<table>
<thead>
<tr>
<th>Title</th>
<th>A plethora of methods for learning English countability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author(s)</td>
<td>Baldwin, Timothy; Bond, Francis</td>
</tr>
<tr>
<td>Date</td>
<td>2003</td>
</tr>
<tr>
<td>URL</td>
<td><a href="http://hdl.handle.net/10220/6822">http://hdl.handle.net/10220/6822</a></td>
</tr>
<tr>
<td>Rights</td>
<td>© 2003 ACL. This is the author created version of a work that has been peer reviewed and accepted for publication by Proceedings of 2003 Conference on Empirical Methods in Natural Language Processing: EMNLP 2003, Association for Computational Linguistics. It incorporates referee’s comments but changes resulting from the publishing process, such as copyediting, structural formatting, may not be reflected in this document. The published version is available at: [DOI: <a href="http://dx.doi.org/10.3115/1119355.1119365">http://dx.doi.org/10.3115/1119355.1119365</a>].</td>
</tr>
</tbody>
</table>
A Plethora of Methods for Learning English Countability

Timothy Baldwin
CSLI
Stanford University
Stanford, CA 94305 USA
tbaldwin@csli.stanford.edu

Francis Bond
NTT Communication Science Laboratories
Nippon Telegraph and Telephone Corporation
Kyoto, Japan
bond@cslab.kecl.ntt.co.jp

Abstract
This paper compares a range of methods for classifying words based on linguistic diagnostics, focusing on the task of learning countabilities for English nouns. We propose two basic approaches to feature representation: distribution-based representation, which simply looks at the distribution of features in the corpus data, and agreement-based representation which analyses the level of token-wise agreement between multiple pre-processor systems. We additionally compare a single multiclass classifier architecture with a suite of binary classifiers, and combine analyses from multiple pre-processors. Finally, we present and evaluate a feature selection method.

1 Introduction
Lexical acquisition can be described as the process of populating a grammar skeleton with lexical items, through a process of mapping word lemmata onto lexical types described in the grammar. Depending on the linguistic precision of the base grammar, lexical acquisition can range in complexity from simple part-of-speech tagging (shallow lexical acquisition) to the acquisition of selectionally-constrained subcategorisation frame clusters or constructional compatibilities (deep lexical acquisition). Our particular interest is in the latter task of deep lexical acquisition with respect to English nouns.

We are interested in developing learning techniques for deep lexical acquisition which take a fixed set of linguistic diagnostics, and classify words according to corpus data. We propose a range of general techniques for this task, as exemplified over the task of English countability acquisition. Countability is the syntactic property that determines whether a noun can take singular and plural forms, and affects the range of permissible modifiers. Many nouns have both countable and uncountable lemmas, with differences in meaning: I submitted two papers "documents" (countable) vs. Please use white paper "substance to be written on" (uncountable).

This research complements that described in Baldwin and Bond (2003), where we present the linguistic foundations and features drawn upon in the countability classification task, and motivate the claim that countability preferences can be learned from corpus evidence. In this paper, we focus on the methods used to tackle the task of countability classification based on this fixed feature set.

The remainder of this paper is structured as follows. Section 2 outlines the countability classes, resources and pre-processors. Section 3 presents two methods of representing the feature space. Section 4 details the different classifier designs and the dataset, which are then evaluated in Section 5. Finally, we conclude the paper with a discussion in Section 6.

2 Preliminaries
In this section, we describe the countability classes, the resources used in this research, and the feature extraction method. These are described in greater detail in Baldwin and Bond (2003).

2.1 Countability classes
Nouns are classified as belonging to one or more of four possible classes: countable, uncountable, plural only and bipartite. **Countable** nouns can be modified by denumerators, prototypically numbers, and have a morphologically marked plural form: *one dog, two dogs*. **Uncountable** nouns cannot be modified by denumerators, but can be modified by unspecific quantifiers such as *much*; they do not show any number distinction (prototypically being singular): *one equipment, some equipment,* two equipments. **Plural only** nouns only have a plural form, such as *goods,* and cannot be either denumerated or modified by *much;* many plural only nouns, such as *clothes,* use the plural form even as modifiers: a clothes horse. **Bipartite** nouns are plural when they head a noun phrase *(trousers)*, but generally singular when used as a modifier *(trouser leg)*; they can
be denumerated with the classifier pair: a pair of scissors.

2.2 Gold standard data

Information about noun countability was obtained from two sources: COMLEX 3.0 (Grishman et al., 1998) and the common noun part of ALT-JE’s Japanese-to-English semantic transfer dictionary (Ikehara et al., 1991). Of the approximately 22,000 noun entries in COMLEX, 13,622 are marked as countable, 710 as uncountable and the remainder are unmarked for countability. ALT-JE has 56,245 English noun types with distinct countability.

