Lexical Acquisition in Statistical NLP

Adapted from:

Manning and Schütze, 1999

Chapter 8 (pp. 265-278; 308-312)

Anjana Vakil

University of Saarland

Outline

- What is lexical information?
- Why is it important for NLP?
- How can we evaluate the performance of NLP systems?
- Example: Verb Subcategorization

What is lexical information?

What is the lexicon?

> That part of the grammar of a language which includes the *lexical entries* for all the words and/or morphemes in the language and which may also include various other information, depending on the particular theory of grammar.
>
> (Trask 1993:159)

Imagine a big, detailed (machine-readable) dictionary

What/how much information? → Varies by theory

Why is it important for NLP?

Many NLP problems can be resolved by looking at lexical information, such as:

- Verb subcategorization
- Attachment ambiguity
- Selectional preferences
- Semantic similarity between words

Why is it important for NLP?

Couldn't we just write a lexicon with the relevant info?

- Building dictionaries by hand is expensive!
- Quantitative information is missing
- Contextual information is missing
- Language is always changing
  - New ideas → new words
  - Old words take on new meanings, usage patterns

How can we evaluate NLP systems?

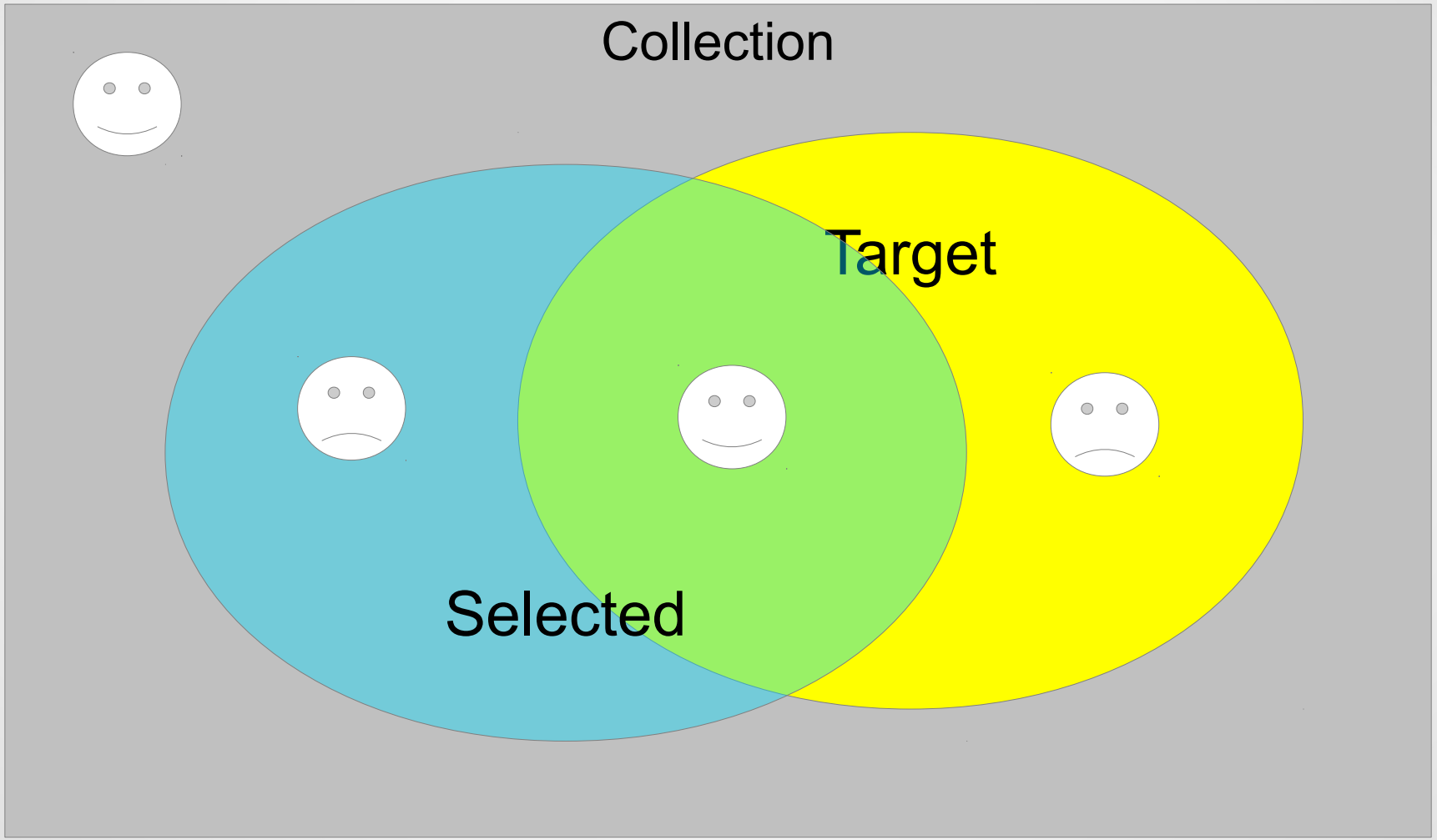
Most important: do the desired task well!

