

Empirical Approaches to Multilingual Lexical Acquisition

Lecturer: Timothy Baldwin



THE UNIVERSITY OF
MELBOURNE

Lecture 5

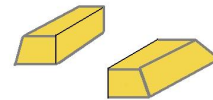
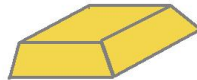
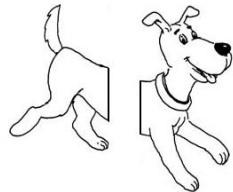
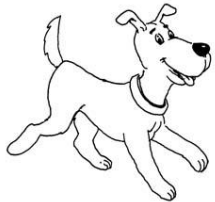
Monolingual Countability Learning

Background

- Countability is a syntactic property of the noun phrase in languages such as English, Dutch, Albanian and Tagalog
- In generation used to decide between:
a cake, cake, a piece of cake
- In analysis, helps to resolve ambiguity:
 - ★ *I need a paper by this evening* (academic/newspaper)
 - ★ *I need some paper by this evening* (material)
 - ★ *I need the paper by this evening* (ambiguous)

Noun Phrase Countability

- Semantically motivated:
 - ★ bounded, indivisible **individuals** (+b)
prototypically COUNTABLE: *a dog, two dogs*
 - ★ unbounded, divisible **substances** (-b)
prototypically UNCOUNTABLE: *gold*



Countability Classes

- **countable:** *book, button, person* (one book, two books)
- **uncountable:** *equipment, gold, wood* (*one equipment, much equipment, *two equipments)
- **plural only:** *clothes, manners, outskirts* (*one clothes, clothes horse)
- **bipartite:** *glasses, scissors, trousers* (*one scissors, scissor kick, pair of scissors)

Applications

- Determination of countability of unknown nouns (e.g. *acyclovir*, *coagulopathy*)
- Detection of countability anomalies in multiword expressions (e.g. *public relations*, *cat's cradle*)
- Extraction of English determinerless PPs (e.g. *by bus*, *at sea*)
- Key component of noun type hierarchy in precision grammars (e.g. ERG, Alpino)

Learning the Countability of English Nouns from Corpus Data

(Baldwin and Bond 2003a)

Learning Countability

- **Observation:** the countability properties of a noun type are reflected in its corpus token occurrences:

*... Cezanne snarling like a **dog** and then ...*

*... doing an impression of a rabid **dog**.*

*... with a pack of **dogs** running beside them.*

*Amnesty International has received **information** ...*

*Recent **information** from former detainees ...*

*... researchers often uncover **information** ...*

Case in Point

Acyclovir is a specifically anti-viral drug ...

Acyclovir has been developed and marketed by ...

Acyclovir given intravenously, ...

Coagulopathy is a well recognised complication ...

... may explain why *coagulopathy* after shunting is ...

... could stimulate a *coagulopathy* ...

... is also probably responsible for a *coagulopathy* ...

... a patient with a *coagulopathy*.

Methodology

- Identify lexical and/or constructional features associated with each countability class
- Determine the relative corpus occurrence of the features for each noun
- Use the noun feature vectors to classify the noun as a member of each of the countability classes, training from gold-standard countability data

Feature Clusters

Head noun number:^{1D} target noun number as head of NP (e.g. *a shaggy dog = SINGULAR*)

Modifier noun number:^{1D} target noun number as modifier in NP (e.g. *dog food = SINGULAR*)

Subject–verb agreement:^{2D} target noun number as subject vs. verb number agreement (e.g. *the dog barks = <SINGULAR,SINGULAR>*)

Coordinate noun number:^{2D} target noun number vs. the number of the head nouns of conjuncts (e.g. *dogs and mud = <PLURAL,SINGULAR>*)

N₁ of N₂ constructions:^{2D} number of N₂ vs. type of N₁ (e.g. *the type of dog* = $\langle \text{TYPE, SINGULAR} \rangle$); total of 11 N₁ types for use in this feature cluster (e.g. COLLECTIVE, LACK, TEMPORAL).

Occurrence in PPs:^{2D} the presence or absence of a determiner ($\pm \text{DET}$) in singular head complement of PP (e.g. *per dog* = $\langle \text{per, -DET} \rangle$).

Pronoun co-occurrence:^{2D} what pronouns occur in the same sentence as singular and plural instances (e.g. *The dog ate its dinner* = $\langle \text{its, SINGULAR} \rangle$). Approximation of pronoun co-indexation.

Singular determiners:^{1D} singular-selecting determiners (e.g. *a dog* = a). Two types: countable (e.g. *another, each*), uncountable (e.g. *much, little*).

Plural determiners:^{1D} plural-selecting determiners (e.g. few *dogs* = few).

Non-bounded determiners:^{2D} non-bounded determiner vs. noun number (e.g. more dogs = $\langle \underline{more}, \underline{PLURAL} \rangle$).

Feature Values

$$1D \quad \text{corpfreq}(f_s, w) = \frac{\text{freq}(f_s|w)}{\text{freq}(*)} \quad (1)$$

$$\text{wordfreq}(f_s, w) = \frac{\text{freq}(f_s|w)}{\text{freq}(w)} \quad (2)$$

$$\text{featfreq}(f_s, w) = \frac{\text{freq}(f_s|w)}{\sum_i \text{freq}(f_i|w)} \quad (3)$$

$$2D \quad \text{featdimfreq}_1(f_{s,t}, w) = \frac{\text{freq}(f_{s,t}|w)}{\sum_i \text{freq}(f_{i,t}|w)} \quad (4)$$

$$\text{featdimfreq}_2(f_{s,t}, w) = \frac{\text{freq}(f_{s,t}|w)}{\sum_j \text{freq}(f_{s,j}|w)} \quad (5)$$

1-D case

$corpfreq(f_1, w)$

$wordfreq(f_1, w)$

$featfreq(f_1, w)$

1-D case



corpfreq(f_2, w)

wordfreq(f_2, w)

featfreq(f_2, w)