2.3 Feature space

Features used in this research are divided up into feature clusters, each of which is conditioned on the occurrence of a target noun in a given construction. Feature clusters are either one-dimensional (describing a single multivariate feature) or two-dimensional (describing the interaction between two multivariate features), with each dimension describing a lexical or syntactic property of the construction in question. An example of a one-dimensional feature cluster is head noun number, i.e. the number (singular or plural) of the target noun when it occurs as the head of an NP; an example of a two-dimensional feature cluster in subject–verb agreement, i.e. the number (singular or plural) of the target noun when it occurs as head of a subject NP vs. number agreement on the verb (singular or plural). Below, we provide a basic description of the 11 feature clusters used in this research and their dimensionality. A 1-dimensional feature cluster with \( x \) unit features, \( x = 2 \) (2-dimensional feature cluster with \( x \times y \) unit features). These represent a total of 206 unit features.

- **Head noun number:** \( [2] \) the number of the target noun when it heads an NP
- **Modifier noun number:** \( [2] \) the number of the target noun when a modifier in an NP
- **Subject–verb agreement:** \( [2 \times 2] \) the number of the target noun in a subject position vs. number agreement on the governing verb
- **Coordinate noun number:** \( [2 \times 2] \) the number of the target noun vs. the number of the head nouns of conjuncts
- **N of N constructions:** \( [11 \times 2] \) the type of the \( N_1 \) (e.g. collective, temporal) vs. the number of the target noun \( N_2 \) in an \( N_1 of N_2 \) construction
- **Occurrence in PPs:** \( [52 \times 2] \) the preposition type vs. the presence or absence of a determiner when the target noun occurs in singular form in a PP
- **Pronoun co-occurrence:** \( [2 \times 2] \) what personal, possessive and reflexive pronouns (e.g. he, their, itself) occur in the same sentence as singular and plural instances of the target noun
- **Singular determiners:** \( [10] \) what singular-selecting determiners (e.g. a, much) occur in NPs headed by the target noun in singular form
- **Plural determiners:** \( [12] \) what plural-selecting determiners (e.g. many, various) occur in NPs headed by the target noun in plural form
- **Non-bounded determiners:** \( [11 \times 2] \) what non-bounded determiners (e.g. more, sufficient) occur in NPs headed by the target noun, and what is the number of the target noun for each

2.4 Feature extraction

The values for the features described above were extracted from the written component of the British National Corpus (BNC, Burnard (2000)) using three different pre-processors: (a) a POS tagger, (b) a full-text chunker and (c) a dependency parser. These are used independently to test the efficacy of the different systems at capturing features used in the classification process, and in tandem to consolidate the strengths of the individual methods.

With the POS extraction method, we first tagged the BNC using an fnTBL-based tagger (Ngai and Florian, 2001) trained over the Brown and WSJ corpora and based on the Penn POS tagset. We then lemmatised this data using a Penn tagset-customised version of morph (Minnen et al., 2001). Finally, we implemented a range of high-precision, low-recall POS-based templates to extract out the features from the processed data.

For the chunker, we ran fnTBL over the lemmatised tagged data, training over CoNLL 2000-style (Tjong Kim Sang and Buchholz, 2000) chunk-converted versions of the full Brown and WSJ corpora. For the NP-internal features (e.g. determiners, head number), we used the noun chunks directly, or applied POS-based templates locally within noun chunks. For inter-chunk features (e.g. subject–verb agreement), we looked at only adjacent chunk pairs so as to maintain a high level of precision.

We read dependency tuples directly off the output of RASP (Briscoe and Carroll, 2002b) in grammatical relation mode. RASP has the advantage that recall is high, although precision is potentially lower.

\(^1\) We used the first parse in the experiments reported here. An alternative method would be to use weighted dependency tuples, as described in Briscoe and Carroll (2002a).
than chunking or tagging as the parser is forced into
resolving phrase attachment ambiguities and com-
mitting to a single phrase structure analysis.

After generating the different feature vectors for
each noun based on the above configurations, we fil-
ter out all nouns which did not occur at least 10
times in NP head position in the output of all three
systems. This resulted in a total of 20,530 nouns,
of which 9,031 are contained in the combined COM-
LEX and ALT-J/E lexicons. The evaluation is based
on these 9,031 nouns.

