

6.863J/9.611J Natural language & computers
Competitive Grammar Writing (CGW)



Professor Robert C. Berwick
berwick@csail.mit.edu

Welcome to the Competitive Grammar Writing Exercise!

Timeline:

- Mon/Tues: We will form random teams of ~2–3 people & email you the list of your team members; CGW handout
- Weds, 2/9: Come to class w/ laptop (1/team), use Athena tools to write grammars, after more intro to the competition
- Weds-Friday, 2/9-2/11: Read handout cgw.pdf & do the Checkpoint Exercise DUE MIDNIGHT FRI
- Monday, 2/14: Come to class w/ laptop (1/team), use Athena tools to write grammars
- Weds, 2/16: Grammar development frozen; generate test data set via grammaticality judgments
- Tuesday, 2/22: Evaluation & prizes awarded

1. Okay, team, please log in

- You should use a laptop
- Log into Athena (see instructions in cgw.pdf)
- Your secret team directory – will be emailed to you

```
cd .../teams/03-turbulent-kiwi
```

- You can all edit files there
- Publicly readable & writeable
- No one else knows the secret directory name

Minimizes
permissions fuss



2. Now Write a Grammar of English

- You have 2 hours. 😊
- What does that mean????

3. Now write a grammar of English

- You have 2 hours. 😊
(actually, more)



What's a grammar?

Here's one to start with.

- 1 $S1 \rightarrow NP VP .$
- 1 $VP \rightarrow VerbT NP$
- 20 $NP \rightarrow Det N'$
- 1 $NP \rightarrow Proper$
- 20 $N' \rightarrow Noun$
- 1 $N' \rightarrow N' PP$
- 1 $PP \rightarrow Prep NP$

Terminology: Context-free grammar, Probabilistic Context-free Grammar (PCFG), Derivation rules, Nonterminals, Preterminals, POS tags, Generate a sentence

3. Now write a grammar of English

Plus initial terminal rules.

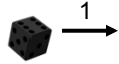
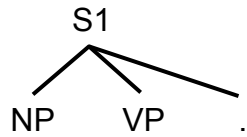
- 1 $Noun \rightarrow castle$
- 1 $Noun \rightarrow king$
- ...
- 1 $Proper \rightarrow Arthur$
- 1 $Proper \rightarrow Guinevere$
- ...
- 1 $Det \rightarrow a$
- 1 $Det \rightarrow every$
- ...
- 1 $VerbT \rightarrow covers$
- 1 $VerbT \rightarrow rides$
- ...
- 1 $Misc \rightarrow that$
- 1 $Misc \rightarrow bloodier$
- 1 $Misc \rightarrow does$
- ...

Here's one to start with.

- 1 $S1 \rightarrow NP VP .$
- 1 $VP \rightarrow VerbT NP$
- 20 $NP \rightarrow Det N'$
- 1 $NP \rightarrow Proper$
- 20 $N' \rightarrow Noun$
- 1 $N' \rightarrow N' PP$
- 1 $PP \rightarrow Prep NP$

Any PCFG is okay

3. Now write a grammar of English



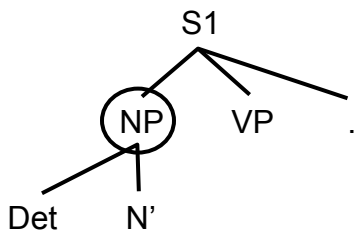
Here's one to start with.

- 1 **S1** → NP VP .
- 1 VP → VerbT NP
- 20 NP → Det N'
- 1 NP → Proper
- 20 N' → Noun
- 1 N' → N' PP
- 1 PP → Prep NP

Use ./randsent to test

Any PCFG is okay
But what is a "PCFG"?

3. Now write a grammar of English



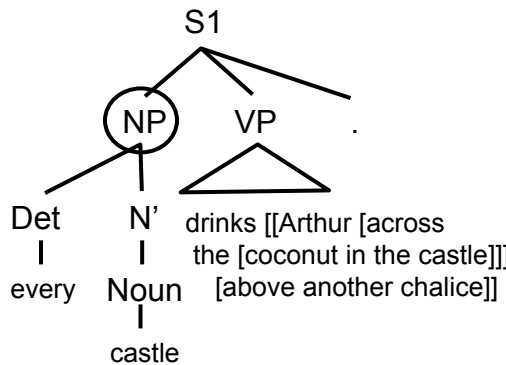
Here's one to start with.

