FNLP Exam Notes

Made by Leon :)

1 smooth and stuff

Definition 1.0.1: Maximum Likelihood Estimates (MLE)

$$P_{RF}(x) = \frac{C(x)}{N}$$

$C(x)$ is the count of x in the dataset, and N is the total number of items in the dataset

- **Problem 1 (Sparse data problem)**: If the count of an item is 0, then the probability will also be 0 - you want the model to be able to calculate sentences with new words in them. **Solution**: Smoothing
- **Problem 2**: Cannot reliably find probability of sentences (the chance of "skibidi sigma gyatt rizz" being already in a corpus is very low). **Solution**: use n-gram models

Definition 1.0.2: n-gram models

Turn a sentence $P(S = w_1 \dots w_n)$ into joint probabilities $P(w_1, \dots, w_n)$. We have $P(X,Y) = P(Y|X)P(X)$. So

$$\begin{aligned} P(a,b,c) &= P(c|a,b)P(a,b) \\ &= P(c|a,b)P(b|a)P(a) \end{aligned}$$

n-gram model just estimates probability to n probabilities

- **Trigram**: $P(w_i|w_1, w_2, \dots, w_{i-1}) \approx P(w_i|w_{i-2}, w_{i-1})$
- **Bigram**: $P(w_i|w_1, w_2, \dots, w_{i-1}) \approx P(w_i|w_{i-1})$
- **Unigram**: $P(w_i|w_1, w_2, \dots, w_{i-1}) \approx P(w_i)$

To be able to detect edges of sentences, add `<s>` and `<\s>` on sentence edges to be factored into the n-gram model

$$\texttt{skibidi rizz} \implies \texttt{<s> skibidi rizz <\textbackslash s>}$$

therefore a bigram like $P(\texttt{<\textbackslash s>}|\texttt{rizz})$ will detect the end of a sentence
Usually, **negative log probs** will be used instead of regular decimals, as the probabilities will get small fast and floating precision issues will happen.

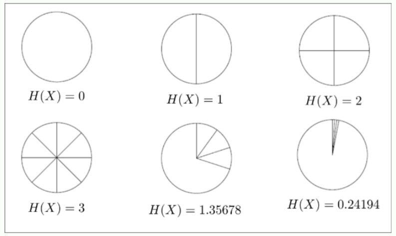
- Probabilities from 0 to 1, but negative log probs go from 0 to ∞
- Log probs are added instead of multiplied like regular probabilities

Definition 1.0.3: Entropy

Entropy of a random variable X:

$$H(X) = \sum_x -P(x)\log_2 P(x)$$

also the expected value of $-\log_2 P(X)$
Higher entropy means less predictable



For $w_1 \dots w_n$ with large n, per-word cross-entropy is well approximated by

$$H_M(w_1 \dots w_n) = -\frac{1}{n}\log_2 P_M(w_1 \dots w_n)$$

Lower cross-entropy $\implies$ model is better at predicting next word
Perplexity: $2^{\text{cross-entropy}}$

Definition 1.0.4: Add-one and Lidstone smoothing

Add one smoothing

$$P_{+1}(w_i|w_{i-2}, w_{i-1}) = \frac{C(w_{i-2}, w_{i-1}, w_i) + 1}{C(w_{i-2}, w_{i-1}) + v}$$

where v is the vocabulary size

Add-α smoothing

$$P_{+\alpha}(w_i|w_{i-1}) = \frac{C(w_{i-1}, w+i) + \alpha}{C(w_{i-1}) + \alpha v}$$

Choosing an α: Use a three-way data split: **training set** (80-90%), **held-out/development set** (5-10%), and **test set** (5-10%)

- Train model (estimate probabilities) on training set with different values of α
- Choose the α that minimizes cross-entropy on development set
- Report final results on test set

More generally, use development set for evaluating different models, debugging, and optimizing choises. This avoids overfitting to the training set and even to the test set

Definition 1.0.5: Good-Turing Smoothing

$$c* = (c+1)\frac{N_{c+1}}{N+c} \qquad P*_c = \frac{c*}{N} = (c+1)\frac{\frac{N_{c+1}}{N_c}}{N}$$

- N_c is the number of occurances with count c
- $P*_c$ is the probability of an item with count c
- $c*$ is the good-turing smoothed version of count
- N is total count

random items

- Probability the next observation is new

$$P(unseen) = \frac{N_1}{N}$$

- Probability the next observation is a specific new object

$$P_{GT} = \frac{1}{N_0}\frac{N_1}{N} \implies c* = \frac{N_1}{N_0}$$

Problems with Good-Turing

- Assumes we know the vocabulary size
- Doesn't allow "holes" in the counts (if $N_i > 0, N_{i-1} > 0$)
- Applies discounts even to high-frequency items
- Assigns equal probability to all unseen events, same with add-α, e.g. "w rizz" vs "w indowpane" shouldn't be equal