3 Feature representation

We test two basic feature representations in this re-
search: distribution-based, which simply looks at
the relative occurrence of different features in the
corpus data, and agreement-based, which analyses
the level of token-wise agreement between multiple
systems.

3.1 Distribution-based feature representation

In the distribution-based feature representation, we
take each target noun in turn and compare its amal-
gamated value for each unit feature with (a) the val-
dues for other target nouns, and (b) the value of other
unit features within that same feature cluster. That
is, we focus on the relative prominence of features
globally within the corpus and locally within each
feature cluster.

In the case of a one-dimensional feature cluster
(e.g. singular determiners), each unit feature \( f_s \)
for target noun \( w \) is translated into 3 separate feature values:

\[
\text{corpfreq}(f_s, w) = \frac{\text{freq}(f_s, w)}{\text{freq}(\ast)}
\]

\[
\text{wordfreq}(f_s, w) = \frac{\text{freq}(f_s, w)}{\text{freq}(w)}
\]

\[
\text{featfreq}(f_s, w) = \frac{\text{freq}(f_s, w)}{\sum_i \text{freq}(f_i, w)}
\]

where \( \text{freq}(\ast) \) is the frequency of all words in the
corpus. That is, for each unit feature we capture the rel-
ative corpus frequency, frequency relative to the tar-
get word frequency, and frequency relative to other
features in the same feature cluster. Thus, for an \( n \)-
valued one-dimensional feature cluster, we generate
\( 3n \) independent feature values.

In the case of a two-dimensional feature ma-
trix (e.g. subject-position noun number vs. verb
number agreement), each unit feature \( f_{s,t} \) for
target noun \( w \) is translated into \( \text{corpfreq}(f_{s,t}, w) \),
\( \text{wordfreq}(f_{s,t}, w) \) and \( \text{featfreq}(f_{s,t}, w) \) as above,
and 2 additional feature values:

\[
\text{featsdmfreq}(f_{s,t}, w) = \frac{\text{freq}(f_{s,t}, w)}{\sum_i \text{freq}(f_{i,t}, w)}
\]

which represent the \( \text{featfreq} \) values calculated along
each of the two feature dimensions. Additionally,
we calculate cumulative totals for each row and
column of the feature matrix and describe each as
for the one-dimensional features above (in the form
of 3 values). Thus, for an \( m \times n \)-valued two-
dimensional feature cluster, we generate a total of
\( 5mn + 3(m + n) \) independent feature values.

The feature clusters produce a combined total of
1284 individual feature values.

3.2 Agreement-based feature representation

The agreement-based feature representation con-
siders the degree of token agreement between the
features extracted using the three different pre-
processors. This allows us to pinpoint the reliable di-
agnostics within the corpus data and filter out noise
generated by the individual pre-processors.

It is possible to identify the features which
are positively-correlated with a unique countability
class (e.g. occurrence of a singular noun with the
determiner \( a \) occurs only for countable nouns), and
for each to determine the token-level agreement be-
tween the different systems. The number of diagno-
tsics considered for each of the countability classes
is: 32 for countable nouns, 19 for uncountable nouns
and 1 for each of plural only and bipartite nouns.
The total number of diagnostics we test agreement
across is thus 53.

The token-level correlation for each feature \( f_s \) is
calculated fourfold according to relative agreement.
the \( \kappa \) statistic, correlated frequency and correlated
weight. The relative agreement between systems
\( sys_1 \) and \( sys_2 \) wrt \( f_s \) for target noun \( w \) is defined to be:

\[
\text{agr}(f_s, w)(sys_1, sys_2) = \frac{|\text{tok}(f_s, w)(sys_1) \cap \text{tok}(f_s, w)(sys_2)|}{|\text{tok}(f_s, w)(sys_1) \cup \text{tok}(f_s, w)(sys_2)|}
\]

where \( \text{tok}(f_s, w)(sys_i) \) returns the set of token
instances of \( (f_s, w) \). The \( \kappa \) statistic (Carletta, 1996)
is recast as:

\[
\kappa(f_s, w)(sys_1, sys_2) = \frac{\text{agr}(f_s, w)(sys_1, sys_2) - \sum_{i=1}^n \beta_{sys_1, sys_2}(f_s, w)}{1 - \sum_{i=1}^n \beta_{sys_1, sys_2}(f_s, w)}
\]

In this modified form, \( \kappa(f_s, w) \) represents the diver-
gence in relative agreement wrt \( f_s \) for target noun \( w \),
relative to the mean relative agreement wrt \( f_s \) over
all words. Correlated frequency is defined to be:

\[
\text{cfreq}(f_s, w)(sys_1, sys_2) = \frac{|\text{tok}(f_s, w)(sys_1) \cap \text{tok}(f_s, w)(sys_2)|}{\text{freq}(w)}
\]
It describes the occurrence of tokens in agreement for \((f_s, w)\) relative to the total occurrence of the target word.