- 1 S1 → NP VP .
- 1 VP → VerbT NP
- 20 **NP** → Det N'
- 1 **NP** → Proper
- 20 N' → Noun
- 1 N' → N' PP
- 1 PP → Prep NP

Sample a sentence
on the blackboard

Any PCFG is okay

3. Now write a grammar of English



Sample a sentence
on the blackboard

Here's one to start with.

- 1 S1 → NP VP .
- 1 VP → VerbT NP
- 20 NP → Det N'
- 1 NP → Proper
- 20 N' → Noun
- 1 N' → N' PP
- 1 PP → Prep NP

Arbitrary PCFG is okay

Initial part of speech (POS) tags given to you

- Just 6
- Noun, Det, Prep, Proper, VerbT, Misc
(noun, determiner, preposition, proper noun,
transitive verb, miscellaneous)

For instance:

The file S1_Vocab.gr has some 'suggestions' for
other POS categories...you may want to use
these, or look at what other people have done, eg

Penn Treebank project POS tags

- | | |
|--|---|
| 1. CC Coordinating conjunction | 19. PPS Possessive pronoun |
| 2. CD Cardinal number | 20. RB Adverb |
| 3. DT Determiner | 21. RBR Adverb, comparative |
| 4. EX Existential there | 22. RBS Adverb, superlative |
| 5. FW Foreign word | 23. RP Particle |
| 6. IN Preposition or subordinating conjunction | 24. SYM Symbol |
| 7. JJ Adjective | 25. TO to |
| 8. JJR Adjective, comparative | 26. UH Interjection |
| 9. JJS Adjective, superlative | 27. VB Verb, base form |
| 10. LS List item marker | 28. VBD Verb, past tense |
| 11. MD Modal | 29. VBG Verb, gerund or present participle |
| 12. NN Noun, singular or mass | 30. VBN Verb, past participle |
| 13. NNS Noun, plural | 31. VBP Verb, non-3rd person singular present |
| 14. NP Proper noun, singular | 32. VBZ Verb, 3rd person singular present |
| 15. NPS Proper noun, plural | 33. WDT Wh-determiner |
| 16. PDT Predeterminer | 34. WP Wh-pronoun |
| 17. POS Possessive ending | 35. WP\$ Possessive wh-pronoun |
| 18. PP Personal pronoun | 36. WRB Wh-adverb |

Initial S1 grammar – very simple

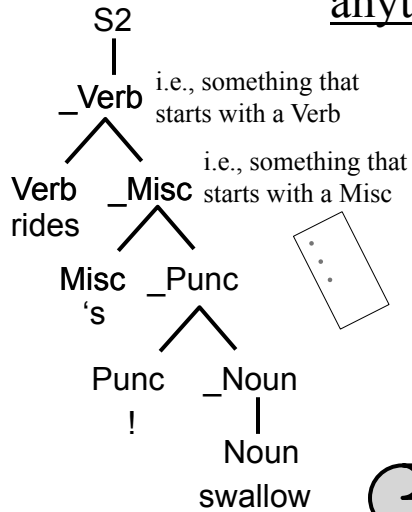
- Parses just 2 example sentences
 - Arthur is the king .
 - Arthur rides the horse near the castle .

What about the rest????

We don't want our parser just to fail (we will give you an infinite penalty for this, as we shall see), so....

We will use the backoff idea: if this grammar fails, we have a grammar S2 that never fails to parse a sentence (but it gives stupid parses)

Use of a backoff grammar: fallback to parse anything



Initial **backoff** grammar

- S2 → _Verb
- S2 → _Noun
- S2 → _Misc **(etc.)**
- _Noun → Noun
- _Noun → Noun _Noun
- _Noun → Noun _Misc
- _Misc → Misc
- _Misc → Misc _Noun
- _Misc → Misc _Misc

Use of a backoff grammar

Init. **linguistic** grammar

- S1 → NP VP .
- VP → VerbT NP
- NP → Det N'
- NP → Proper
- N' → Noun
- N' → N' PP
- PP → Prep NP

Initial **backoff** grammar

- S2 → _Verb
- S2 → _Noun
- S2 → _Misc **(etc.)**
- _Noun → Noun
- _Noun → Noun _Noun
- _Noun → Noun _Misc
- _Misc → Misc
- _Misc → Misc _Noun
- _Misc → Misc _Misc

Use of a backoff grammar

Initial **master** grammar TOP

Mixture
model

- START → S1
- START → S2

Choose these
weights wisely!