2-D case

$corpfreq(f_{1,1}, w)$

$wordfreq(f_{1,1}, w)$

$featfreq(f_{1,1}, w)$

2-D case



$featdimfreq_1(f_{1,1}, w)$

2-D case

$featdimfreq_2(f_{1,1}, w)$	

2-D case

$corpfreq(f_{*,1}, w)$

$wordfreq(f_{*,1}, w)$

$featfreq(f_{*,1}, w)$

2-D case

corpfreq($f_{1,*}, w$)
wordfreq($f_{1,*}, w$)
featfreq($f_{1,*}, w$)

Feature Value Extraction

- POS tagging and templates
 - ★ extract features with regexp-base templates
- Full text chunking
 - ★ conservative inter-chunk attachment disambiguation
- Robust parsing (RASP)
- Concatenated feature values from three systems

POS Tagger-based Feature Extraction: Example

country_NN fund_NNS offer_VBP an_DT easy_JJ way_NN
to_TO get_VB a_DT taste_NN of_IN foreign_JJ stock_NNS
without_IN the_DT hard_JJ research_NN of_IN seek_VBG
out_RP individual_JJ company_NNS ...

POS Tagger-based Feature Extraction: Example

country_NN fund_NNS offer_VBP an_DT easy_JJ way_NN
to_TO get_VB a_DT taste_NN of_IN foreign_JJ stock_NNS
without_IN the_DT hard_JJ research_NN of_IN seek_VBG
out_RP individual_JJ company_NNS ...

POS Tagger-based Feature Extraction: Example

country_NN fund_NNS offer_VBP an_DT easy_JJ way_NN
to_TO get_VB a_DT taste_NN of_IN foreign_JJ stock_NNS
without_IN the_DT hard_JJ research_NN of_IN seek_VBG
out_RP individual_JJ company_NNS ...

Chunker-based Feature Extraction: Example

[*NP* country_NN fund_NNS] [*VP* offer_VBP] [*NP* an_DT easy_JJ
way_NN] [*VP* to_TO get_VB] [*NP* a_DT taste_NN] [*PP* of_IN]
[*NP* foreign_JJ stock_NNS] [*PP* without_IN] [*NP* the_DT
hard_JJ research_NN] [*PP* of_IN] [*VP* seek_VBG] [*PartP* out_RP]
[*NP* individual_JJ company_NNS] [*O* .-.]

Chunker-based Feature Extraction: Example

[*NP* country_NN fund_NNS] [*VP* offer_VBP] [*NP* an_DT easy_JJ
way_NN] [*VP* to_TO get_VB] [*NP* a_DT taste_NN] [*PP* of_IN]
[*NP* foreign_JJ stock_NNS] [*PP* without_IN] [*NP* the_DT
hard_JJ research_NN] [*PP* of_IN] [*VP* seek_VBG] [*PartP* out_RP]
[*NP* individual_JJ company_NNS] [*O* .-.]

Chunker-based Feature Extraction: Example

[*NP* country_NN fund_NNS] [*VP* offer_VBP] [*NP* an_DT easy_JJ
way_NN] [*VP* to_TO get_VB] [*NP* a_DT taste_NN] [*PP* of_IN]
[*NP* foreign_JJ stock_NNS] [*PP* without_IN] [*NP* the_DT
hard_JJ research_NN] [*PP* of_IN] [*VP* seek_VBG] [*PartP* out_RP]
[*NP* individual_JJ company_NNS] [*O* .-.]

RASP-based Feature Extraction: Example

```
(|ncsubj| |offer:3_VV0| |fund+s:2_NN2| _)  
(|ncsubj| |get:8_VV0| |way:6_NN1| _)  
(|xcomp| |to| |offer:3_VV0| |get:8_VV0|)  
(|dobj| |offer:3_VV0| |way:6_NN1|)  
(|det| |way:6_NN1| |an:4_AT1|)  
(|ncmod| _ |way:6_NN1| |easy:5_JJ|)  
(|ncmod| _ |fund+s:2_NN2| |Country:1_NNL1|)
```

⋮

RASP-based Feature Extraction: Example

```
(|ncsubj| |offer:3_VV0| |fund+s:2_NN2| _)  
(|ncsubj| |get:8_VV0| |way:6_NN1| _)  
(|xcomp| |to| |offer:3_VV0| |get:8_VV0|)  
(|dobj| |offer:3_VV0| |way:6_NN1|)  
(|det| |way:6_NN1| |an:4_AT1|)  
(|ncmod| _ |way:6_NN1| |easy:5_JJ|)  
(|ncmod| _ |fund+s:2_NN2| |Country:1_NNL1|)
```

⋮

Classifier architecture

- **Training data:** generated from combination of **ALT-J/E** and **COMLEX** (5,943 common nouns in BNC)
 - ★ positive examples in both **ALT-J/E** and **COMLEX**
 - ★ negative examples in neither **ALT-J/E** nor **COMLEX**
- **Test data:** nouns with ≥ 10 BNC instances for all 3 methods (20,530 common nouns)
- Four binary supervised classifiers, one per countability class (learned using TiMBL and k -NN)

Cross-validated Countability Results

- Good results (particularly for countable and uncountable nouns), well above the baseline accuracy in each case
- Best results for combined method (concatenation of three pre-processors)

Manual Evaluation over Open Data

- Very high classifier precision relative to lexicons
- Manually annotated 100 nouns from the test data:
 - ★ classifiers agree with corpus as well as lexicons

