86%	8.02%	7.87%	14.16%	7.07%	2.02%

5.2.3 Method 2

Method 2 requires that each noun token from the BNC match, at most, one entry in the NTT lexicon. Multiple entries for a single word, as in [17], are collapsed into a single entry, as in [19]. When a word has multiple entries in the NTT lexicon, the countability for the collapsed entry is the countability value that was listed in the most entries for the word prior to collapse. See 3.4.3 for more details.

Again, the Distribution line reveals the breakdown of countability values within the NTT lexicon, after multiple entries have been collapsed into single entries. The following lines give the countability matches with nouns from the BNC (as described above).

	CO	CB	UB	UC	UP	NN
Distribution	68.92%	2.45%	3.15%	21.77%	3.02%	0.69%
COUNT	81.01%	9.13%	5.44%	3.91%	0.20%	0.31%
NON	48.04%	11.48%	13.14%	26.66%	0.24%	0.43%
UNSURE	66.40%	7.91%	7.01%	9.38%	8.37%	0.94%

5.2.4 Method 3

Method 3 is very similar to method 2. However, when multiple entries are collapsed into a single entry, CO and CB values are treated as a single value (CO/CB) and UC and UB are treated as a single value. See 3.4.4 for a detailed explanation with an example.

	COUNT	NON	UP	NN
Method 3	71.11%	25.23%	3.03%	0.63%
COUNT	87.97%	11.45%	0.25%	0.32%
NON	53.84%	43.84%	2.03%	0.29%
UNSURE	71.21%	19.00%	8.98%	0.81%

5.3 Evaluation summary

The values reported in the previous two sections confirm that the automatic countability tagger performs a useful function. The ACT consistently provides results higher than the baseline of choosing the most common countability feature. Agreement between the ACT results and each of the two lexicons, COMLEX and the NTT lexicon, is good, but does not reach the level of agreement that the two lexicons show when evaluated against each other.

Chapter 6

Extensions

Even given the relatively simple cues used in the ACT tagging algorithm, the tagger is able to perform well. It is expected that as additional cues are added to expand the tagging algorithm, results will further improve. An decrease in the number of nouns tagged as UNSURE is especially expected with extensions to the system.

This section provides a starting point for extensions to the base system which was implemented in this project. It is regrettable, but time did not permit the completion of these extensions in time for inclusion in this thesis.

6.1 CIDE

The first extension is a simple expansion of the lexicon base. The Cambridge International Dictionary of English (CIDE) is another machine-readable lexicon which has countability marked in nouns. With a small amount of pre-processing, CIDE could be used as an additional testing resource. This would supplement the use of COMLEX and the NTT lexicon.

6.2 Plurals

The ACT in this work did not consider plural nouns. However, as section 2.2.2 points out, many plural nouns can be disambiguated with respect to countability with relative easy. The presence or absence of a regularly occuring singular form of a plural noun is a strong cue about its countability. Plurals with a singular form are usually countable, while those with no singular form are usually not countable. Alternatively, Bond (2001) argues for a distinct countability category for plural only nouns (UP).

6.3 Countability Preference Estimation

The output of the automatic countability tagger is a massive (16.27 million) list of noun tokens from the BNC. Each token is marked with a countability tag, either COUNTABLE, NON-COUNTABLE, or UNSURE.

One goal of this thesis was that the ACT should be able to produce not just a list with three possible countability values per token, but at list of nouns, each marked with the probability that it would have certain countablity values.

This extension was started, but due to time constraints was not completed.

It was intended that the output of the baseline ACT could act as input to a second program. This program would calculate the probability that a given noun may appear in a COUNTABLE context. The program would use the noun's prior countability behavior to guide its decision.

For example, a noun X that was tagged as COUNTABLE by the ACT 20 times, tagged NON-COUNTABLE 1 time, and tagged UNSURE 5 times would be given a high probability of countability. Likewise, a noun Y with the following distribution wolud be given a low probability of countability: 3 COUNTABLE, 15 NON-COUNTABLE, 4 UNSURE.

Data sparsity will likely be a major issue in the implementation of this extension. The above behavior depends on having numerous examples of each noun. For less common nouns, this system is likely to be unreliable without some form of backoff.

The largest unresolved issue in this extension is how exactly to distribute the probability mass of UNSURE instances. Even nouns which do not suffer from data sparsity may have distributions where the number of UNSURE instances outnumber the other two. One possibility is to incorporate the semantic Wordnet hierarchy discussed below as a form of backoff.

6.4 Back-off

In certain cases it may be helpful to back off to secondary methods rather than simply to tag nouns with the UNSURE countability. Multiple options are available for use in backoff.

6.4.1 Bond's tests

Bond (2001) poses three question to help decide a noun's countability preference:

- Does the noun have a plural form?
- Can the noun have a or an as a determiner?
- Can the noun have much as a determiner?

Based solely on the answers to these three questions, the noun can be classified as either fully countable (CO), partially countable (either CB or UB), uncountable (UC), semi-countable (like CO but can take a as a determiner), and plural only (UP).

6.4.2 Hypernyms and WordNet

It has been observed that there is a widespread similarity between semantically related nouns to have the same countability. For example, most nouns which refer to animals are countable, while most nouns which refer to chemical substances are non-countable.

It would be useful to determine how well semantic classes can predict countability. Once it was established that semantic classes can be used in countability tagging, a reasonable extension would be to incorporate such a semantic class network into the ACT, most likely for use in backoff.