The metrics are used to derive three separate feature values for each diagnostic over the three preprocessor system pairings. We additionally calculate the mean value of each metric across the system pairings and the overall correlated weight for each countability class \(C\) as:

\[
cw_{(C, w)}(sys_1, sys_2) = \frac{\sum_{i \in c'}|tok(f_s, w)(sys_1) \cap tok(f_s, w)(sys_2)|}{\sum_{i} |tok(f_s, w)(sys_1) \cap tok(f_s, w)(sys_2)|}
\]

Correlated weight describes the occurrence of correlated features in the given countability class relative to other correlated features.

We test agreement: (a) for each of these diagnostics individually and within each countability class (Agree\((Token, \ast)\)), and (b) across the amalgam of diagnostics for each of the countability classes (Agree\((Class, \ast)\)). For Agree\((Token, \ast)\), we calculate agr, \(\kappa\) and cfreq values for each of the 53 diagnostics across the 3 system pairings, and additionally calculate the mean value for each value. We additionally calculate the overall \(cw\) value for each countability class. This results in a total of 640 feature values \((3 \times 53 \times 3 + 53 \times 3 + 4)\). In the case of Agree\((Class, \ast)\), we average the agr, \(\kappa\) and cfreq values across each countability class for each of the three system pairings, and also calculate the mean value in each case. We further calculate the overall \(cw\) value for each countability class, culminating in 52 feature values \((3 \times 4 \times 3 + 4 \times 3 + 4)\).

### 4 Classifier Set-up and Evaluation

Below, we outline the different classifiers tested and describe the process used to generate the gold-standard data.

#### 4.1 Classifier architectures

We propose a variety of unsupervised and supervised classifier architectures for the task of learning countability, and also a feature selection method. In all cases, our classifiers are built using TiMBL version 4.2 (Daelemans et al., 2002), a memory-based classification system based on the \(k\)-nearest neighbour algorithm. As a result of extensive parameter optimisation, we settled on the default configuration\(^2\) for TiMBL with \(k\) set to 9.\(^3\)

\(^2\)IBL with weighted overlap, gain ratio-based feature weighting and equal weighting of neighbours.

\(^3\)We additionally experimented with the kernel-based TinySVM system, but found TiMBL to be the marginally superior performer in all cases, a somewhat surprising result given the high-dimensionality of the feature space.

### Full-feature supervised classifiers

The simplest system architecture applies the supervised learning paradigm to the distribution-based feature vectors for each of the POS tagger, chunker and RASP (Dist\((POS, \ast)\), Dist\((chunk, \ast)\) and Dist\((RASP, \ast)\), respectively). For the distribution-based feature representation, we additionally combine the outputs of the three pre-processors by: (a) concatenating the individual distribution-based feature vectors for the three systems (resulting in a 3852-element feature vector: Dist\((All_{CON}, \ast)\)); and (b) taking the mean over the three systems for each distribution-based feature value (resulting in a 1284-element feature vector: Dist\((All_{MEAN}, \ast)\)). The agreement-based feature representation provides two additional system configurations: Agree\((Class, \ast)\) and Agree\((Token, \ast)\) (see Section 3.2).

Orthogonal to the issue of how to generate the feature values is the question of how to classify a given noun according to the different countability classes. The two basic options here are to either have a single classifier and define multiclases according to all observed combinations of countability classes (Dist\((\ast, SINGLE)\)), or have a suite of binary classifiers, one for each countability class (Dist\((\ast, SUITE)\)). The SINGLE classifier architecture has advantages in terms of speed (a \(4 \times \) speed-up over the classifier suite) and simplicity, but runs into problems with data sparseness for the less-commonly attested multi-classes given that a single noun can occur with multiple countabilities. The SUITE classifier architecture delineates the different countability classes more directly, but runs the risk of a noun not being classified according to any of the four classes.