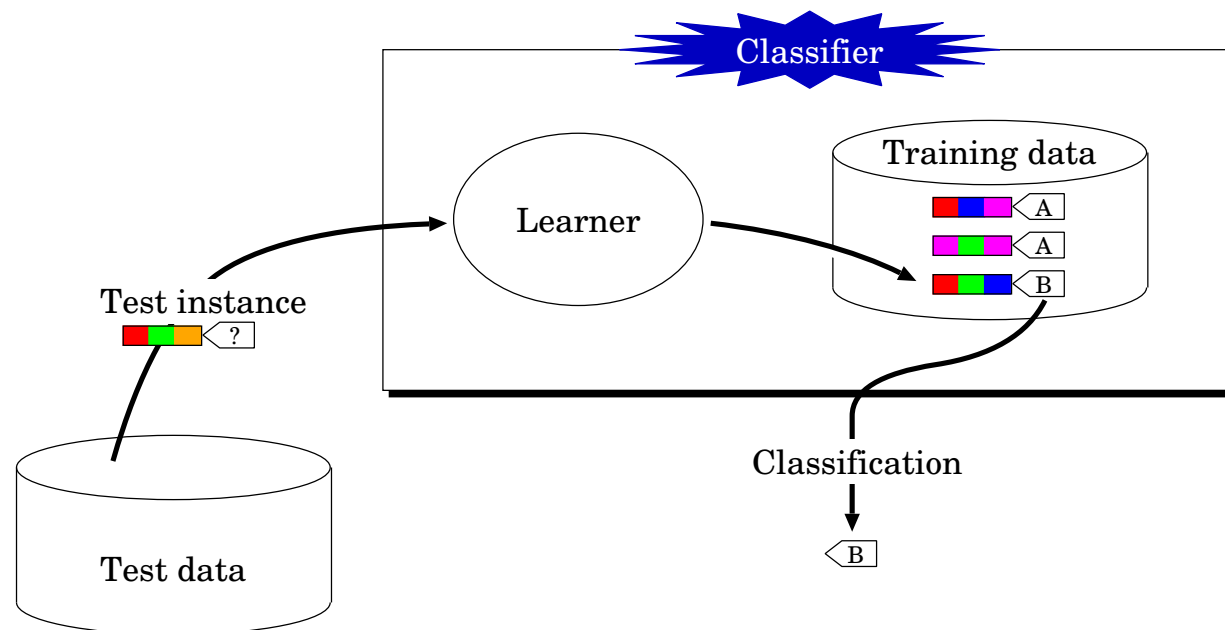
Reflections

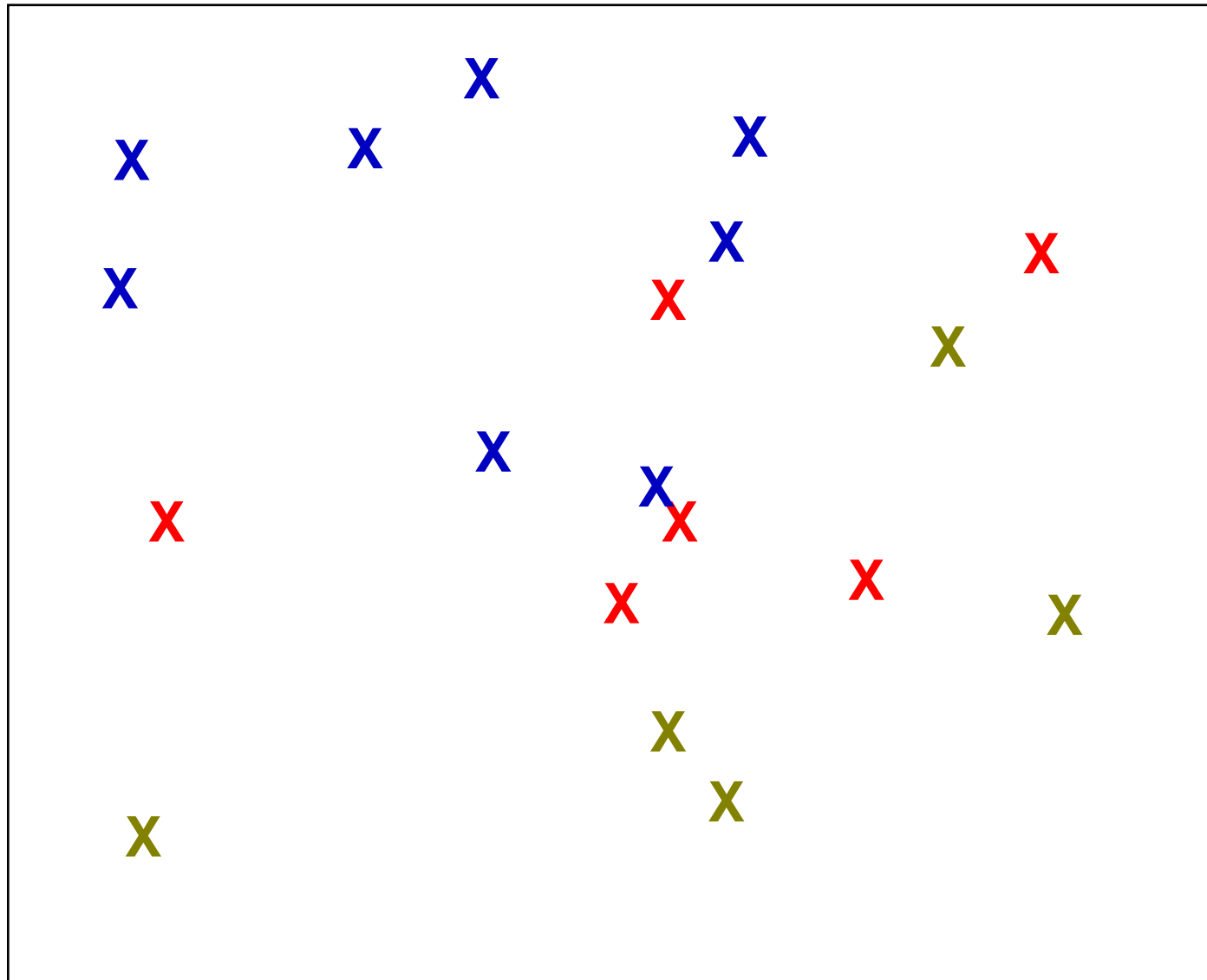
- Impressive results, but still room for improvement (particularly for the less-populated countability classes)
- Boundary between motivated countabilities and conversions (e.g. *chicken vs. elephant vs. dog*)
- Difficulties caused by MWEs (e.g. *cat's cradle*)
- Sense and frequency effects (e.g. *information*)