Bond and Vatikiotis-Bateson (2002) present just such research. In their experiments, an ontology of 2,710 semantic nodes was linked to NTT's ALT-J/E lexicon. The authors take the most commonly occuring noun countability preference in a semantic class as the countability value for that semantic class. Countability values in their ontology are able to successfully predict the countability of the words in the lexicon up to 77.9% of the time.

6.4.3 Default to countable

As last step in backoff, a tagger may assume that every noun not otherwise tagged is countable. Bond (2001) advocates this position, stating that the default countability feature for nouns should be fully countable (CO). The high percentage of countable nouns in the countability distributions of COMLEX (62.00%) and the NTT lexicon (71.11%) also support this position.

It should be noted that the extensions listed in this chapter vary in their degrees of reliability. The cues used in the basic ACT are considered to be quite reliable. It is expected that backoff to countable, would not be as reliable. But, this broad set of extensions allows the user of an automatic countability tagger to deteremine their own balance between coverage and accuracy. If higher quality is desired, the basic methods for tagging can be used. But, if wider coverage is required, the backoff methods provide two additional levels of countability estimation.

6.5 Expanded tagging algorithm

The following tagging algorithm should provide good results, based on the foundation of this system's base implementation and the extensions described above. This algorithm is, however, untested.

[23]

- If the noun is a proper noun, ignore.
- Else if the noun is plural...
  - If a singular form of the same noun exists, tag countability as COUNTABLE.
  - Else tag countability as PLURAL-ONLY.
- Else if the determiner is *a*, *an*, *another*, *each*, *either*, *every*, *neither*, or *one*, tag countability of head noun as COUNT.
- Else if the determiner is *enough*, *insufficient*, *less*, *little*, *more*, *most*, *much*, *overmuch*, *such*, *sufficient* tag countability of head noun as NON-COUNT.
- Else if the determiner is null (ϕ)...
  - If the noun is a noun that can be used countably as a predicative complement, and the noun is being used as a predicative complement, tag countability as COUNTABLE.
  - Else tag countability as NON-COUNTABLE.
- Else if the determiner is all...
  - If the noun is a unit of time, tag countability as COUNTABLE
  - Else tag countability as NON-COUNTABLE
- Else if no standard plural form exists for the noun, tag countability as NON-COUNTABLE.
- Else back off to countability suggested by the Wordnet backoff, or backoff to Bond's three-point test.
- Else assume countability of head noun is COUNTABLE (or tag as UNSURE).

Chapter 7

Conclusion

Lexical acquisition techniques have been shown to be successful in areas such as verb subcategorization frame learning. This work attempts to take automatic lexical aqcuisition to a new area: noun countability.

Despite using relatively simple determiner-based cues, the automatic countability tagger presented in this work is able to accurately tag nouns with countability features. The automatic countability tagger can correctly tag nouns with countability in up to 87% of noun phrases.

The use of the British National Corpus provided a very lage, part-of-speech tagged corpus on which to test the automatic countability tagger. The COMLEX resource and the NTT countability lexicon were extremely useful resources, providing countability-marked lexicons against which to test the results of the automatic countability tagger.

It is unfortunate that time did not permit the completion of the countability preference estimation extension. I believe that this extension has the potential to make the ACT presented here even more useful by providing more fine-grained countability results (such as those used by Bond).

The other extensions offer additional possibilities to make this work's automatic countability tagger an extremely useful tool in lexical acquisition from corpora.

Appendix A

Bipartite nouns

This work takes exception to the inclusion by Huddleston (1988) of bipartite nouns[1] with other non-countable plurals. I believe that such nouns (Huddleston gives *scissors* and *pyjamas* as examples) are better classified as a special subcategory of countable nouns. Huddleston and Pullum (2002) acknowledge that there is disagreement between dialects (and even between speakers) in the usage of certain bipartites in reference to groups of the respetive item. The authors give an example where such usage is generally accepted: *All the scissors need sharpening.* Other examples of bipartites acting in a fully plural context include:

[24] *How many jeans did you pack?*
Thirty goggles hung on pegs at the entrance to the workshop.
I can't believe you actually fold all of your boxers.

The fact that certain bipartites can be used explicitly as plurals begs the question: What countability and number features did these bipartites have before they were coerced into an undeniably plural usage? The fact that numerous (though by no means all) bipartites can pluralize beyond their base form as in [24] above suggests that bipartites are are conceptually (even if not syntactically) singular, and that they are in fact countable. Indeed, Huddleston and Pullum (2002) themselves observe that bipartites generally consist of two physical parts, but still function as a single entity.

A contrasting approach is that of Bond (2001). Bond places bipartites, as well as all plural-only nouns, in a separate countability class. This approach seems much more reasonable than the one proposed above by Huddleston. This work adopts the treatment that bipartites should be either treated as a proper (albeit slightly unusual) subset of countable nouns or as plural only, using Bond's terminology.

[1] See (Huddleston and Pullum, 2002) pp. 340-345 for an expanded discussion of bipartites and other plural-only nouns.

Appendix B

Coverage Rates

Appendix C

Code

The amount of code presented in this appendix is not completely indicitive of the level of computational effort involved. The primary corpus used was the BNC, which is 100 million lines of text. Much of the effort in implementing this project involved manipulations of the BNC and the other lexical resources. The GNU Text Utilities, especially join, uniq, sort, grep, and wc were invaluable in pre-processing the various text files and in evaluating the resources against each other.

Several Perl programs were written for use in manually tagging data and for incorporating the manual tags into other data. Since no manual tagging results were used in the actual thesis, I have omitted these programs.

I have also omitted several trivial Perl programs. These programs were primarily used in text manipulation (such as extracting a single column of text) when the GNU Text Utilities could not easily handle the task.