### Feature-selecting supervised classifiers

We improve the performance of the basic classifiers by way of best-\(N\) filter-based feature selection. Feature selection has been shown to improve classification accuracy over a variety of tasks (Liu and Motoda, 1988), but in the case of memory-based learners such as TiMBL, has the additional advantage of accelerating the classification process and reducing memory overhead. The computational complexity of memory-based learners is proportional to the number of features, so any reduction in the feature space leads to a proportionate reduction in computational time. For tasks such as countability classification with a large number of both feature values and test instances (particularly if we are to classify all noun types in a given corpus), this speed-up is vital.
Our feature selection method uses a combined feature relevance metric to estimate the best-\(N\) features for each countability class, and then restricts the classifier to operate over only those \(N\) features. Feature relevance is estimated through analysis of the correspondence between class and feature values for a given feature, through metrics including shared variance and information gain. These individual metrics tend to be biased toward particular features: information gain and gain ratio, e.g., tend to favor features of higher cardinality (White and Liu, 1994). In order to minimise such bias, we generate a feature ranking for each feature selection metric (based on the relative feature relevance scores), and simply add the absolute ranks for each feature together. By re-ranking the features in increasing order of summed rank, we can generate a generalised feature relevance ranking. We are now in a position to prune the feature space to a pre-determined size, by taking the best-\(N\) features in the feature ranking.

The feature selection metrics we combine are those implemented in TiMBL, namely: shared variance, chi-square, information gain and gain ratio.

### Unsupervised classifier

In order to derive a common baseline for the different systems, we built an unsupervised classifier which, for each target noun, simply checks to see if any diagnostic (as used in the agreement-based feature representation) was detected for each of the countability classes; even a single occurrence of a diagnostic is taken to be sufficient evidence for membership in that countability class. Elementary system combination is achieved by voting between the three pre-processor outputs as to whether the target noun belongs to a given countability class. That is, the target noun is classified as belonging to a given countability class iff at least two of the pre-processors furnish linguistic evidence for membership in that class.

#### 4.2 Training data

Training data was generated independently for the SINGLE and SUITE classifiers. In each case, we first extracted all countability-annotated nouns from each of the ALT-J/E and COMLEX lexicons which are attested at least 10 times in the BNC, and composed the training data from these pre-filtered sets. In the case of the SINGLE classifier, we simply classified words according to the union of all countabilities from ALT-J/E and COMLEX, resulting in the following dataset:

<table>
<thead>
<tr>
<th>Class</th>
<th>Countable</th>
<th>Uncountable</th>
<th>Plural only</th>
<th>Bipartite</th>
<th>No.</th>
<th>Freq</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>4,342</td>
<td>1,476</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Uncountable</td>
<td>1,519</td>
<td>5,471</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Plural only</td>
<td>84</td>
<td>5,639</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bipartite</td>
<td>35</td>
<td>5,639</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

From this, it is evident that some class combinations (e.g. plural only+bipartite) are highly infrequent, hinting at a problem with data sparseness.

For the SUITE classifier, we generate the positive exemplars for the countable and uncountable classes from the intersection of the COMLEX and ALT-J/E data for that class; negative exemplars, on the other hand, are those not annotated as belonging to that class in either lexicon. With the plural only and bipartite data, COMLEX cannot be used as it does not describe these two classes. We thus took all members of each class listed in ALT-J/E as our positive exemplars, and all remaining nouns with non-identical singular and plural forms as negative exemplars. This resulted in the following datasets:

<table>
<thead>
<tr>
<th>Class</th>
<th>Positive data</th>
<th>Negative data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Countable</td>
<td>4,342</td>
<td>1,476</td>
</tr>
<tr>
<td>Uncountable</td>
<td>1,519</td>
<td>5,471</td>
</tr>
<tr>
<td>Plural only</td>
<td>84</td>
<td>5,639</td>
</tr>
<tr>
<td>Bipartite</td>
<td>35</td>
<td>5,639</td>
</tr>
</tbody>
</table>

### 5 Evaluation

Evaluation of the supervised classifiers was carried out based on 10-fold stratified cross-validation over the relevant dataset, and results presented here are averaged over the 10 iterations. Classifier performance is rated according to classification accuracy (the proportion of instances classified correctly) and F-score (\(\beta = 1\)). In the case of the SINGLE classifier, the class-wise F-score is calculated by decomposing the multiclass labels into their components. A countable+uncountable instance misclassified as countable, for example, would count as a misclassification in terms of classification accuracy, a correct classification in the calculation of the countable F-score, and a misclassification in the calculation of the uncountable F-score. Note that the SINGLE classifier is run over a different dataset to each member of the SUITE classifier, and cross-comparison of the classification accuracies is not representative of the relative system performance (classification accuracies for the SINGLE classifier are given in parentheses to reinforce this point). Classification accuracies are thus simply used for classifier comparison within a basic classifier architecture (SINGLE or SUITE), and F-score is
the evaluation metric of choice for overall evaluation. We present the results for two baseline systems for each countability class: a majority-class classifier and the unsupervised method. The Majority class system is run over the binary data used by the SUITE classifier for the given class, and simply classifies all instances according to the most commonly-attested class in that dataset. Irrespective of the majority class, we calculate the F-score based on a positive-class classifier, i.e. a classifier which naively classifies each instance as belonging to the given class; in the case that the positive class is not the majority class, the F-score is given in parentheses.