Definition 1.0.6: Interpolation and backoff

Interpolation: Combines higher and lower order N-gram models, since they have different strengths and weaknesses

- high-order N-grams are sensitive to more context, but have sparse counts
- low-order N-grams have limited context but robust counts

If P_N is N-gram estimate

$$P_{\text{INT}}(w_3|w_1, w_2) = \lambda_1 P_1(w_3) + \lambda_2 P_2(w_3|w_2) + \lambda_3 P_3(w_3|w_1, w_2)$$

Katz-backoff: Trust the highest order language model that contains the N-gram. Requires an adjusted prediction model: $P*(w_i|w_{i-N+1}, \dots, w_{i-1})$ and backoff weights: $\alpha(w_1, \dots, w_{N-1})$
Kneser-Ney: Takes diversities of histories into account

- count of distinct histories for a word

$$N_{1+}(\circ w_i) = |\{w_{i-1} : c(w_{i-1}, w_i) > 0\}|$$

- In KN smoothing, replace raw counts with count of histories:

$$P_{\text{ML}}(w_i) = \frac{C(w_i)}{\sum_w C(w)} \implies P_{\text{KN}}(w_i) = \frac{N_{1+(\circ w_i)}}{\sum_w N_{1+}(\circ w)}$$

Method use cases:

- Uniform probabilities - add-α, Good-Turing
- Probabilities from lower-order n-grams - interpolation, backoff
- Probability of appearing in new contexts - Kneser-Ney

2 Text Classification

Categorizing the *content* of the text. e.g.

- Spam detection (binary classification: spam/not spam)
- Sentiment analysis (binary / multiway)
  - movie, restaurant, product reviews (pos/neg, or 1-5 stars)
  - political argument (pro/con or pro/con/neutral)
- Topic classification (multiway: sport/finance/travel/etc)

Or, categorizing the *author* of the text (authorship attribution)

- Native language identification (e.g. to tailor language tutoring)
- Diagnosis of disease (psychiatric or cognitive impairments)
- Identification of gender/dialect/educational background (e.g. forensics [legal matters], advertising/marketing)

n-gram models are not as useful for classification - often we can just consider a **bag of words** and not worry about the order that the words come in

2.1 Naive Bayes

Definition 2.1.1: Naive Bayes

Given document d and set of categories C we want to assign d to the most probable category $\hat{c}$

$$\hat{c} = \arg\max_{c \in C} P(c|d) = \arg\max_{c \in C} P(d|c)P(c)$$

Represent d as the set of features (words) it contains: $f_1, f_2, \ldots, f_n$

$$P(d|c) = P(f_1, f_2, \ldots, f_n|c)$$

Then make **naive Bayes assumption** that features are conditionally independent given the class

$$P(f_1, f_2, \ldots, f_n|c) \approx P(f_1|c)P(f_2|c)\ldots P(f_n|c)$$

i.e. the probability of a word happening depends **only** on the class, not on words occuring before/after (n-gram), or even what other words occurred at all. Basically we only care about the **count** of each feature in a document

---

Naive Bayes classifier: Given a document with features $f_1, f_2, \ldots, f_n$ and set of categories C, choose the class $\hat{c}$ where

$$\hat{c} = \arg\max_{c \in C} P(c) \prod_{i=1}^{n} P(f_i|c)$$

- $P(c)$ is the **prior probability** of class c before observing any data. normally estimated with MLE:
$$\hat{P}(c) = \frac{N_c}{N}$$
  - N_c is the number of training documents in class c
  - N is the total number of training documents.

  Therefore, $\hat{P}(c)$ is the proportion of training documents in class c
- $P(f_i|c)$ is the probability of seeing feature f_i in class c. Normall estimated with simple smoothing:
$$\hat{P}(f_i|c) = \frac{\text{count}(f_i, c) + \alpha}{\sum_{f \in F}(\text{count}(f, c) + \alpha)}$$
  - $\text{count}(f_i, c)$: the number of times f_i occurs in class c
  - F: the set of possible features
  - α: the smoothing parameter, optimized on held-out data

2.1.2 Negative Log Probabilities

Same with n-gram models, usually uses **negative log probabilities** - adjusted equation:

$$\hat{c} = \arg\min_{c \in C} +(-\log P(x) + \sum_{i=1}^{n} -\log P(f_i|c))$$

This amounts to classification using a linear function (in log space) of the input features. Therefore Naive bayes is called a **linear classifier**