C.1 Countable Head Nouns

Listing for countable_head_nouns.pl:

```
#!/usr/bin/perl -w

# OPEN THE BNC #
open(<BNC>, "/usr/groups/corpora/bnc/text");

 $this_is_a_reloop    = 0; #define as    initially    false

$already_in_an_NP = ""; #define as    initially    false

#$x = 0;
#$num_of_NPs = 0;

while (<BNC>) {
```

```
unless ( $this_is_a_reloop ) {

    ( $word, $tag1 , $tag2 ) = split ;

    if ( defined ( $tag2 ) ) {
        unless ( $tag2 eq "" ) {
            $there_is_ambiguity ++;
        }
    }

    unless ( defined ( $word) && defined ($tag1) ) {
        next;
    }
}

 $this_is_a_reloop = 0; # reset to false



unless ( $already_in_an_NP ) {

    $det = ""; ### Reset to no determiner if this is a new NP ###

    if ( $tag1 eq "NP0") {
        # print "propre noun\n";
        $already_in_an_NP = "NP0";
        push( @current_NP, $word );
    }
    elsif ( $tag1 eq "AT0" || $tag1 eq "DPS" || $tag1 eq "DT0" || $tag1 eq "
        DTQ" ){
        #3 print "determiner\n";
        $det = $word;
        $already_in_an_NP = "DET";
        push( @current_NP, $word );
    }
    elsif ( $tag1 eq "AJ0" || $tag1 eq "AJC" || $tag1 eq "AJS" || $tag1 eq "
        ORD" ) {
        # print " adjective \n";
        $already_in_an_NP = "ADJ";
        push( @current_NP, $word );
    }
    elsif ( $tag1 eq "AV0") {
        # print "adverb\n";
        $already_in_an_NP = "ADV";
        push( @current_NP, $word );
    }
    elsif ( $tag1 eq "NN0" || $tag1 eq "NN1" ) {
        # print "common noun\n";
        $already_in_an_NP = "common_noun";
        push( @current_NP, $word );
    }
    elsif ( $tag1 eq "NN2" ) {
        # print "common noun plural\n";
        $already_in_an_NP = "common_noun_plural";
```

```
            push( @current_NP, $word );
        }
        else {
            # print "not start of an NP\n";
        }
    }
    else {

        if ( ( $tag1 eq "NP0") && ( $already_in_an_NP eq "NP0" ) ) {
            # Proper nouns can only occur within an NP after another proper nounw
            $already_in_an_NP = "NP0";
            push( @current_NP, $word );
        }
        elsif ( $tag1 eq "NN0" || $tag1 eq "NN1" ) {
            # Common nouns may occur at the end of the NP, followed only by other
                common nouns
            push( @current_NP, $word );
            $already_in_an_NP = "common_noun";
        }
        elsif ( $tag1 eq "NN2" ) {
            # Common nouns may occur at the end of the NP, followed only by other
                common nouns
            push( @current_NP, $word );
            $already_in_an_NP = "common_noun_plural";
        }
        elsif ( ( $tag1 eq "AJ0" || $tag1 eq "AJC" || $tag1 eq "AJS" || $tag1 eq "
            ORD" ) && ( $already_in_an_NP eq "ADJ" || $already_in_an_NP eq "DET"
            || $already_in_an_NP eq "ADV" ) ) {
            # Adjectives may occur after determiners , adjectives , and adverbs
            push( @current_NP, $word );
            $already_in_an_NP = "ADJ";
        }
        elsif ( ( $tag1 eq "AV0" ) && ( $already_in_an_NP eq "ADV" ||
            $already_in_an_NP eq "DET" ) ) {
            # Adverbs may occur after determiners and adverbs
            push( @current_NP, $word );
            $already_in_an_NP = "ADV";
        }
        else { # At the end of the current NP

            $head_noun = $current_NP[-1];
            #if ( ! $there_is_ambiguity && ( $head_noun eq "NP0" || $head_noun eq "
                NN0" || $head_noun eq "NN1" || $head_noun eq "NN2" )) {
            if ( $already_in_an_NP eq "NP0" || $already_in_an_NP eq "common_noun"
                || $already_in_an_NP eq "common_noun_plural" ) {
                #$num_of_NPs++;
                #print ( " det :", $det , " -- hn : " , $head_noun, " --\n");
                #print ( $x , " ", $num_of_NPs , ":\"", $det ,"\"", " \"" , $head_noun
                    , "\"\n");



                # Tag stuff !!!
```

```
            $countability = "---";

            if ( $already_in_an_NP eq "common_noun" ){

                if ( $det eq "a" || $det eq "A" || $det eq "an" || $det eq "AN"
                    || $det eq "one" || $det eq "ONE" || $det eq "One" || $det
                    eq "another" || $det eq "ANOTHER" || $det eq "Another" ||
                    $det eq "each" || $det eq "EACH" || $det eq "Each" || $det
                    eq "every" || $det eq "EVERY" || $det eq "Every" || $det eq
                    "either" || $det eq "EITHER" || $det eq "Either" || $det eq
                    "neither" || $det eq "NEITHER" || $det eq "Neither") {
                    $countability = "count"; }

                elsif ( $det eq "enough" || $det eq "ENOUGH" || $det eq "Enough"
                    || $det eq "much" || $det eq "MUCH" || $det eq "Much" || $det
                    eq "most" || $det eq "MOST" || $det eq "Most" || $det eq "
                    more" || $det eq "MORE" || $det eq "More" || $det eq " little
                    " || $det eq "LITTLE" || $det eq " Little " || $det eq "less"
                    || $det eq "LESS" || $det eq "Less" || $det eq " sufficient "
                    || $det eq "SUFFICIENT" || $det eq " Sufficient " || $det eq "
                    insufficient " || $det eq "INSUFFICIENT" || $det eq "
                    Insufficient " || $det eq "overmuch" || $det eq "OVERMUCH
                    " || $det eq "Overmuch" || $det eq "such" || $det eq "SUCH" ||
                    $det eq "Such" || $det eq "") { $countability = "non"; }

        }

            if ( $det eq "") {$det = "---";}

            # print ( $det , " ", $head_noun, "\n");

            # print ( $head_noun , " ", $countability , "\ n");
            if ( $countability eq "count") {print $head_noun, "\n";}

            #$curr_NP = " @current_NP";

            # printf "%10s %7s %-15s %s \n", $countability , $det ,
                $head_noun, $curr_NP;


        }

         $this_is_a_reloop = 1; # Set to true
        $already_in_an_NP = ""; # Set to false

         $there_is_ambiguity = 0; # set to false
        @current_NP = ();


        # print "--------------------\n";
        redo;
    }

}
```

```
}

close BNC;
```