The results for the different system configurations over the four countability classes are presented in Tables 1–4, in which the highest classification accuracy and F-score values for each class are presented in **boldface**. The classifier Dist(AllCON,SUITE), for example, applies the distribution-based feature representation in a SUITE classifier configuration (i.e. it tests for binary membership in each countability class), using the concatenated feature vectors from each of the tagger, chunker and RASP.

Items of note in the results are:

- all system configurations surpass both the majority-class baseline and unsupervised classifier in terms of F-score
- for all other than bipartite nouns, the SUITE classifier outperforms the SINGLE classifier in terms of F-score
- the best of the distribution-based classifiers was, without exception, superior to the best of the agreement-based classifiers
- chunk-based feature extraction generally produced superior performance to POS tag-based feature extraction, which was in turn generally better than RASP-based feature extraction; statistically significant differences in F-score (based on the two-tailed t-test, \( p < .05 \)) were observed for both chunking and tagging over RASP for the plural only class, and chunking over RASP for the countable class
- for the SUITE classifier, system combination by either concatenation (Dist(AllCON,SUITE)) or averaging over the individual feature values (Dist(AllMEAN,SUITE)) generally led to a statistically significant improvement over each of the individual systems for the countable

### Table 1: Basic results for countable nouns

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Accuracy</th>
<th>F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Majority class</td>
<td>.783</td>
<td>(.357)</td>
</tr>
<tr>
<td>Unsupervised</td>
<td>.342</td>
<td>.391</td>
</tr>
<tr>
<td>Dist(POS,SUITE)</td>
<td>.945</td>
<td>.876</td>
</tr>
<tr>
<td>Dist(POS,SINGLE)</td>
<td>.850</td>
<td>.861</td>
</tr>
<tr>
<td>Dist(chunk,SUITE)</td>
<td>.945</td>
<td>.876</td>
</tr>
<tr>
<td>Dist(chunk,SINGLE)</td>
<td>.853</td>
<td>.861</td>
</tr>
<tr>
<td>Dist(RASP,SUITE)</td>
<td>.944</td>
<td>.872</td>
</tr>
<tr>
<td>Dist(RASP,SINGLE)</td>
<td>.847</td>
<td>.851</td>
</tr>
<tr>
<td>Dist(AllCON,SUITE)</td>
<td>.952</td>
<td>.892</td>
</tr>
<tr>
<td>Dist(AllCON,SINGLE)</td>
<td>.857</td>
<td>.873</td>
</tr>
<tr>
<td>Dist(AllMEAN,SUITE)</td>
<td><strong>.954</strong></td>
<td><strong>.895</strong></td>
</tr>
<tr>
<td>Agree(Token,SUITE)</td>
<td>.923</td>
<td>.825</td>
</tr>
<tr>
<td>Agree(Class,SUITE)</td>
<td>.923</td>
<td>.824</td>
</tr>
</tbody>
</table>

### Table 2: Basic results for uncountable nouns

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Accuracy</th>
<th>F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Majority class</td>
<td>.746</td>
<td>.855</td>
</tr>
<tr>
<td>Unsupervised</td>
<td>.798</td>
<td>.879</td>
</tr>
<tr>
<td>Dist(POS,SUITE)</td>
<td>.928</td>
<td>.953</td>
</tr>
<tr>
<td>Dist(POS,SINGLE)</td>
<td>.850</td>
<td>.940</td>
</tr>
<tr>
<td>Dist(chunk,SUITE)</td>
<td>.933</td>
<td>.956</td>
</tr>
<tr>
<td>Dist(chunk,SINGLE)</td>
<td>.853</td>
<td>.942</td>
</tr>
<tr>
<td>Dist(RASP,SUITE)</td>
<td>.923</td>
<td>.950</td>
</tr>
<tr>
<td>Dist(RASP,SINGLE)</td>
<td>.847</td>
<td>.940</td>
</tr>
<tr>
<td>Dist(AllCON,SUITE)</td>
<td><strong>.939</strong></td>
<td><strong>.960</strong></td>
</tr>
<tr>
<td>Dist(AllCON,SINGLE)</td>
<td>.857</td>
<td>.944</td>
</tr>
<tr>
<td>Dist(AllMEAN,SUITE)</td>
<td>.937</td>
<td>.959</td>
</tr>
<tr>
<td>Agree(Token,SUITE)</td>
<td>.902</td>
<td>.936</td>
</tr>
<tr>
<td>Agree(Class,SUITE)</td>
<td>.911</td>
<td>.941</td>
</tr>
</tbody>
</table>

