Set ~~up~~ CMS!

Lois; Venkat online; ~~highlighted~~ bookmarked.
J; M bookmarks.

you can book me

(Census of famous 1st sentences)

11/13/14
Lec 23.

Admin-wise:

	11/13 ← today
(11/18)	11/20
11/25	—
12/2	12/4
12/9	12/11

Q: do we want a "hackathon" next week (Tues, say), to organize any tasks that can be done jointly?
(probably not a hackathon per se, but people ~~no~~ might want to coordinate)
→ Jack already passed some code along, as did Longqi

Q: do we want in-class presentations? (5 groups 'standing')
Piazza'd, to get initial input.

5 | 75
15.

– (could be done in a single class period, and is good impetus to have stuff done; but, takes away from time

→ (I think we're possibly already past the last possible day when this working would be useful.

~~I think 11/18 – 11/20~~

30
5
150 = 75 x 2 !!

One ~~alt~~ possibility:

lec.	11/18 lecture ? ~~[illegible]~~	11/20 checkup.
	11/25 lecture	—
	12/2 checkup	12/4 ~~checkup~~ project presentations
	appts	final.

<is on web PS>

Another:

11/18 lecture ? ~~[illegible]~~	11/20 lecture checkup ← ~~is this pushing the next checkup too late~~??
11/25 ~~checkup~~ lecture	—
12/2 lecture	12/4 project presentations.

- entropy.
- smoothing
- perplexity

things you can use.
NE
what's the diff btwn Stanford TM toolkit? Mallet?
new tools for webscale ngrams.
pronunciation dict.
Stanford parser: syntactic relations. (CS 347)
both const & dependency
MPQA

Back to Louis & Nenkova 2013, "What Makes Writing Great".

Recall: I mentioned that this paper is really worth reading line-by-line for being inventive about finding new features for a new problem, and for being quite detailed.

Remember this problem.

→ should serve as a source of inspiration for you when you try to think of interesting features for your interesting problems.

- Today, I want to first quickly point at some tools / techniques of immediate potential use to you,
but ~~also~~ more I'm going to use the paper as a jumping-off point to @ least touch on some more advanced topics.

---

§3.2, "use of people.

(named-entity recognition) - active area of research -
- Am. airlines example from J&M - AMR Corp. unlikely to be in your dictionary
✓ [Stanford NER tool (+ others - see resources mentioned on course webpgs)]
- has a ML-model ~~w/~~ w/ 7 classes, Time, location ... money, percent, date.

you may want to be able to identify these.

§3.3 "beautiful language" → unusual word combinations
you should care b/c this may be relevant to your projects
(a take on "what makes two types of lang diff.
- Fightin' words" n-grams last time
- here, syntax is taken into account ... altho' perhaps really as a way to limit interesting ngrams.

<"adjacent" nouns are still ngrams>
> show exples on same pg.

- note reference to 2 diff. types of parses.

although the missing article makes the 2nd interpretation seem less likely.

(also, can't we allowed to say/interpret agrammatical things?)

CFG-style: (using Jurafsky & Martin textbook slides ☺, ch 14)

(hide ~~irrelevant~~ non-used slides, to make things easier to find)

- slide 7 tree structure ["Book [the dinner flight]" vs. "Book [the dinner] flight$], explain that if "book [me] [a flight]" no ok, sty. "a flight" & "flight" interchangeable, then the 2nd is a possible parse.

(adjacent nouns in noun phrase, as Lois & Nevkova are interested in)

left:

- constituents as labeled subtrees:

- slide 5: CFG rules (can we blow up the "outline" pane so these can be put side-by-side)

- known parsing algs for this.

- relations btwn constituents.

(skip category-splitting à la Klein & Manning, for time)

-- but lexical info is important

At this point, we diverge to talking about formalisms,

skipped

- slide 20: "into" acts differently than "of" wrt attachment points "sacks" vs. "caught"

- slide 22: track lexical items + heads of phrases.

- ~~introduce complexity into the parsing/learning~~

Dependency-style: direct relations btwn lexical items

- fig 12.14 of J&M.

∃ linear-time deterministic parsers - MALT parser, for example.

"dependency"

- entropy: characterizing a distribution

use θ_i notation: (extend to items, not necessarily single words).

~~use gnuplot~~ $\sum_i \theta_i \log\left(\frac{1}{\theta_i}\right)$

"surprisal".

gnuplot: show $0\log(0) \doteq 0$ by limiting arguments.

A reconstruction of what we talked about in class, w/ accessory notes.

- We have intuitions that there are ~~substructures~~ hidden and/or relations to ~~language~~, sentences (sorta like how we inferred the presence of hidden substructures in discourse).