- Break it down: evaluate (& adjust) system components
- Hopefully, better component performance → better overall performance on the task

Need a convention for evaluating certain components:

precision vs. **recall**

How can we evaluate NLP systems?



How can we evaluate NLP systems?

selected, target = *tp* = true positives

selected, ~target = *fp* = false positives (Type II errors)

~selected, target = *fn* = false negatives (Type I errors)

~selected, ~target = *tn* = true negatives

How can we evaluate NLP systems?

One approach:

Just compare the number of things we got right:

$tp + tn$ (accuracy)

to the number of things we got wrong:

$fp + fn$ (error)

→ What's the problem?

Precision vs. Recall

Better questions to ask:

How many of the things we found were correct?

precision = *tp* / (*tp* + *fp*) = *tp* / |selected|

How many of the things we were supposed to find did we actually find?

recall = *tp* / (*tp* + f*n*) = *tp* / |target|

Precision vs. Recall

Q: What could we do to get 100% recall?

A: Select everything!

Q: What would happen to precision in this case?

A: Approaches zero

Q: Which is more important, precision or recall?

A: It depends!

The *F* measure

- Combines precision & recall performance into one score

$$F = \frac{1}{\alpha \frac{1}{P} + (1-\alpha)\frac{1}{R}}$$

- α determines weighting of precision vs. recall

α:	< 0.5	= 0.5	> 0.5
Preference:	recall	equal	precision

- With equal weighting (α = 0.5), $F = \frac{2PR}{P+R}$

Exercise: Rhymes for “go”

do	grew	know
glow	though	to
throw	cow	apple
lemon	no	show
flow	sew	tomato
banana	slow	how
so	few	enough
thorough	blow	two
now	goo	orange
through	follow	crow