Init. **linguistic** grammar

- S1 → NP VP .
- VP → VerbT NP
- NP → Det N'
- NP → Proper
- N' → Noun
- N' → N' PP
- PP → Prep NP

Initial **backoff** grammar

- S2 → _Verb
- S2 → _Noun **(etc.)**
- S2 → _Misc
- _Noun → Noun
- _Noun → Noun _Noun
- _Noun → Noun _Misc
- _Misc → Misc
- _Misc → Misc _Noun
- _Misc → Misc _Misc

4. Okay – go!



How will
we be tested
on this?

Evaluation Procedure



How will we be tested on this?

- We'll sample 20 random sentences from your PCFG (and every other teams').
- Human judges will vote on whether each sentence is grammatical.
 - By the way, y'all will be the judges (double-blind).

this is educational

- You probably want to use the sampling script to keep testing your grammar along the way

5. Evaluation procedure

- We'll sample 20 random sentences from your PCFG
- Human judges will vote on whether each sentence is grammatical
- We'll lump all the teams' sentences together to form the "test set"
- You're right:
This only tests **precision**
- **How about recall? (productivity)**



Ok, we're done!
All our sentences are already grammatical.

• 1 S1 → NP VP .
• 1 VP → VerbT NP
• 20 NP → Det N'
• 1 NP → Proper
• 20 N' → Noun
• 1 N' → N' PP
• 1 PP → Prep NP

The two sides of the grammar building coin: the Goldilocks Principle

- Undergeneration: does not parse some of the sentences you want it to parse (or assigns some of them the wrong structure)
 - Test via program `parse`
- Overgeneration: parses some sentences that are ‘not’ English (“ungrammatical”)
 - Test via program `randsent`
 - If it generates gubbish, it’s too “lax”
 - the unable corner rides frequently above another defeater .
 - Arthur should have been covering to Arthur at Uther Pendragon for they at Arthur .
- We want grammar to be “just right”

Not so fast...development set

You should strive to get your grammar to generate (and parse) 27 sentences... ..

- Arthur is the king .
- Arthur rides the horse near the castle .
- riding to Camelot is hard .
- do coconuts speak ?
- what does Arthur ride ?
- who does Arthur suggest she carry ?
- why does England have a king ?
- are they suggesting Arthur ride to Camelot ?
- five strangers are at the Round Table .
- Guinevere might have known .
- Guinevere should be riding with Patsy .
- it is Sir Lancelot who knows Zoot !
- either Arthur knows or Patsy does .
- neither Sir Lancelot nor Guinevere will speak of it .

} covered by initial grammar
We provide a file of 27 sample sentences illustrating a range of grammatical phenomena

questions, movement, (free) relatives, clefts, agreement, subcat frames, conjunctions, auxiliaries, gerunds, sentential subjects, appositives ...

Development set

You might want your grammar to generate ...

- the Holy Grail was covered by a yellow fruit .
- Zoot might have been carried by a swallow .
- Arthur rode to Camelot and drank from his chalice .
- they migrate precisely because they know they will grow .
- do not speak !
- Arthur will have been riding for eight nights .
- Arthur , sixty inches , is a tiny king .
- Arthur knows Patsy , the trusty servant .
- Arthur and Guinevere migrate frequently .
- he knows what they are covering with that story .
- Arthur suggested that the castle be carried .
- the king drank to the castle that was his home .
- when the king drinks , Patsy drinks .

questions, movement,
(free) relatives, clefts,
agreement, subcat frames,
conjunctions, auxiliaries,
gerunds, sentential subjects,
appositives ...

How to evaluate?

- On a 'held-out' test set?
- every coconut of his that the swallow dropped sounded like a horse .
- **Precision:** how good at generating a valid sentence
- **Recall:** how good at analyzing test sentences



How should we
parse sentences
with OOV words?

No OOVs allowed
in the test set.
Fixed vocabulary.



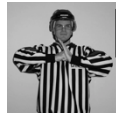
Evaluation: recall (productivity)

What we actually do, to heighten competition & creativity:
Test set comes from the participants!

You should try to generate sentences that your opponents can't parse.



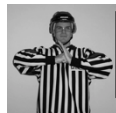
In Boggle, you get points for finding words that your opponents don't find.



Use the fixed vocabulary creatively.

The initial grammar sticks to 3rd-person singular transitive present-tense forms. All grammatical.

But we provide 183 Misc words (not accessible from initial grammar) that you're free to work into your grammar ...



Use the fixed vocabulary creatively.