### Table 3: Basic results for plural only nouns

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Accuracy</th>
<th>F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Majority class</td>
<td>.983</td>
<td>(.023)</td>
</tr>
<tr>
<td>Unsupervised</td>
<td>.411</td>
<td>.033</td>
</tr>
<tr>
<td>Dist(POS,SUITE)</td>
<td>.989</td>
<td>.558</td>
</tr>
<tr>
<td>Dist(POS,SINGLE)</td>
<td>.850</td>
<td>.479</td>
</tr>
<tr>
<td>Dist(chunk,SUITE)</td>
<td>.990</td>
<td>.568</td>
</tr>
<tr>
<td>Dist(chunk,SINGLE)</td>
<td>.853</td>
<td>.495</td>
</tr>
<tr>
<td>Dist(RASP,SUITE)</td>
<td>.989</td>
<td>.415</td>
</tr>
<tr>
<td>Dist(RASP,SINGLE)</td>
<td>.847</td>
<td>.360</td>
</tr>
<tr>
<td>Dist(AllCON,SUITE)</td>
<td><strong>.990</strong></td>
<td><strong>.582</strong></td>
</tr>
<tr>
<td>Dist(AllCON,SINGLE)</td>
<td>.857</td>
<td>.500</td>
</tr>
<tr>
<td>Dist(AllMEAN,SUITE)</td>
<td>.990</td>
<td>.575</td>
</tr>
<tr>
<td>Agree(Token,SUITE)</td>
<td>.988</td>
<td>.409</td>
</tr>
<tr>
<td>Agree(Class,SUITE)</td>
<td>.988</td>
<td>.401</td>
</tr>
</tbody>
</table>

### Table 4: Basic results for bipartite nouns

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Accuracy</th>
<th>F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Majority class</td>
<td>.984</td>
<td>(.012)</td>
</tr>
<tr>
<td>Unsupervised</td>
<td>.931</td>
<td>.137</td>
</tr>
<tr>
<td>Dist(POS,SUITE)</td>
<td>.997</td>
<td>.752</td>
</tr>
<tr>
<td>Dist(POS,SINGLE)</td>
<td>.850</td>
<td>.857</td>
</tr>
<tr>
<td>Dist(chunk,SUITE)</td>
<td><strong>.997</strong></td>
<td>.704</td>
</tr>
<tr>
<td>Dist(chunk,SINGLE)</td>
<td>.853</td>
<td>.865</td>
</tr>
<tr>
<td>Dist(RASP,SUITE)</td>
<td>.997</td>
<td>.700</td>
</tr>
<tr>
<td>Dist(RASP,SINGLE)</td>
<td>.847</td>
<td>.798</td>
</tr>
<tr>
<td>Dist(AllCON,SUITE)</td>
<td>.996</td>
<td>.723</td>
</tr>
<tr>
<td>Dist(AllCON,SINGLE)</td>
<td>.857</td>
<td>.730</td>
</tr>
<tr>
<td>Dist(AllMEAN,SUITE)</td>
<td><strong>.997</strong></td>
<td><strong>.710</strong></td>
</tr>
<tr>
<td>Agree(Token,SUITE)</td>
<td><strong>.997</strong></td>
<td><strong>.710</strong></td>
</tr>
<tr>
<td>Agree(Class,SUITE)</td>
<td><strong>.997</strong></td>
<td><strong>.695</strong></td>
</tr>
</tbody>
</table>
Figure 1: Effects of feature selection

and uncountable classes, but there was no statistical difference between these two architectures for any of the 4 countability classes; for the SINGLE classifier, system combination (Dist(AllICON,SUITE)) did not lead to a significant performance gain.

To evaluate the effects of feature selection, we graphed the F-score value and processing time (in instances processed per second) over values of N from 25 to the full feature set. We targeted the Dist(AllICON,SUITE) system for evaluation (3852 features), and ran it over both the countable and uncountable classes. We additionally carried out random feature selection as a baseline to compare the feature selection results against. Note that the domain feature selection as a baseline to compare the features), and ran it over both the countable and uncountable classes, thus allows us to maintain the full classification potential of the feature set while enabling a speedup greater than an order of magnitude, potentially making the difference in practical utility for the proposed method.