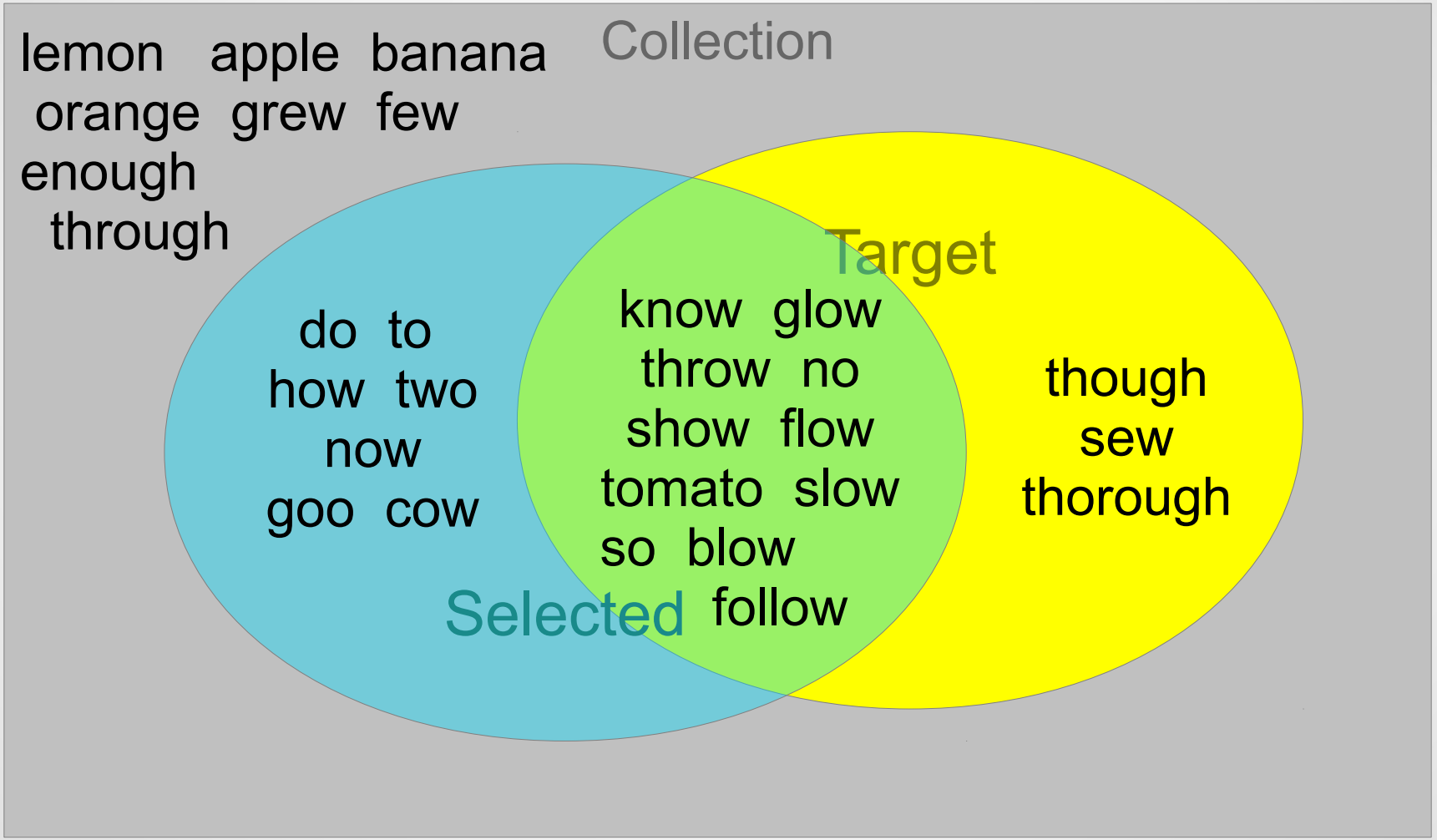
- What is the target set?
- What feature(s) should we look for?

- Select: *-o* and *-ow* words

  Calculate:

  - Precision
  - Recall
  - *F* (even P/R weights)

How can we evaluate NLP systems?



Verb Subcategorization

- Verb categories: based on semantic arguments taken

I gave	*him*	*a present.*	*I ate*	*a hamburger.*	
	RECIPIENT	THEME		THEME	
**I gave*	*him.*		**I ate*	*him*	*a hamburger.*
	RECIPIENT			RECIPIENT	THEME

Verb Subcategorization

- Categories can be divided into subcategories based on how arguments are represented syntactically

  I gave [$_{NP}$ *him*] [$_{NP}$ *a present*].

  I gave [$_{NP}$ *a present*] [$_{PP}$ *to him*].

  * *I gave* [$_{PP}$ *to him*] [$_{NP}$ *a present*].

- We call the structures a verb allows its **subcategorization frames**

  give subcategorizes for “NP NP” and “NP PP”, not “PP NP”

  (NB: subject NP left out – all English verbs require this)

Verb Subcategorization

- Why might subcategorization information be helpful?