2.1.3 Issues with choosing features

- Sentiment analysis might need domain-specific non-sentiment words e.g. "quiet" or "memory" for computer reviews
- Stopwords might be useful features for other tasks, e.g. People with schizophrenia use more 2nd-person pronouns, and people with depression use more 1st-person
- Probably better to use too many irrelevant feaetures than not enough relevant ones

2.1.4 Problems with annotation

Usually hard to come by already annotated text - ergo you need someone to label text. On the other hand there is usually a lot of unannotated texts.
Solution: Use semi-supervised learning

1. Train NB on labeled data alone
2. Predict labels on unlabelled data
3. Re-estimate NB, but now using also self-labelled data

2.1.5 Self Training

- **Advantages**: Simplicity and applicable to any classifier
- **Disadvantages**: Does not account for uncertainty of a classifier, and no theoretical motivation
- To make it work needs discarding low-confidence predictions, and curriculum (start with examples similar to labeled data)

2.1.6 Expectation Maximisation for Semi-supervised Learning

- Train NB on labelled data alone
- Make soft prediction on unlabelled data ("E-step")
- Recompute NB parameters using the soft counts

Self-training for NB is known as "hard EM"

2.1.7 Advantages of Naive Bayes

- Very easy to implement
- Very fast to train and classify new docs (good for huge datasets)
- Doesn't require as much training data as some other methods (good for small datasets)
- Usually works reasonably well
- Should be the baseline method for any classification task

2.1.8 Evolving past naive Bayes

- Assuming that all features are conditionally independent can have some issues, and often we have enough training data for a better model.
- Adding multiple feature types (e.g. words and morphemes) often leads to even stronger correlations between features
- Accuracy of classifier can sometimes still be ok, but it will be highly overconfident in its decision, e.g. NB sees 5 features that all point to class 1, treats them as 5 independent sources of evidence - like asking 5 friends for an opinion when some got theirs from each other

Definition 2.1.9: Maximum Entropy / Logistic Regression

Most commonly **multinomial logistic regression**. **multinomial** if more than two possible classes, otherwise just **logistic regression**
Like Naive Bayes, assign a document x to class $\hat{c}$ where

$$\hat{c} = \arg\max_{c \in C} P(c|x)$$

unlike Naive Bayes, model $P(c|x)$ directly instead of using Baye's rule

---

Discrimination

- Trained to discriminate correct vs wrong values of c given input x
- Need not be probabilistic
- Examples: artificial neural networks, decision trees, nearest neighbour methods, support vector machines
- Here we only consider one method: MaxEnt models which are probabilistic

---

Feature Functions: Like Naive Bayes, MaxEnt models use **features** we think will be useful for classification.
However, features are treated different in the two models

- NB: Features are **directly observed** (e.g. words in doc): no difference between features and data
- MaxEnt: We will use $\vec{x}$ to represent the observed data. Features are **functions** that depend on both observations $\vec{x}$ and class c

---

Classification with MaxEnt
Choose the class that has highest probability according to

$$P(c|\vec{x}) = \frac{1}{Z} \exp\left(\sum_i w_i f_i(\vec{x}, c)\right)$$

- Normalization constant $Z = \sum_{c'} \exp(\sum_i w_i f_i(\vec{x}, c))$
- Inside brackets is just a dot product $\vec{w} \cdot \vec{f}$
- $P(c|\vec{x})$ is a **monotonic function** of this dot product
- So, we will end up choosing the class for which $\vec{w} \cdot \vec{f}$ is highest

---

Training the model Given annotated data, choose weights that make the labels most probable under the model That is, given items $x^{(1)} \ldots x^{(N)}$ with labels $c^{(1)} \ldots c^{(N)}$, choose

$$\hat{w} = \arg\max_{\vec{w}} \sum_j \log P(c^{(j)}|x^{(j)})$$

This is called **conditional maximum likelihood estimation** (CMLE)
Like MLE, CMLE will overfit, so use **regularization** to avoid that

2.1.10 Relation to Naive Bayes

Naive Bayes is also a linear classifier, and can be expressed in the same form. Should the features actually be independent they would converge to the same solution as the amount of training data increases

2.1.11 Downside to MaxEnt models

- Supervised MLE in generative models is easy - compute counts and normalize
- Supervised CMLE in MaxEnt is not so easy
  - requires multiple iterations over the data to gradually improve weights (using gradient ascent)
  - Each iteration computese $P(c^{(j)}|x^{(j)})$ for all j, and each possible $c^{(j)}$