• subjects; predicates, modifiers; modifiees, etc.

If we assume some sort of grammar describing the possible structures, then "parsing" is the process of recovering the possible structures of a sentence w.r.t. that grammar.

CFG-~~style~~s : constituents are diff types (noun phrase NP, verb phrase VP, adjectives ADJ, etc) of word subsequences; [lexical items are a disjoint finite set of words. Com constituent is distinguished start symbol S].

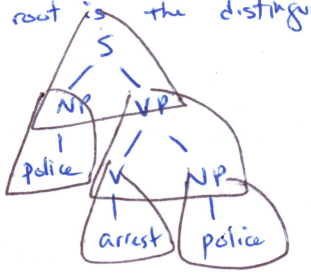
~~basic~~ "rules" ~~rules~~ consist of a finite set of depth-1 decompositions of these types (think of them as depth-1 branches:

: ~~sent~~ a sentence can be made up of an NP then a VP

: a verb can be made up of a verb and then an NP (its direct object).

(pos)

Valid trees have each branch being ~~part of the fun~~ one of the finite set of "branch" rules, plus all the leaves are lexical items, and root is the distinguished start-symbol.

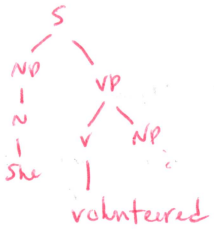


Sounds quite reasonable, but here's one problem in practice:

• (Except in grad school), "volunteer" is an intransitive verb. (no direct object)
So we expect trees like this

But since our lang also has transitive verbs, we must have a rule for allowing a direct object:

... which means we're going to have to have trees like



which is bad, b/c "volunteer" should not take a direct object.

A technical fix is to say that there isn't just one category "VP", but to say there are two types, "VP-trans" and "VP-intrans", where we wouldn't allow

and one can do this, altho' category proliferation is going to be a bit of a beast that one will have to manage. And there are other issues,

<see posted notes from previous CS6740>

- ~~also, we didn't talk about~~
- ~~<lexicalization should~~
  ~~importance of~~

<one approach is feature-based context-free grammars>.

But sticking w/ CFGs has the advantage that there are parsing algorithms that recover all possible structures for a sentence of length n in $O(n^3)$.

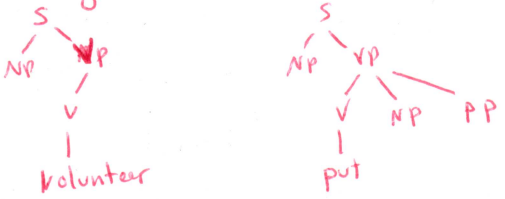
(implicitly)

~~You~~

Alternatively, we can move beyond CFGs, to formalisms where we take a hit in parsing time, but get some ~~modeling~~ linguistic modeling elegance in return.

conceptual step (1): what about letting our rules be not single ~~branch~~ depth-1 branches, but larger subtrees? (*)

Tree substitution grammars "rules":



note lexicalization: (optional, but potentially desirable)
these trees are now the items in your dictionary.

You can, as a grammar engineer, directly encode the argument structure your words expect.
→ linguistically nice!

"combination" is substitution: of a tree w/ root labeled by some category X into a tree's leaf when that leaf has the same label X.

So, clearly CFG's are a special case.

(*) ~~in class, I wrongly stated~~ (*) in class, I wrongly stated that tree insertion grammars are the same. They are not, but allow certain types of adjunction.

TSG's are weakly equivalent to CFGs: they generate the same sub of strings.
But who cares about generation? We're generally in a setting where we've been given some text; we have to analyze it.
It's the analyses, we care about.

TSG's are not strongly equivalent to CFGs: they generate diff. analysis sets.

ex: a TSG w/ only the "rule" S → S / a generates only one tree structure.

a strongly-equivalent CFG would have to have the rules S → S / a : S → a

But then it must also allow the analysis

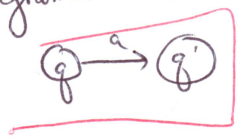
, a 2nd tree. Contradicting strong equivalence, since TL should only allow a single tree,

(one might also note that for TSGs, the derived tree – the analysis – doesn't necessarily match the ~~derivation~~. derivation tree: the description of which rules were used to make the derived tree.

This would seem to complicate parsing. ~~check~~ (but since TIG is parseable in time cubic in string length, TSGs must be as well.

Aside: by the way, if checking whether a string is "legal" or not as fast as possible is your goal, you might ~~to~~ want to go to sthg less complex than CFGs.