C.2 Automatic Chunker and Tagger

Listing for tag_auto.pl:

```
#!/usr/bin/perl -w


#To use: fali:scripts$ ./corpus_tools/clean_training_corpus_for_tagger.pl | tag_auto
    .pl


 $this_is_a_reloop = 0; #define as initially false

$already_in_an_NP = ""; #define as initially false

#$x = 0;
#$num_of_NPs = 0;


open(COUNT, ">/local/scratch/los20/count_NPs.txt") || die "Unable to open /tmp/los20/
    count_NPs.txt for output.";
open(NON, ">/local/scratch/los20/non_count_NPs.txt") || die "Unable to open /tmp/
    los20/non_count_NPs.txt for output.";
open(UNSURE, ">/local/scratch/los20/unsure_count_NPs.txt") || die "Unable to open /
    tmp/los20/unsure_count_NPs.txt for output.";


while (<>) {


    unless ( $this_is_a_reloop ) {

        ( $word, $tag1, $tag2 ) = split;

        if ( defined ( $tag2 ) ) {
            unless ( $tag2 eq "" ) {
                $there_is_ambiguity ++;
            }
        }

        unless ( defined ($word) && defined ($tag1) ) {
            next;
        }
    }

     $this_is_a_reloop = 0; # reset to false



    unless ( $already_in_an_NP ) {

        $det = ""; ### Reset to no determiner if this is a new NP ###
```

```
    if ( $tag1 eq "AT0" || $tag1 eq "DPS" || $tag1 eq "DT0" || $tag1 eq "DTQ
        " ) {
        #3 print " determiner \n";
        $det = $word;
        $already_in_an_NP = "DET";
        push( @current_NP, $word );
    }
    elsif ( $tag1 eq "AJ0" || $tag1 eq "AJC" || $tag1 eq "AJS" || $tag1 eq "
        ORD" ) {
        # print " adjective \n";
        $already_in_an_NP = "ADJ";
        push( @current_NP, $word );
    }
    elsif ( $tag1 eq "AV0") {
        # print "adverb\n";
        $already_in_an_NP = "ADV";
        push( @current_NP, $word );
    }
    elsif ( $tag1 eq "NN0" || $tag1 eq "NN1" ) {
        # print "common noun\n";
        $already_in_an_NP = "common_noun";
        push( @current_NP, $word );
    }
    elsif ( $tag1 eq "NN2" ) {
        # print "common noun plural\n";
        $already_in_an_NP = "common_noun_plural";
        push( @current_NP, $word );
    }
    else {
       # print "not start of an NP\n";
    }
}
else {


    if ( $tag1 eq "NN0" || $tag1 eq "NN1" ) {
        # Common nouns may occur at the end of the NP, followed only by other
            common nouns
        push( @current_NP, $word );
        $already_in_an_NP = "common_noun";
    }
    elsif ( $tag1 eq "NN2" ) {
        # Common nouns may occur at the end of the NP, followed only by other
            common nouns
        push( @current_NP, $word );
        $already_in_an_NP = "common_noun_plural";
    }
    elsif ( ( $tag1 eq "AJ0" || $tag1 eq "AJC" || $tag1 eq "AJS" || $tag1 eq "
        ORD" ) && ( $already_in_an_NP eq "ADJ" || $already_in_an_NP eq "DET"
        || $already_in_an_NP eq "ADV" ) ) {
        # Adjectives may occur after determiners , adjectives , and adverbs
        push( @current_NP, $word );
        $already_in_an_NP = "ADJ";
    }
```

```
elsif    ( ( $tag1 eq "AV0" ) && ( $already_in_an_NP eq "ADV" ||
      $already_in_an_NP eq "DET" ) ) {
     # Adverbs may occur after determiners and adverbs
     push( @current_NP, $word );
     $already_in_an_NP = "ADV";
}
else { # At the end of the current NP

     $head_noun = $current_NP[-1];

     if ( $already_in_an_NP eq "common_noun" || $already_in_an_NP eq "
          common_noun_plural" ) {
          #$num_of_NPs++;
          # print ( " det :", $det , " -- hn : " , $head_noun, " --\n");
          # print ( $x , " ", $num_of_NPs , ":\"", $det ,"\"", " \"" , $head_noun
               , "\"\n");



          # Tag stuff !!!

          $countability = "---";

          if ( $already_in_an_NP eq "common_noun" ){

               if ( $det eq "a" || $det eq "A" || $det eq "an" || $det eq "AN"
                    || $det eq "one" || $det eq "ONE" || $det eq "One" || $det
                    eq "another" || $det eq "ANOTHER" || $det eq "Another" ||
                    $det eq "each" || $det eq "EACH" || $det eq "Each" || $det
                    eq "every" || $det eq "EVERY" || $det eq "Every" || $det eq
                    " either " || $det eq "EITHER" || $det eq " Either " || $det eq
                    " neither " || $det eq "NEITHER" || $det eq "Neither") {
                    $countability = "count"; }

               elsif ( $det eq "enough" || $det eq "ENOUGH" || $det eq "Enough"
                    || $det eq "much" || $det eq "MUCH" || $det eq "Much" || $det
                     eq "most" || $det eq "MOST" || $det eq "Most" || $det eq "
                    more" || $det eq "MORE" || $det eq "More" || $det eq " little
                    " || $det eq "LITTLE" || $det eq " Little " || $det eq " less "
                    || $det eq "LESS" || $det eq "Less" || $det eq " sufficient "
                    || $det eq "SUFFICIENT" || $det eq " Sufficient " || $det eq "
                     insufficient " || $det eq "INSUFFICIENT" || $det eq "
                    Insufficient " || $det eq "overmuch" || $det eq "OVERMUCH
                    " || $det eq "Overmuch" || $det eq "such" || $det eq "SUCH" ||
                    $det eq "Such" || $det eq "") { $countability = "non"; }

     }

          if ( $det eq "") {$det = "---";}

          # print ( $det , " " , $head_noun, "\n");

          # print ( $head_noun , " " , $countability , "\ n");
```

```
                if ( $countability eq "count") {
                    print COUNT $det, " " , $head_noun, "\n";
                }
                elsif ( $countability eq "non") {
                    print NON $det, " " , $head_noun, "\n";
                }
                elsif ( $countability eq "---") {
                    print UNSURE $det, " " , $head_noun, "\n";
                }
                else {
                    #ERROR: Should never reach this point.
                    die "Error in print block.\nShould never reach this point.\
                        nComplain to the programmer.\n";
                }

                #$curr_NP = " @current_NP";

                # printf "%10s %7s %-15s %s \n", $countability , $det ,
                    $head_noun, $curr_NP;


            }

             $this_is_a_reloop = 1; # Set to true
            $already_in_an_NP = ""; # Set to false

             $there_is_ambiguity = 0; # set to false
            @current_NP = ();


            # print "--------------------\n";
            redo;
        }

    }


}



close COUNT;
close NON;
close UNSURE;
```