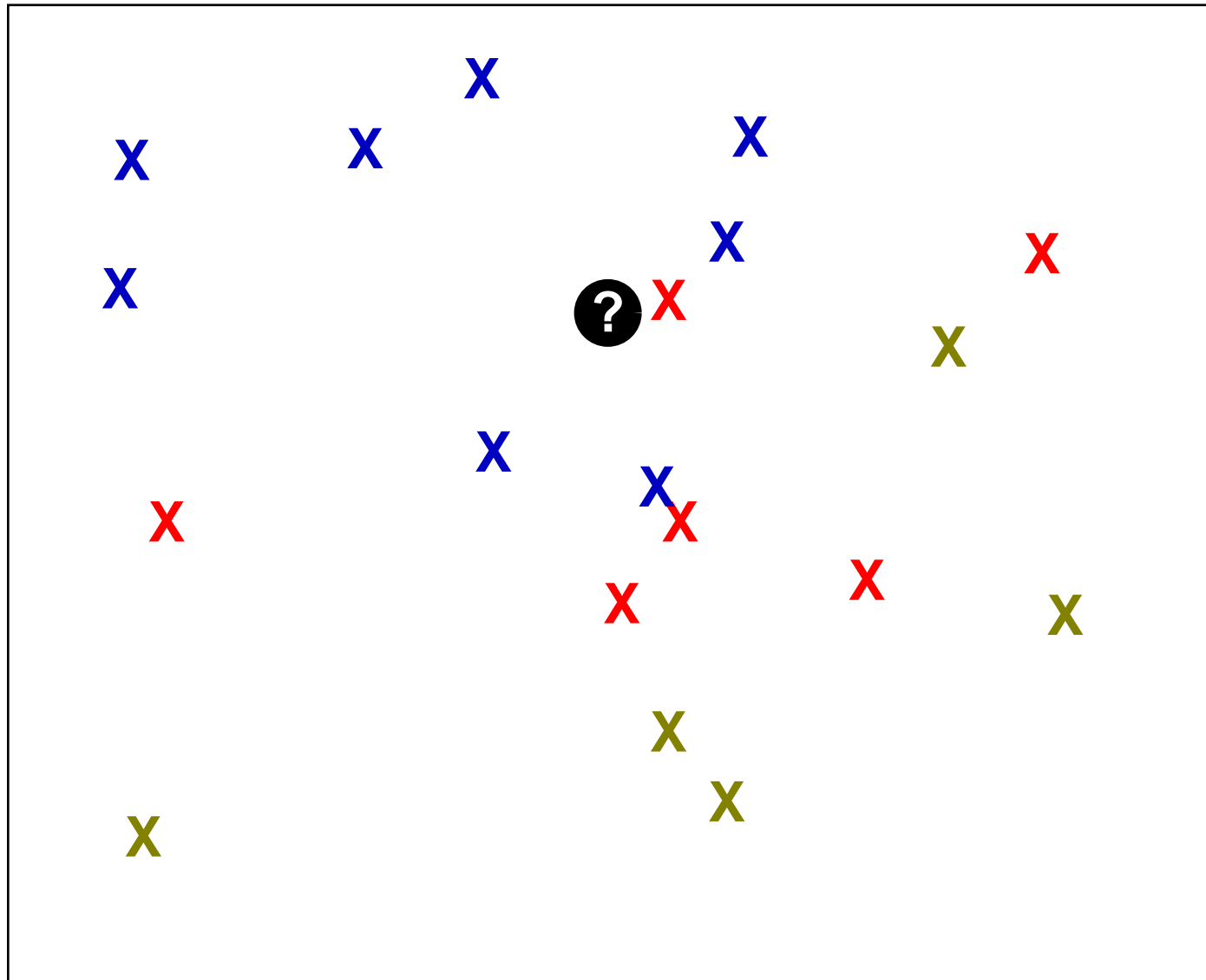
Instance-based Learning

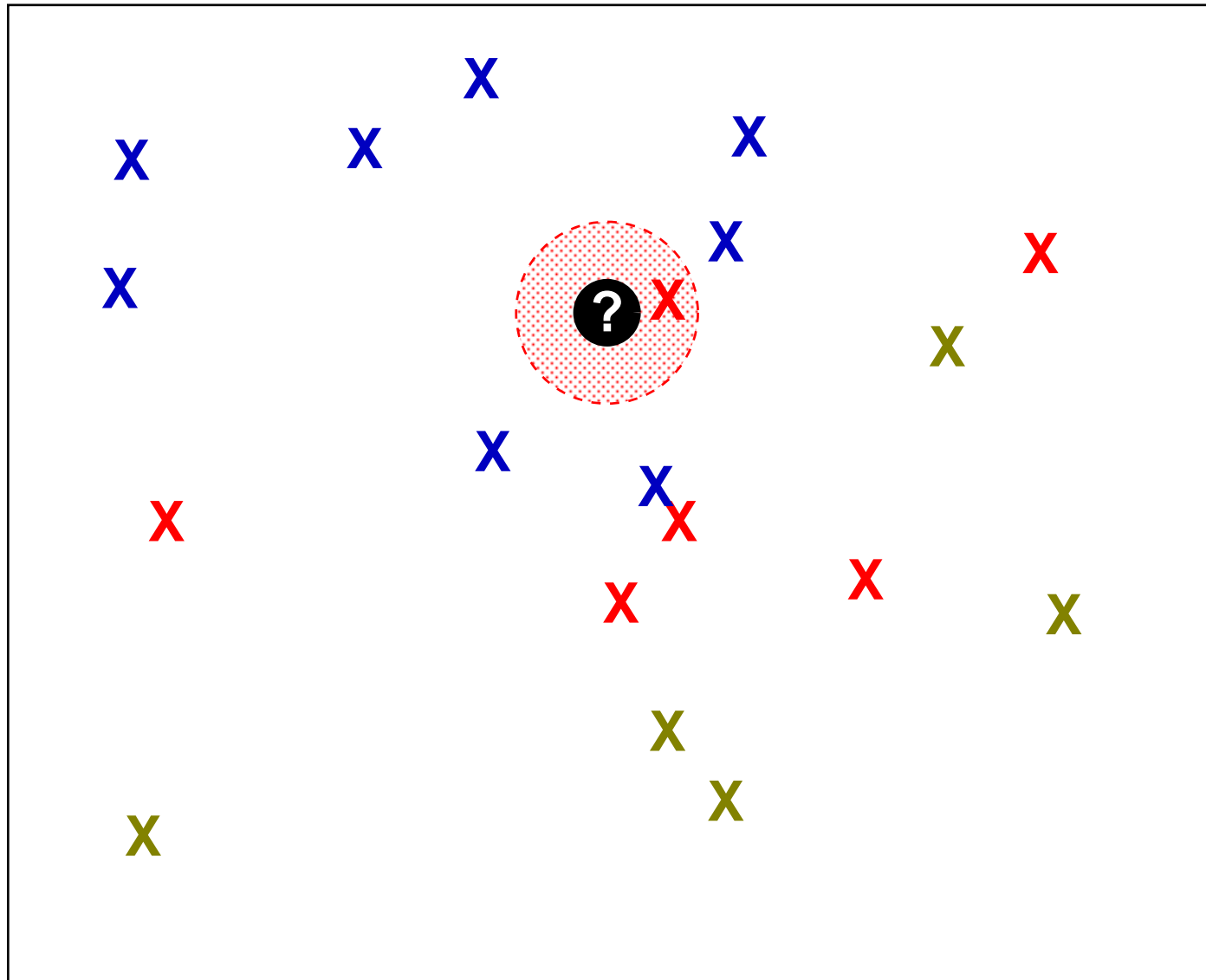
The “Lazy” View of Learning

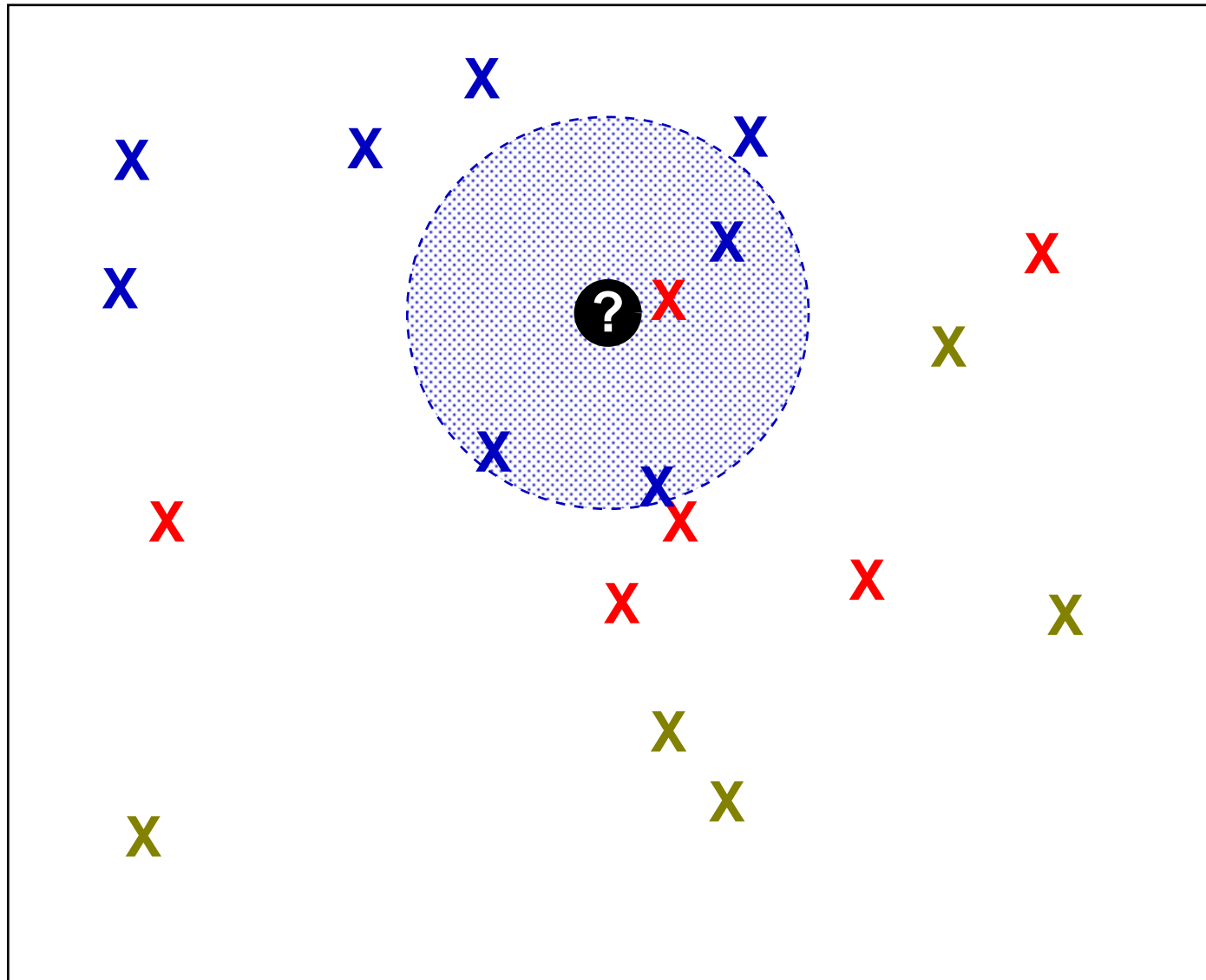
- In “lazy” learning, we map a test instance directly onto the training instances, and classify it according to the most similar training instance(s) (= **nearest neighbour(s)**):