Initial terminal rules

- 1 Noun castle
- 1 Noun king
- ...
- 1 Proper Arthur
- 1 Proper Guinevere
- ...
- 1 Det a
- 1 Det every
- ...
- 1 VerbT covers
- 1 VerbT rides
- ...
- 1 Misc that
- 1 Misc bloodier
- 1 Misc does
- ...

Initial terminal rules

pronouns (various cases),
 plurals,
 various verb forms,
 non-transitive verbs,
 adjectives (various forms),
 adverbs & negation,
 conjunctions & punctuation,
 wh-words,
 ...

The initial grammar sticks to 3rd-person singular transitive present-tense forms. All grammatical.

But we provide 183 Misc words (not accessible from initial grammar) that you're free to work into your grammar ...

Use the fixed vocabulary creatively.

- 1 Misc that
- 1 Misc bloodier
- 1 Misc does
- ...

Evaluation of recall (productivity)

What we actually do, to heighten competition & creativity:
 Test set comes from the participants!

You should try to generate sentences that your opponents can't parse.

We'll score your cross-entropy when you try to parse the sentences that the other teams generate.
 (Only the ones judged grammatical.)

- You probably want to use the parsing script to keep testing your grammar along the way.

What's 'cross-entropy'?

- Well, remember what entropy is?
- Let X be a random variable; and P the probability function s.t. $P(x_i) = \text{prob}(X=x_i)$
- The entropy of X (or P) is defined as:
$$H(P) = H(X) = -\sum_i p(x_i) \log_2(p(x_i))$$
- This is the minimum # of bits to encode values of x_i under this distribution p (the 'best possible coding')
- Idea of cross-entropy: p denotes the 'true' probability distribution for the language; but we encode using a (probably different) *model* m
- Q : how far away are we from the *true* distribution?

Cross-entropy measures how close we are to the 'true' probability function

- $H(p,m) = -\sum_i p(x_i) \log_2(m(x_i))$
- Note when $p=m$, this means model is in effect identical to the true language encoding
then $H(p,m) = H(p,p) = -\sum_i p(x_i) \log_2(p(x_i))$, or ordinary entropy; this is 'as good as it gets'
- If we use non-optimal coding (model), $H(p,m) \geq H(p)$
- So we can use this measure to test how close a fit we have (random sentence generator gives pr for sentences)
- Example.

Example distribution p and 2 model approximations, m_1 and m_2

- p =true; m_1 = uniform dist; m_2 = observe A, D 4x more than B, C :

	A	B	C	D
p	0.4	0.1	0.25	0.25
m_1	0.25	0.25	0.25	0.25
m_2	0.4	0.1	0.1	0.4

$$H(p) = -\sum_i p(x_i) \log_2(p(x_i)) = 1.86$$

$$H(p, m_1) = -[0.4 \log_2(0.25) + 0.1 \log_2(0.25) + 0.25 \log_2(0.25) + 0.25 \log_2(0.25)] \\ = (0.4 + 0.1 + 0.25 + 0.25) (-2) = 2$$

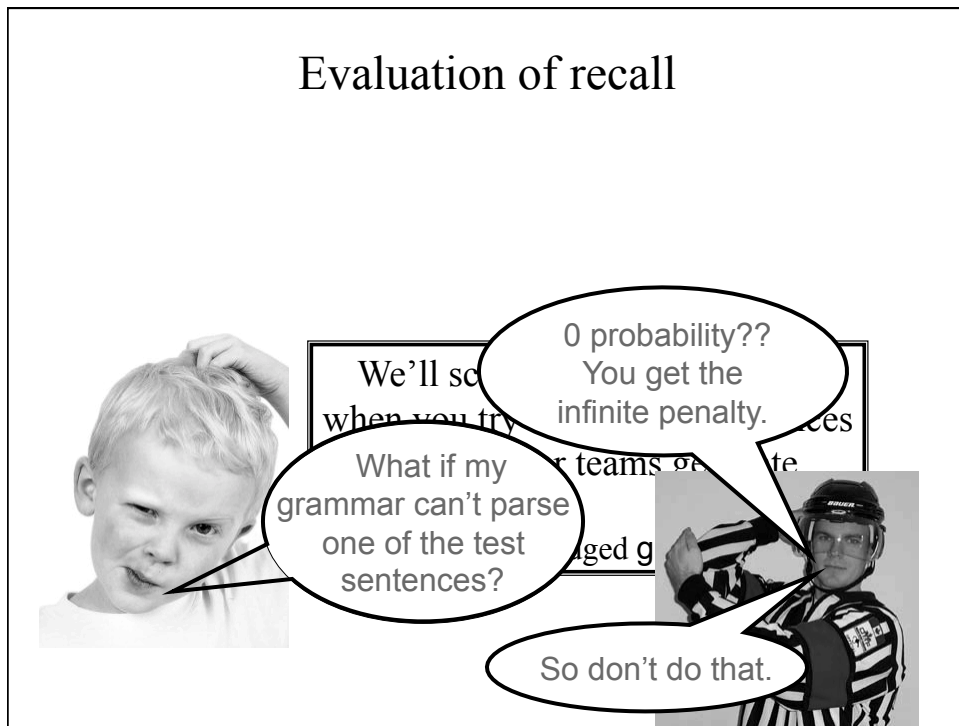
$$H(p, m_2) = -[0.4 \log_2(0.4) + 0.1 \log_2(0.1) + 0.25 \log_2(0.1) + 0.25 \log_2(0.4)] \\ = -[0.4 \times -1.32 + 0.1 \times -3.32 + 0.25 \times -3.32 + 0.25 \times -1.32] = 2.02$$