C.3 Other Perl Scripts

Listing for comlex_lines.pl:

```
#!/usr/bin/perl -w


# All of the following ignore the 17 initial lines of comments in COMLEX.


# There are 63901 lines of data in COMLEX.
#
# Use to check:     cat /usr/groups/dict/comlex/comlex_synt/comlex_synt_1.1.1 | grep
      ^[^\;] |  wc -l


# There are 38326 entries in COMLEX.
#
# Use to check:     cat /usr/groups/dict/comlex/comlex_synt/comlex_synt_1.1.1 | grep
      ^\( |  wc -l


# There are 25575 lines that start with white space. These are carryovers from entry
      lines.
#
# Use to check:     cat /usr/groups/dict/comlex/comlex_synt/comlex_synt_1.1.1 | grep
      ^[^\(\;] |  wc -l
# Use to check:     cat /usr/groups/dict/comlex/comlex_synt/comlex_synt_1.1.1 | grep
      ^[\ ] |  wc -l

##### 38326 start lines + 25575 continue lines = 63901 total lines ######


# There are 21225 noun entries in COMLEX.
#
# Use to check:     cat /usr/groups/dict/comlex/comlex_synt/comlex_synt_1.1.1 | grep
      ^\( NOUN | wc -l


# There are 3137 adverb entries in COMLEX.
#
# Use to check:     cat /usr/groups/dict/comlex/comlex_synt/comlex_synt_1.1.1 | grep
      ^\( ADVERB | wc -l

# There are 5583 verb entries in COMLEX.
#
# Use to check:     cat /usr/groups/dict/comlex/comlex_synt/comlex_synt_1.1.1 | grep
      ^\( VERB | wc -l

# There are 22 determiner entries in COMLEX.
#
# Use to check:     cat /usr/groups/dict/comlex/comlex_synt/comlex_synt_1.1.1 | grep
      ^\( DET | wc -l
```

```
# There are 7885 adjective entries in COMLEX.
#
# Use to check: cat /usr/groups/dict/comlex/comlex_synt/comlex_synt_1.1.1 | grep
    ^\(ADJECTIVE | wc -l

# There are 23 WORD entries in COMLEX.
#
# Use to check: cat /usr/groups/dict/comlex/comlex_synt/comlex_synt_1.1.1 | grep
    ^\(WORD | wc -l

# There are 451 other entries in COMLEX.
#
# Use to check: cat /usr/groups/dict/comlex/comlex_synt/comlex_synt_1.1.1 | egrep
    -v '^\((WORD|ADVERB|NOUN|VERB|DET|ADJECTIVE)' | egrep -v '^;' | egrep
    -v '^ ' | wc -l

##### 21225 nouns + 3137 adverbs + 5583 verbs + 22 determiners + 7885 adjectives
    + 23 WORDs + 451 others = 38326 entries ##########


# There are 13187 countable nouns in COMLEX.
#
# Use to check: cat /usr/groups/dict/comlex/comlex_synt/comlex_synt_1.1.1 | egrep
    'COUNTABLE' | wc -l


# There are 649 NCOLLECTIVE nouns in COMLEX.
#
# Use to check: cat /usr/groups/dict/comlex/comlex_synt/comlex_synt_1.1.1 | egrep
    'NCOLLECTIVE' | wc -l

# There are 27 nouns that are BOTH countable AND ncollective.
#
# Use to check: ./comlex_lines.pl | egrep -v 'O--' | egrep 'NCOLLECTIVE' |
    egrep 'COUNTABLE' | wc -l

# There are 7416 other nouns, ambiguous towards number in COMLEX.
#
# Use to check: 21225 nouns - (13187 countable nouns + 649 ncollective nouns - 27
    ncollective / countables)


####### 13187 countable nouns + 649 ncollective nouns + 7416 other nouns - 27
    ncollective / countables = 21225 nouns ###############


#caga: scripts$ ./comlex_lines.pl | egrep '^N' | wc -l
#  21225
#caga: scripts$ ./comlex_lines.pl | egrep '^NC' | wc -l
#  13187
#caga: scripts$ ./comlex_lines.pl | egrep '^NN' | wc -l
#    622
#caga: scripts$ ./comlex_lines.pl | egrep '^NZ' | wc -l
#   7416
```

```
#caga: scripts$  ./ comlex_lines.pl | egrep '^O' |  wc -l
#  17101


# caga: scripts$  ./ comlex_lines.pl | egrep '__COUNTABLE' | wc -l
#    13160
# caga: scripts$  ./ comlex_lines.pl | egrep '__COUNT_and_NCOLL' | wc -l
#         27
# caga: scripts$  ./ comlex_lines.pl | egrep '__NCOLLECTIVE' | wc -l
#       622
# caga: scripts$  ./ comlex_lines.pl | egrep '__AMBIGUOUS' | wc -l
#      7416
# caga: scripts$  ./ comlex_lines.pl | egrep '(__COUNTABLE|__COUNT_and_NCOLL|
      __NCOLLECTIVE|__AMBIGUOUS)' | wc -l
#    21225


open(COMLEX, "/usr/groups/dict/comlex/comlex_synt/comlex_synt_1.1.1") || die "Can't
     open COMLEX!";

$noun_line = "";
$x = 0;

while ( <COMLEX> ){ $x++; if ($x >= 17) {last;}}

while(<COMLEX>) {

    chomp;

    if ( $_ =~ /^[^\(]/ )   { $noun_line .= $_; }
    else {
        if ( $noun_line =~ /^\( NOUN /) {

            @orth = split (/\"/, $noun_line);