Finite-state grammars can be encoded as right-branching CFGs:

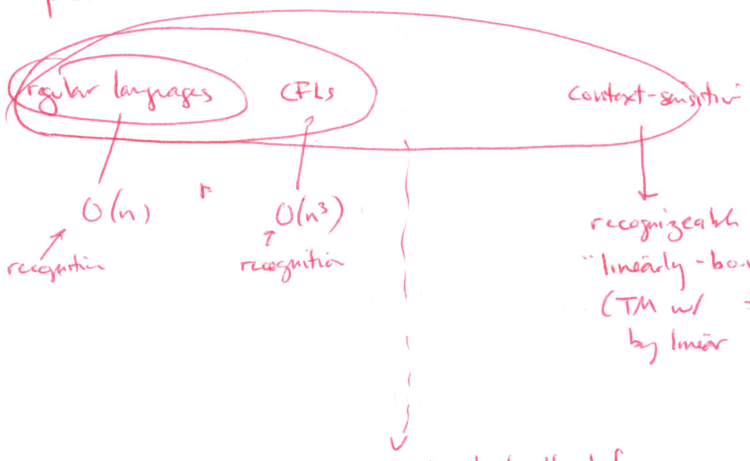


can be encoded as a branch

-ngram lang. models have this "flavor", and you have linear-time recognition.

Should you be ~~If you're instead~~ willing to move up the Chomsky hierarchy of language classes?; ~~what~~
(in terms of weak generative capacity)

- ~~parsing time~~ will increase:
traditional picture



~~turns out that 4 diff~~
turns out that 4 diff formalisms yielded the same class — which makes you more confident that the class is "real" — and it's been dubbed the "mildly context-sensitive languages".

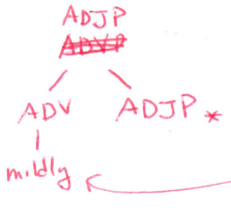
~~for tree-~~
for tree-adjoining grammars, $O(n^6)$.

But, some nice ~~linguistic or~~ linguistically-oriented properties, and some arguments that ~~not CFLs~~ natural lang is indeed not CF.

- citation to Pullum's account of the history, including the "tie" btwn Shieber (Swiss-German) and Culy (African language Bambara) showing crossing dependencies, a non-CF phenomena.
({ww}, in formal-lang speak)

TAG's: ~~not~~ addition of new tree type; new operation (adjunction)

root has same type label as foot, distinguished by *.

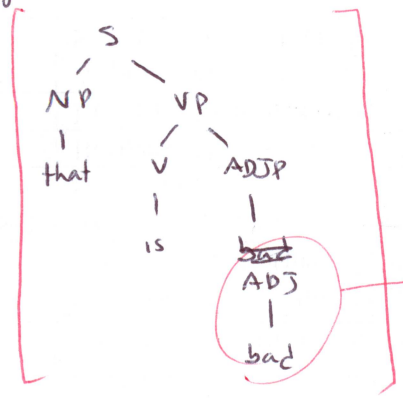
ex:


note: this is a modifier

ex:


(so, need to distinguish foot since could have multiple leaves w/ same label)

idea: spse you have some "base" structure already:



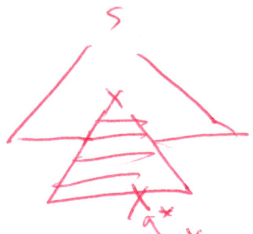
→ and then you'd like to modify:

adjunction:

base structure

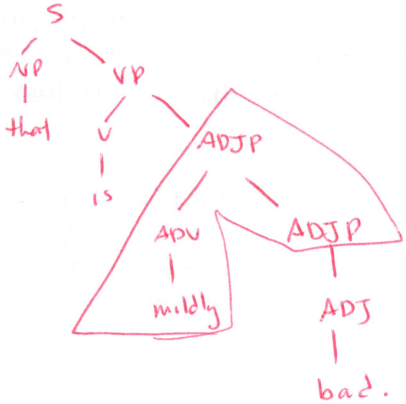
+

~~split~~ remove  and replace w/ adjonction tree:

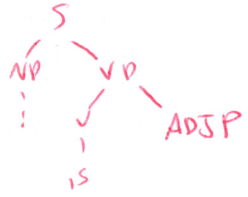


now, when does the leftover go? That's what the foot is for!

--->



contrast what you'd have to do w/ a CFG or a TSG: b/c you need a leaf node avail to add "mildly" into, you have to decide from this stage:



whether to do the modification, which is non-intuitive that the decision to insert "mildly" is done before you "decide" what "mildly" is going to modify ("bad")!