Instance-based Learning

- Basic process:
 1. calculate the distance between the test instance and each training instances
 2. predict the class for test instance based on the class distribution of the k most similar instances (= k **nearest neighbours** or k -**NN**)
- Claim that lazy learning is superior to eager learning because “forgetting is harmful” in some applications
- Also known as **memory-based learning** or **analogy-based learning**

Eager vs. Lazy Learning

- Eager learning attempts to “learn” a structural representation of the classification task, whereas lazy learning doesn’t use any explicit structure
- There is no training phase in lazy learning
- Training can be time-consuming in eager learning, but testing is generally cheap (esp. parametric methods); the testing time with lazy learning is directly proportional to the number of training instances
- **Warning:** the eager/lazy distinction is a cline not a dichotomy

Core Components of Instance-based Learning

1. training data
2. a distance metric to compute inter-instance distance
3. a setting for k (the number of neighbours to retrieve)

Distance Metrics

- The distance between instances $X = \langle x_1, x_2, \dots, x_N \rangle$ and $Y = \langle y_1, y_2, \dots, y_N \rangle$ can be calculated via:

1. Euclidean distance:

$$D(X, Y) = \sqrt{\sum_{i=1}^N (x_i - y_i)^2}$$

2. Manhattan distance:

$$D(X, Y) = \sum_{i=1}^N |x_i - y_i|$$

3. Overlap metric:

$$D(X, Y) = \sum_{i=1}^N \frac{|x_i - y_i|}{\max_{\alpha}(\alpha_i) - \min_{\alpha}(\alpha_i)}$$

where α is each instance in the training/test data

- For nominal attributes, we generally use the distance metric:

$$\delta(x_i, y_i) = \begin{cases} 0 & \text{if } x_i = y_i \\ 1 & \text{if } x_i \neq y_i \end{cases}$$

- It is also possible to generate continuous distances over nominal attributes, using the **Modified Value Difference Metric (MVDM)**:

$$\delta(V_1, V_2) = \sum_{i=1}^N |P(C_i|V_1) - P(C_i|V_2)|$$

Distance Metrics in Action

- For the following instance pair:

	Outlook	Temperature	Humidity	Windy	Play
A:	sunny	69	normal	FALSE	yes
B:	sunny	75	normal	TRUE	(yes)

- Euclidean distance:

$$D(A, B) = \sqrt{0 + (69 - 75)^2 + 0 + 1} = \sqrt{36 + 1} \approx 6.08$$

- Manhattan distance:

$$D(A, B) = 0 + |69 - 75| + 0 + 1 = 6 + 1 = 7$$

- Overlap metric-based distance:

$$D(A, B) = 0 + \frac{|69 - 75|}{85 - 64} + 0 + 1 = \frac{6}{21} + 1 \approx 1.29$$

- Overlap metric-based distance with MVDM:

$$D(A, B) = 0 + \frac{|69 - 75|}{85 - 64} + 0 + \left(\left| \frac{2}{7} - \frac{2}{3} \right| + \left| \frac{5}{7} - \frac{1}{3} \right| \right) \approx 1.05$$

Continuous Weather Dataset

Outlook	Temperature	Humidity	Windy	Play
TRAINING DATA				
sunny	85	high	FALSE	no
sunny	80	high	TRUE	no
overcast	83	high	FALSE	yes
rainy	70	high	FALSE	yes
rainy	68	normal	FALSE	yes
rainy	65	normal	TRUE	no
overcast	64	normal	TRUE	yes
sunny	72	high	FALSE	no
sunny	69	normal	FALSE	yes
rainy	75	normal	FALSE	yes
TEST DATA				
sunny	75	normal	TRUE	(yes)

Handling Noisy Attributes

- Instance-based learning is particularly susceptible to the effects of noise, motivating some combination of **feature selection** (see next lecture) and **feature weighting**
- With feature weighting, we simply weight the distance calculation for each feature:

$$D(X, Y) = \sum_{i=1}^N w_i \delta(x_i, y_i)$$

- Common methods for feature weighting are:
 - ★ information gain
 - ★ gain ratio
 - ★ χ^2
- In calculating the feature weights, we discretise each feature (but don't use the discretisation when classifying the data)
- It is also possible to have class-specific weights

- Feature weights based on information gain:

$$w_i = H(C) - \sum_{v \in V_i} P(v) \times H(C|v)$$

$$IG(\text{outlook}|R) = 0.247$$

$$IG(\text{temperature}|R) = 0.029$$

$$IG(\text{humidity}|R) = 0.152$$

$$IG(\text{windy}|R) = 0.048$$

Full weather.nominal Dataset

	Outlook	Temperature	Humidity	Windy	Play
a:	sunny	hot	high	FALSE	no
b:	sunny	hot	high	TRUE	no
c:	overcast	hot	high	FALSE	yes
d:	rainy	mild	high	FALSE	yes
e:	rainy	cool	normal	FALSE	yes
f:	rainy	cool	normal	TRUE	no
g:	overcast	cool	normal	TRUE	yes
h:	sunny	mild	high	FALSE	no
i:	sunny	cool	normal	FALSE	yes
j:	rainy	mild	normal	FALSE	yes
k:	sunny	mild	normal	TRUE	yes
l:	overcast	mild	high	TRUE	yes
m:	overcast	hot	normal	FALSE	yes
n:	rainy	mild	high	TRUE	no