            # print $noun_line, "\n";
            if ( $noun_line =~ /COUNTABLE/) {
                if ( $noun_line =~ /NCOLLECTIVE/) {
                    if ( $noun_line =~ /AGGREGATE/) {
                        print $orth[1], "     COUNT_NCOLL_and_AGGR\n";
                    }
                    else {
                        print $orth[1], "     COUNT_and_NCOLL\n";
                    }
                }
                elsif ( $noun_line =~ /AGGREGATE/) {
                    print $orth[1], "     COUNT_and_AGGR\n";
                }
                else {
                    print $orth[1], "     COUNTABLE\n";
                }
                # print "NC---",$orth[1],"---",$noun_line, "\n";
            }
            elsif ( $noun_line =~ /NCOLLECTIVE/) {
                if ( $noun_line =~ /AGGREGATE/) {
```

```
                    print $orth [1], "\t\tNCOLL_AGGREGATE\n";
                }
                else {
                    print $orth [1], "\t\tNCOLLECTIVE\n";
                }
                #print "NN---",$noun_line, "\n";
            }
            elsif ( $noun_line =~ / AGGREGATE/) {
                print $orth [1], "\t\tAGGREGATE\n";
            }
            else {
                print $orth [1], "\t\tAMBIGUOUS\n";
                #print "NZ---",$noun_line, "\n";
            }
        }
        else {
            # print "O---",$noun_line,"\n";
        }

        $noun_line = $_;
    }

}

close COMLEX;
```

Listing for score_against_COMLEX.pl

```
#!/usr/bin/perl -w

# Scores the results of the automatic tagger against the converted COMLEX data.



$number_matched_correctly = 0;

# $number_matched_incorrectly = 0;

# $number_not_matched_to_any_entry_in_COMLEX = 0;



# open(COMLEX, "~/thesis/perl/results/COMLEX_nouns_marked_with_countability.txt") ||
    die "Can't open COMLEX!";
#open(COMLEX, "/home/haverhill/los20/thesis/perl/results/
    COMLEX_nouns_marked_with_countability.txt") || die "Can't open COMLEX!";

#open(TAGGED_COUNTABLE, "/home/haverhill/los20/thesis/perl/results/
    sorted_alphabetically/countable_head_nouns_in_training_corpus1.txt") || die "Can'
    t open TAGGED_COUNTABLE!";

open(TAGGED_COUNTABLE, "/local/scratch/los20/COMLEX_noun_phrases/
    raw_NPs_with_dets/underscores_for_blanks_and_no_special_symbols/count_NPs.txt");


# $_ = <TAGGED_COUNTABLE>;
 $total_lines = 0;

while (<TAGGED_COUNTABLE>) {
    $total_lines ++;
    chomp;

    @tags = split ;
    ###   if ($tag[1] =~ /\'/)  { print $tag[1]; last ; }
    ###   $tags[1] =~ s/'/\'/g;

    #print $tags[1], "\n";

    $syscomm = "egrep -q -e " . "^$tags[1] " . " /home/haverhill/los20/thesis/perl/
        results/COMLEX_nouns_marked_with_countability.txt > /dev/null";
    #print $syscomm, "\n";
    # print "egrep -q -e " . "$tags[1]" . "\n";

    #$x = system "egrep ", "'^ $tags[1] '", " /home/haverhill/los20/thesis/perl/
        results/COMLEX_nouns_marked_with_countability.txt";
    $x = system($syscomm);
    #print "----$x----\n";
    if ($x == 0) {
        $number_matched_correctly++;
        # print "$number_matched_correctly of $total_lines words tagged so far are in
            COMLEX.\n";
```

```
        # print $tags [1], " ", $x , " ", $number_matched_correctly , "\n";
        print " ++ $total_lines ++  "
    }
    elsif ($x == 1 || $x == 256) {
        #Not a match
        # print $tags [1], " not a match.\n";
        print "  -- $total_lines --  "
    }
    elsif ($x == 2) {
        print "\n\n\nError matching: ", $tags [1], "\n";
        print "$number_matched_correctly of  $total_lines  words tagged so far are in
            COMLEX.\n\n\n";
    }
    else {
        print "\n\n\n----$x----\n";
        print "should never get here.\n";
        print "Error matching: ", $tags [1], "\n";
        print "$number_matched_correctly of  $total_lines  words tagged so far are in
            COMLEX.\n\n\n";
    }
}

#close COMLEX;
close TAGGED_COUNTABLE;

print "$number_matched_correctly of  $total_lines  words tagged are in COMLEX.\n";
#print $number_matched_correctly , " matched.\n";
```