  Parsing:

  I told her where the CoLi students eat.

  She found the table where the CoLi students eat.

- How could we acquire this information automatically?

Acquiring Verb Subcategorization Info

Brent, Michael R. 1993. “From grammar to lexicon: Unsupervised learning of lexical syntax.” *Computational Linguistics* 19:243-262

- *Lerner* system
  - Determine “cues” for certain subcat frames
  - Find verbs in corpus sentences
  - See if the word(s) following the verb fit the cue(s) for a certain frame
  - Use this to decide how likely it is that the verb allows that frame

Acquiring Verb Subcategorization Info

Table 2
Lexical categories used in the definitions of the cues.

SUBJ:	I | he | she | we | they
OBJ:	me | him | us | them
SUBJ_OBJ:	you | it | yours | hers | ours | theirs
DET:	a | an | the | her | his | its | my | our | their | your | this | that | whose
+TNS:	has | have| had | am | is | are | was | were | do | does | did | can | could | may | might | must | will | would
CC:	when | before | after | as | while | if
PUNC:	. | ? | ! | , | ; | :

Reproduced from (Brent 1993)

Acquiring Verb Subcategorization Info

Table 1
The six syntactic frames studied in this paper.

SF Description	Good Example	Bad Example
NP only	greet them	*arrive them
tensed clause	hope he'll attend	*want he'll attend
infinitive	hope to attend	*greet to attend
NP & clause	tell him he's a fool	*yell him he's a fool
NP & infinitive	want him to attend	*hope him to attend
NP & NP	tell him the story	*shout him the story

Table 3
Cues for syntactic frames. The category V is initially empty and is filled out during the first pass. "cap" stands for any capitalized word and "cap+" stands for any sequence of capitalized words.

Frame	Symbol	Cues
NP only	NP	(OBJ | SUBJ_OBJ | cap) (PUNC | CC)
Tensed Clause	cl	(that (DET | SUBJ | SUBJ_OBJ | cap+)) | SUBJ | (SUBJ_OBJ +TNS)
Infinitive VP	inf	to V
NP & clause	NPcl	(OBJ | SUBJ_OBJ | cap+) cl
NP & infinitive	NPinf	(OBJ | SUBJ_OBJ | cap+) inf
NP & NP (dat.)	NPNP	(OBJ | SUBJ_OBJ | cap+) NP

Reproduced from (Brent 1993)

Acquiring Verb Subcategorization Info

Analyze a corpus

v^i = verb you're interested in

f^j = frame you're investigating

c^j = cue you've defined for that frame

ϵ_j = probability of error for c^j

$n = C(v^i)$ = occurrences of verb in corpus

$m = C(v^i, c^j)$ = co-occurrences of verb & cue

Acquiring Verb Subcategorization Info

Hypothesis testing

H_0 = The verb *does not* permit the frame

H_1 = The verb *does* permit the frame

Assume H_0, and calculate the probability of obtaining your data if H_0 is true

$$p_E = P\left(\left(v^i\left(f^j\right)=0\right)\middle|\left(C\left(v^i, c^j\right)\right)\geqslant m\right) = \sum_{r=m}^{n}\binom{n}{r}\epsilon_j{}^r\left(1-\epsilon_j\right)^{n-r}$$

If p_E is small enough (compared to α), we can reject H_0

Exercise: Manning's implementation

Verb	Correct	Incorrect	OALD
bridge	1	1	1
burden	2		2
depict	2		3
emanate	1		1
leak	1		5
occupy	1		3
remark	1	1	4
retire	2	1	5
shed	1		2
troop	0		3

Table 8.3 Some subcategorization frames learned by Manning's system. For each verb, the table shows the number of correct and incorrect subcategorization frames that were learned and the number of frames listed in the Oxford Advanced Learner's Dictionary (Hornby 1974). Adapted from (Manning 1993).

Reproduced from (Manning and Schütze 1999, p. 274)

- Calculate precision
- Calculate recall
- What do these numbers imply about the system?
- How could we do better?