Attribute Weighting in Action

- Overlap metric-based distance:

$$D(A, B) =$$

$$0.32 \times 0 + 0.97 \times \frac{|69 - 75|}{85 - 64}$$

$$+ 0.13 \times 0 + 0.09 \times 1$$

$$= 0.97 \times \frac{6}{21} + 0.09$$

$$\approx 0.37$$

Feature	Weight
Outlook	0.32
Temperature	0.97
Humidity	0.13
Windy	0.09

Weights based on Information Gain (with equal-width discretisation over 20 intervals)

Voting Strategies

- There are two basic strategies for coming up with the set of neighbours:
 1. retrieve k nearest **instances**
 2. retrieve instances at k nearest **distances**
- There are then a number of voting strategies:
 1. give each neighbour equal weight (= classify according to the **majority class**)
 2. weight the vote of each instance by its **inverse linear distance**

from the test instance:

$$w_j = \begin{cases} \frac{d_k - d_j}{d_k - d_1} & \text{if } d_k \neq d_1 \\ 1 & \text{if } d_k = d_1 \end{cases}$$

where d_1 is the nearest neighbour, and d_k is the furthest neighbour

3. weight the vote of each instance by its **inverse distance** from the test instance:

$$w_j = \frac{1}{d_j + \epsilon}$$

Voting Strategies in Action

- majority class voting:

$$\underline{\text{yes}} = 3 \text{ vs. } \text{no} = 1$$

Instance	Class	Distance
d_1	no	0
d_2	yes	1
d_3	yes	1.5
d_4	yes	2

- ILD-based voting:

$$\text{yes} = \left(\frac{1}{2} + \frac{1}{4} + 0\right)$$

$$\text{vs. } \underline{\text{no}} = 1$$

- ID-based voting:

$$\text{yes} = \left(\frac{1}{1.5} + \frac{1}{2} + \frac{1}{2.5}\right) \text{ vs.}$$

$$\underline{\text{no}} = \frac{1}{0.5}$$

Breaking Ties

- In the case that we have an equal number of votes for a given class, we need some tie breaking mechanism:
 - ★ random tie breaking
 - ★ take class with highest prior probability
 - ★ see if the addition of the $k + 1$ th instance(s) breaks the tie

Choosing the Value of k

- Smaller values of k tend to bias the classifier performance due to noise
- Larger values of k tend to bias the classifier performance toward Zero-R performance
- Generally trial and error over the training data is the only way of getting k just right

Note: k is generally set to an odd value ... why?

Theoretical Properties of Instance-based Learning

- **Multiclass** classification method
- **Non-parametric**
 - always have to store all instances
- **Incremental**
 - easy to add extra data to the classifier on the fly
- Handles both **nominal** and **continuous** features
- No abstraction over the data

- No training phase (“lazy learning”)
- Simple (→ [generally] fast)
- Able to model arbitrarily-shaped decision boundaries

Practical Properties of Instance-based Learning

- Consistent performer
- Susceptible to noisy instances/irrelevant features (bias)
- Native handling of continuous attributes (almost more natural than handling of nominal attributes!)
- Arbitrary k value
- Overlap metric used most widely

- MVDM effective for features with small numbers of values and/or classification tasks with small numbers of classes

Using an Ontology to Determine English Countability

(Bond and Vatikiotis-Bateson 2002)

Semantic Predictability of Countability

- How far is English countability predictable from meaning?
- Countability is to some degree deterministic given the semantics of a word:

dog, pooch, canine, mongrel, ...

BUT suitcases vs. luggage, leaves vs. foliage, etc.

Case in Point

Coagulopathy: group of conditions of the blood clotting (coagulation) system in which bleeding is prolonged and excessive, a bleeding disorder

Acyclovir: antiviral drug

Word Denotation and Countability

- Knowing the referent is not enough, e.g. *scales*
 1. Thought of as being made of two arms: (British)
a pair of scales
 2. Thought of as a set of numbers: (Australian)
a set of scales
 3. Thought of as discrete whole objects: (American)
one scale/two scales

Methodology

- Take an existing ontology and determine the default countability for each synset (semantic class)
- Test how reliably defaults predict the countability of members of each synset
- Base experimentation on the **ALT-J/E** semantic transfer lexicon and ontology