Listing for test_NTT_majority_2.pl

```
#!/usr/bin/perl -w


$current_word = "";
# $current_count = 0;
$current_weight = 0;

$count_weight = 0;
$noncount_weight = 0;
$UP_weight = 0;
$nn_weight = 0;



$_ = <STDIN>;
chomp;
@tags = split ;
 $current_countability = $tags [2];
$current_word = $tags [1];
$current_weight = $tags [0];

if ( $current_countability eq "CO" || $current_countability eq "CB") {
    $count_weight = $current_weight ;
}
 elsif ( $current_countability eq "UB" || $current_countability eq "UC") {
    $noncount_weight = $current_weight ;
}
 elsif ( $current_countability eq "UP") {
    $UP_weight = $current_weight ;
}
 elsif ( $current_countability eq "nn") {
    $nn_weight = $current_weight ;
}
else {
    print $count_weight , " ", $noncount_weight , " " , $UP_weight, " " , $nn_weight , " "
        , $current_countability , "\n";
    die "Error - should never get here !!!!!!!!!" ;
}


########3 print $current_word , " ", $best_countability , "\ n";

while (<>) {

    chomp;
    @tags = split ;

    if ( $tags [1] eq $current_word ) {

        if ( $tags [2] eq "CO" || $tags [2] eq "CB") {
            $count_weight += $tags [0];
        }
         elsif ( $tags [2] eq "UB" || $tags [2] eq "UC") {
```

```
            $noncount_weight += $tags [0];
        }
        elsif ( $tags [2] eq "UP") {
            $UP_weight += $tags [0];
        }
        elsif ( $tags [2] eq "nn") {
            $nn_weight += $tags [0];
        }
        else {
            die "Error - should never get here !!!!!!!!!" ;
        }

        ###### if ( $tags [0] > $highest_count ) { ###### <---------- FIX
            ###########
        ######       $current_countability = $tags [0];
        ###### }
    }
    else {
        $best_countability = "";

        if ( $count_weight >= $noncount_weight && $count_weight >= $UP_weight &&
            $count_weight >= $nn_weight ){
            $best_countability = "COUNT";
        }
        elsif ( $noncount_weight >= $count_weight && $noncount_weight >= $UP_weight
            && $noncount_weight >= $nn_weight ){
            $best_countability = "NON";
        }
        elsif ( $UP_weight >= $count_weight && $UP_weight >= $noncount_weight &&
            $UP_weight >= $nn_weight ){
            $best_countability = "UP";
        }
        elsif ( $nn_weight >= $count_weight && $nn_weight >= $noncount_weight &&
            $nn_weight >= $UP_weight ) {
            $best_countability = "nn";
        }
        else {
            $best_countability = "tie";
        }

        if ( $best_countability eq "tie" ) { print $current_word , " ", $count_weight
            , " ", $noncount_weight , " ", $UP_weight, " ", $nn_weight , " ",
            $best_countability , "\n"; }
        else { print $current_word , " ", $best_countability , "\n";}

        $current_countability = $tags [2];
        $current_word = $tags [1];

        $count_weight = 0;
        $noncount_weight = 0;
        $UP_weight = 0;
        $nn_weight = 0;
```

```
            if ( $current_countability eq "CO" || $current_countability eq "CB") {
                $count_weight = $current_weight ;
            }
            elsif ( $current_countability eq "UB" || $current_countability eq "UC") {
                $noncount_weight = $current_weight ;
            }
            elsif ( $current_countability eq "UP") {
                $UP_weight = $current_weight ;
            }
            elsif ( $current_countability eq "nn") {
                $nn_weight = $current_weight ;
            }
            else {
                print $count_weight , " " , $noncount_weight , " " , $UP_weight, " " ,
                      $nn_weight , " " , $current_countability , "\n";
                die "Error - should never get here !!!!!!!!!" ;
            }

        }

}
```