To determine the relative impact of the component feature values on the performance of the distribution-based feature representation, we used the Dist(AllMEAN,SUITE) configuration to build: (a) a classifier using a single binary value for each unit feature, based on simple corpus occurrence (Binary); and (b) 3 separate classifiers based on each of the corpfreq, wordfreq, and featfreq features values only (without the 2D feature cluster totals). In each case, the total number of feature values is 206.

The results for each of these classifiers over countable and uncountable nouns are presented in Table 5, as compared to the basic Dist(AllMEAN,SUITE) classifier with all 1,284 features (All features) and also the best-200 features (Best-200). Results which differ from those for All features to a level of statistical significance are asterisked. The binary classifiers performed significantly worse than All features for both countable and uncountable nouns, underlining the utility of the distribution-based feature representation. wordfreq is marginally superior to corpfreq as a standalone feature representation, and both of these were on the whole slightly below the full feature set in performance (although no significant difference was observed). featfreq performed slightly worse again, significantly below the level of the full feature set. Results for the best-200 classifier were marginally better than those for each of the individual feature representations in the case of the countable class, but marginally below the results for corpfreq and wordfreq in the case of the uncountable class. The differences here are not statistically significant, and additional evaluation is required to determine the relative success of feature selection over simply using wordfreq values, for example.

6 Discussion

There have been at least three earlier approaches to the automatic determination of countability: two using semantic cues and one using cor-

---

Table 5: Results for restricted feature sets

<table>
<thead>
<tr>
<th>Feature space</th>
<th>COUNTABLE</th>
<th></th>
<th>UNCOUNTOBLE</th>
<th>Acc</th>
<th>F-score</th>
<th>Acc</th>
<th>F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>All features</td>
<td>.937</td>
<td>.959</td>
<td>.934</td>
<td>.954</td>
<td>.895</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Best-200</td>
<td>.934</td>
<td>.956</td>
<td>.949</td>
<td>.894</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Binary</td>
<td>.904*</td>
<td>.931</td>
<td>.930*</td>
<td>.833</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Corpus freq</td>
<td>.929</td>
<td>.954</td>
<td>.952</td>
<td>.889</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Word freq</td>
<td>.933</td>
<td>.956</td>
<td>.954</td>
<td>.896</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature freq</td>
<td>.928</td>
<td>.952*</td>
<td>.934*</td>
<td>.869*</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

*No significant performance difference was observed for: Dist(ChunMEAN,SUITE) vs. Dist(All,SUITE) for countable nouns, and Dist(POSCON,SUITE) vs. Dist(AllICON,SUITE) for uncountable nouns.

*As evaluated on an AMD Athlon 2100+ CPU with 3GB of memory.

*We focus exclusively on countable and uncountable nouns here and in the remainder of supplementary evaluation as these are by far the most populous countability classes.
pora. Bond and Vatikiotis-Bateson (2002) determine a noun’s countability preferences—as defined in a 5-way classification—from its semantic class in the ALT-J/E lexicon, and show that semantics predicts countability 78% of the time. O’Hara et al. (2003) implemented a similar approach using the much larger Cyc ontology and achieved 89.5% accuracy, mapping onto the 2 classes of countable and uncountable. Schwartz (2002) learned noun countabilities by looking at determiner occurrence in singular noun chunks and was able to tag 11.7% of BNC noun tokens as countable and 39.5% as uncountable, achieving a noun type agreement of 88% and 44%, respectively, with the ALT-J/E lexicon. Our results compare favourably with each of these.

In a separate evaluation, we took the best-performing classifier (\( \text{Dist(Alt\text{-}CON, SUITE)} \)) and ran it over open data, using best-500 feature selection (Baldwin and Bond, 2003). The output of the classifier was evaluated relative to hand-annotated data, and the level of agreement found to be around 92.4%, which is approximately equivalent to the agreement between COMLEX and ALT-J/E of 93.8%.

In conclusion, we have presented a plethora of learning techniques for deep lexical acquisition from corpus data, and applied each to the task of classifying English nouns for countability. We specifically compared two feature representations, based on relative feature occurrence and token-level classification, and two basic classifier architectures, using a suite of binary classifiers and a single multi-class classifier. We also analysed the effects of combining the output of multiple pre-processors, and presented a simple feature selection method. Overall, the best results were obtained using a distribution-based suite of binary classifiers combining the output of multiple pre-processors.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. BCS-0094638 and also the Research Collaboration between NTT Communication Science Laboratories, Nippon Telegraph and Telephone Corporation and CSLI, Stanford University. We would like to thank Leonoor van der Beek, Slaven Bilac, Ann Copestake, Ivan Sag and the three anonymous reviewers for their valuable input on this research, and John Carroll for providing access to RASP.

References