We will actually use this closely related formula, where $p(s_i)$ is the probability of your grammar generating test sentence s_i

$$\sum_{i=1}^{i=n} \log_2 p(s_i) / |s_i|$$

So if your grammar can't generate a test sentence *at all*, its probability is 0, $\log(0)$ is – gulp!; if it *does* generate a sentence w/ probability 1, $\log(1)=0$, this is the best you can do (this is why the backoff grammar is helpful...sometimes...)

Evaluation of recall



A walk-through of modifying the grammar (from the handout)

- who does Arthur suggest she carry ?
- New POS: (Penn treebank tags)

<i>who</i>	WP	a wh-pronoun
<i>does</i>	VBZ	a 3 rd person verb ('z' at end)
<i>Arthur</i>	NNP	Proper noun singular
<i>suggest</i>	VB	verb base form
<i>she</i>	PRP	personal pronoun
<i>carry</i>	VB	verb base form
<i>?</i>	PuncQ	punctuation question mark

Try doing this bottom up

- Work shown on board
- Start with ‘she carry’, then ‘suggest she carry’, ...
- What’s wrong with this answer...?

6. Discussion questions to keep in mind

- What did you do? How?
- Was CFG expressive enough?
 - How would you improve the formalism?
 - Would it work for other languages?
- How should one pick the weights?
 - And how could you build a better backoff grammar?
 - Is grammaticality well-defined? How is it related to probability?
- What if you had 36 person-months to do it right?
 - What other tools or data do you need?
 - What would the resulting grammar be good for?
 - What evaluation metrics are most important?

6. Discussion questions

- What strategies did you employ in constructing your grammars?
- Did you tackle the problem in a top-down or bottom up manner (or both)?
- What were some of the linguistic phenomena you handled, and how?
- What considerations and assumptions did you make in deciding what types of part-of-speech, or what types of phrases (grammar nonterminal names) to include in your final grammar?
- Was the grammar formalism expressive enough? What would you do to improve it? Would it work for other languages? What if you and your teammates don't speak this "other" language?
- If s is a sentence, and $p1(s)$ and $p2(s)$ denote the respective probabilities that S1 and S2 generate s , then what is the probability that the whole grammar generates s ?
- How could you improve the probabilities assigned by S2, without losing the guarantee that S2 can generate any string of words?
- What would happen if you did not have S2?
- Is there any other way to combine S1 and S2 besides the one we recommended?
- Suppose you strategically decided to make your grammar devote most of its probability to a really unusual sentence. Who would that hurt more – your team or the other teams?
- How did you manually tune your weights? Did you use the tools provided? Did you write your own tools? Did they help?
- How could you build a better back-off grammar?
- Your grammar scores well when it finds a parse of high probability. But S1 might produce several different parses of a sentence, dividing the probability among them.
- Is the scoring method fair?

Evaluation for the Grammy Awards

- Use "gold standard" test set not the development set! (why?)
- 3 metrics, equally weighted:
 1. Grammaticality: fraction of 20 sentences generated by your team judged grammatical by 3 other people, akin to *precision*
 2. S1 recall: fraction of test set parseable w/o recourse to backoff S2 grammar
 3. Cross-entropy (bits/word over test set) – *lower* is better (closer match to 'true' probability)