Listing for test_NTT_majority.pl

```
#!/usr/bin/perl -w


$current_word = "";
 $best_countability = "";
$highest_count = 0;


$_ = <STDIN>;
chomp;
@tags = split ;
 $highest_count = $tags [0];
$current_word = $tags [1];
 $best_countability = $tags [2];

# print $current_word , " ", $best_countability , "\ n";

while (<>) {

    chomp;
    @tags = split ;

    if ( $tags [1] eq $current_word ) {
        if ( $tags [0] > $highest_count ) {
            $highest_count = $tags [0];
             $best_countability = $tags [2];
        }
    }
    else {
        print $current_word , "_" , $best_countability , "\n";

        $highest_count = $tags [0];
        $current_word = $tags [1];
         $best_countability = $tags [2];
    }

}
```

Listing for word_net_distro.pl

```
#!/usr/bin/perl -w



while (<>) {

    @tags = split ;

    # print $tags [0], " ", $tags [1], " ", $tags [1]/65290, " ", $tags [2], " ", $tags
        [2]/109680, "\ n";

    if ( $tags[1]/65290 < $tags[2]/109680) {print $tags [0], "\n";}

}
```

Bibliography

Valerio Allegranza. “Determiners as Functors: NP Structure in Italian.” In *Romance in Head-driven Phrase Structure Grammar*, CSLI Lecture Notes, Vol. 75, CSLI Publications, Stanford. 1998.

Francis Bond and Caitlin Vatikiotis-Bateson. “Using an Ontology to Determine English Countability.” To appear at *COLING 2002 Conference.* 2002.

Francis Bond. *Determiners and Number in English contrasted with Japanese, as exemplified in Machine Translation.* PhD thesis. University of Queensland. Decemeber 2001.

Francis Bond and Satoru Ikehara. “When and how to disambiguate? — countability in machine translation —.” In *International Seminar on Multimodal Interactive Disambiguation: MIDDIM-96*, p. 149-160, Grenoble, 1996.

Francis Bond, Kentaro Ogura, and Satoru Ikehara. “Classifiers in Japanese-to-English machine translation.” In *16th International Conference on Computational Linguistics: COLING-96*, Copenhagen

Francis Bond, Kentaro Ogura, and Satoru Ikehara. “Countability and number in Japanese-to-English machine translation.” In *15th International Conference on Computational Linguistics: COLING-94*, Kyoto, 1994.

Michael Brent. “From grammar to lexicon: unsupervised learning of lexical syntax.” *Computational Linguistics*, 19(2): 242-262, 1993.

Ted Briscoe and John Carroll. “Automatic Extraction of Subcategorization from Corpora.” In *Proceedings of the 5th Conference on Applied Natural Language Processing (ANLP-97)*, Washington, DC, USA. 1997.

Ted Briscoe and John Carroll. “Generalised probabilistic LR parsing of natural language (corpora) with unification-based grammars.” Computational Linguistics, 19(1). 1993.

Ted Briscoe and Ann Copestake. “Sense extensions as lexical rules.” In *Proceedings of the IJCAI Workshop on Computational Approaches to Non-Literal Language*, Sydney, Australia, 1991.

Ann Copestake. “The representation of group denoting nouns in a lexical knowledge base.” In *Computational Lexical Semantics*, P. St. Dizier and E. Viegas (editors), Cambridge University Press, 1995.

Ralph Grishman, Catherine Macleod, and Adam Meyers. “Comlex Syntax: Building a Computational Lexicon.” Presented at Coling 1994, Kyoto. (http://xxx.lanl.gov/abs/cmp-lg/9411017).

Rodney Huddleston. *English grammar: an outline.* Cambridge University Press, 1988.

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

Satoru Ikehara, Satoshi Shirai, Akio Yokoo, and Hiromi Nakaiwa. “Toward an MT system without pre-editing — the effects of new methods in ALT-J/E —” In *Third Machine Translation Summit: MT Summit III*. Washington, D.C. 1991.

Paul Kay. “The emergence of basic color lexicons hypothesis.” In *The Language of Color in the Mediterranean*, Alexander Borg (ed.). Stolkholm: Almqvist and Wiksell International, 1999.

Geoffrey Leech. “A Brief Users’ Guide to the grammatical tagging of the British National Corpus.” http://www.hcu.ox.ac.uk/BNC/what/gramtag.html, 1997.

Catherine Macleod and Ralph Grishman. *COMLEX Syntax Reference Manual.* New York University. 1993.

Christopher Manning. “Automatic acquisition of a large subcategorization dictionary from corpora.” In *Proceedings of the 31st Annual Meeting of the Assn. for Computational Linguistics*, p. 235-242, Columbus, OH, 1993.

R. Quirk, S. Greenbaum, G. Leech and J. Svartvik. *A Comprehensive Grammar of the English Language.* London and New York, Longman. 1985.

Cristina Schmitt and Alan Munn. “Bare Nominals, Morphosyntax, and the Nominal Mapping Parameter.”
http://www.msu.edu/user/amunn/psfiles/barenominals.pdf, 22 Dec 2000.

Cristina Schmitt and Alan Munn. “Bare Nouns and the Morphosyntax of Number.” In *Proceedings of the Linguistic Symposium on Romance Languages*, 1999.

Cristina Schmitt and Alan Munn. “Against the nominal mapping parameter: bare nouns in Brazilian Portuguese.” In *Proceedings of NELS 29*, 1998.

Mona Singh. “The Perfective Paradox: Or How to Eat Your Cake and Have it Too.” In *Proceedings of the Berkeley Linguistic Society*, 1992.

Jan van Eijck. “Capita Selecta in Natural Language Semantics.” Contributions to the *Encyclopedia of Language and Linguistics*, 1991.

Claire Warwick. “What is the BNC?” http://www.hcu.ox.ac.uk/BNC/what/, 1997.

Susanne Rohen Wolff, Catherine Macleod, and Adam Meyers. *COMLEX Word Classes Manual.* New York University. 1994.