Lexicon

- ALT-J/E's semantic transfer lexicon

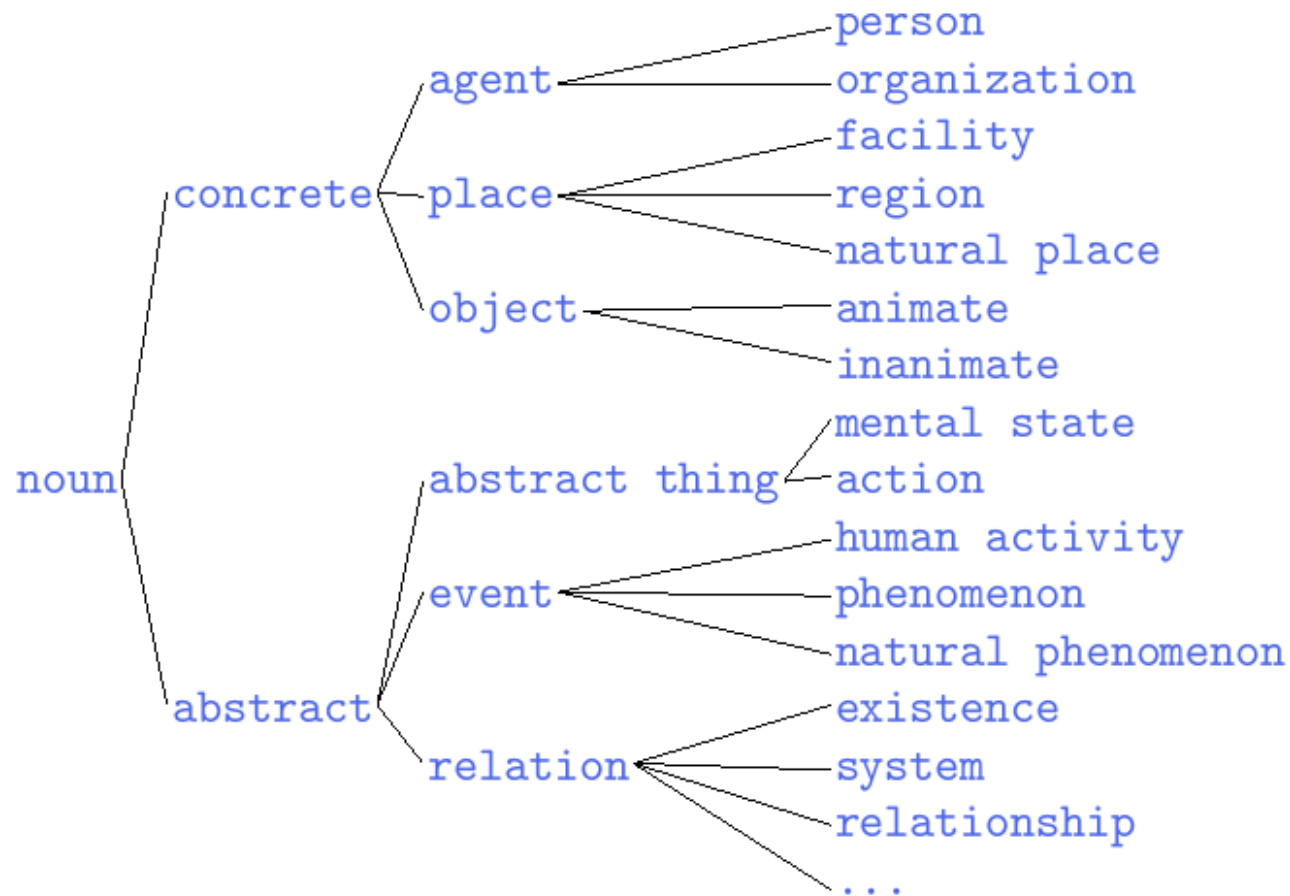
INDEX	<i>usagi</i>		
SENSE 1	ENGLISH TRANSLATION	<i>rabbit</i>	
	PART OF SPEECH	noun	
	NOUN COUNTABILITY PREF.	strongly countable	
	DEFAULT NUMBER	singular	
	SEMANTIC CLASSES	[COMMON NOUN animal, meat]	

- 71,833 linked Japanese-English noun pairs

The Goi-Taikei Ontology

- A rich ontology and wide coverage of Japanese
- Used in many NLP applications such as MT
- 2,710 semantic classes (12-level tree structure) for common nouns
- Constructed from translation pairs (without countability in mind)

Top Four Levels of Ontology



Noun Countability Preferences in ALT-J/E

Noun Countability Preference	Code	Example	Default Number	Default Classifier	#	%
fully countable	CO	<i>knife</i>	sg	—	47,255	65.8
strongly countable	BC	<i>cake</i>	sg	—	3,110	4.3
weakly countable	BU	<i>beer</i>	sg	—	3,377	4.7
uncountable	UC	<i>furniture</i>	sg	<i>piece</i>	15,435	21.5
plural only	PT	<i>scissors</i>	pl	<i>pair</i>	2,107	2.9

Experiment

- Treat every combination of semantic classes as a different semantic class.
- Most frequent NCP is assigned to all members of a class.
 - ★ Ties are resolved as follows: `fully countable` beats `strongly countable` beats `weakly countable` beats `uncountable` beats `plural only`.
- Baseline (all `fully countable` = 65.8%)

Example

- Semantic Class 910:tableware
 - ★ *crockery* ⇔ *toukirui* (UC)
 - ★ *dinner set* ⇔ *youshokki* (CO)
 - ★ *tableware* ⇔ *shokki* (UC)
 - ★ *Western-style tableware* ⇔ *youshokki* (UC)
- The most common NCP is UC
Associated **uncountable** with 910:tableware.
- This predicts the NCP correctly 75% of the time.

Discussion

- Semantics predicts countability around 78% of the time → supports hypothesis that countability is semantically motivated
- Less successful than corpus-based countability learning
- Problems of granularity/translation-orientation of lexicon
- Problems with noise in lexicon

Acknowledgements

- Thanks to Francis Bond and Caitlin Vatikiotis-Bateson for sharing their wonderful slides and graphics!

References

- BALDWIN, TIMOTHY, and FRANCIS BOND. 2003a. Learning the countability of English nouns from corpus data. In *Proc. of the 41st Annual Meeting of the ACL*, 463–70, Sapporo, Japan.
- , and —— . 2003b. A plethora of methods for learning English countability. In *Proc. of the 2003 Conference on Empirical Methods in Natural Language Processing (EMNLP 2003)*, 73–80, Sapporo, Japan.
- BOND, FRANCIS, and CAITLIN VATIKIOTIS-BATESON. 2002. Using an ontology to determine English countability. In *Proc. of the 19th International Conference on Computational Linguistics (COLING 2002)*, Taipei, Taiwan.
- DAELEMANS, WALTER, JAKUB ZAVREL, KO VAN DER SLOOT, and ANTAL VAN DEN BOSCH, 2003. *TiMBL: Tilburg Memory Based Learner, version 5.0, Reference Guide*. ILK Technical Report 03-10.
- TAN, PANG-NING, MICHAEL STEINBACH, and VIPIN KUMAR. 2006. *Introduction to Data Mining*. Addison Wesley.
- WITTEN, IAN H., and EIBE FRANK. 2005. *Data Mining: Practical Machine Learning Tools and*

Techniques with Java Implementations. San Francisco, USA: Morgan Kaufmann, second edition.