- This can be time-consuming, especially if there are a large number of classes and/or thousands of features to extract from each training example

2.1.12 Robustness: MaxEnt and Naive Bayes

- Imagine that in training there is one very frequent predictive feature, e.g. in training sentiment data contained emoticons but not at test time
- The model can quickly learn to rely on this feature
  - model is confident on examples with emoticons
  - the gradient on these examples gets close to zero
  - the model does not learn other features
- In MAxEnt, a feature weight will depend on the precense of other predictive features
- Naive Bayes will rely on all features - the weight of a feature is not affected by how predictive other features are
- This makes NB more robust than (basic) MaxEnt when test data is (distributionally) different from training data

3 POS Tagging and HMMs

3.1 Intro

3.1.1 Why do we care about POS tagging

First step towards syntactic analysis, which is useful for semantic analysis Simpler models and often faster than full parsing, but sometimes enough to be useful. e.g. POS tags can be useful features in text classification or word sense disambiguation

3.1.2 Word types

- Open class words (or content words)
  - nouns, verbs, adjectives, adverbs
  - refers to objects, actions and features. Open class since there is no limit to the words
- Closed class words (or function words)
  - pronouns, determiners, prepositions, connectives
  - limited number, they mainly tie concepts of a sentence together

3.1.3 Why is POS tagging hard?

- Ambiguity - e.g. "glass of water/NOUN" vs "water/VERB the plants"
- Sparse data - words we haven't seen before, or word-tag pairs we haven't seen before

We want a model that decides tags based on:

- The word itself
  - Some words may only be nouns
  - Some words are ambiguous
  - Probabilities may help, if one tag is more likely than the other
- Tags of surrounding words
  - two determiners rarely follow each other
  - two base form verbs rarely follow each other
  - determiner is almost always followed by adj or noun

3.2 Hidden Markov Models

viterbi lol

3.2.1 HMM adjacent

Using Viterbi, we can find the best tags for a sentence (decoding) and get $P(y,x|0)$
We might also want to:

- Compute the likelihood i.e. the probability of a sentence regardless of its tags (a language model!) $P(x|\theta)$
- Learn the best set of parameters $\hat{\theta}$ given only an unannotated corpus of sentences

We know that $P(x|\theta) = \sum_y P(x,y|\theta)$, but there are an exponential number of sequences y. By computing and storing partial results we can solve efficiently. This is the forward algorithm

Definition 3.2.2: Forward Algorithm

$$\text{Initialise:} v_j^1 = a_{\text{START},\ jb_j, x^1} \qquad v_j^1 = a_{\text{START},\ jb_j, x^1}$$

$$\text{Recomput:} v_j^t = \left(\max_i v_i^{t-1} a_{ij}\right) b_{j,x^t} \qquad v_j^t = \left(\sum\nolimits_i v_i^{t-1} a_{ij}\right) b_{j,x^t}$$

$$\text{Final:} v_{\text{STOP}}^{|x|+1} = \max_i v_i^{|x|} a_{i,\text{STOP}} \qquad v_{\text{STOP}}^{|x|+1} = \sum\nolimits_i v_i^{|x|} a_{i,\text{STOP}}$$

4 Syntax and Parsing

4.1 Intro

4.1.1 Intro intro

Various ways to model word behaviour:

- Bag-of-words: ignore word order entirely
- N-gram model: capture a fixed length history to predict word sequences
- HMM: Also capture fixed-length history using latent variables

Useful for various tasks but a really accurate model of language needs more than a fixed-length history

4.1.2 Constituents

Substitutability at the phrasal level -

- POS categories indicate which words are substitutable e.g. sub adjectives

  I saw a red cat $\iff$ I saw a former cat

- Phrasal categories indicate which phrases are substitutable. e.g. sub NP

  Dogs sleep soundly $\iff$ My next-door neighbours sleep soundly

Constituent tests
"The lecture was absolutely brilliant"

- **Substitution**: replace constituent with base

  it was absolutely brilliant

- **Clefing**: replace with "it was ____ that ____"

  it was the lecture that was absolutely brilliant

- **Coordination**: Add word of same type with "and/or/but"

  the lecture and the cheat sheet was absolutely brilliant

- **Wh-movement**: Add a "what/when/who/.." question at the start

  what was absolutely brilliant? the lecture

4.2 Theories of Syntax

A **theory of syntax** should explain which sentences are **well-formed** (grammatical) and which are not. Two theories: **Constituency structures**, and **Dependency structures**

Definition 4.2.1: Context Free Grammar

A CFG is a tuple of 4 elements $G = (V, \Sigma, R, S)$

- V - the set of non-terminals
- Σ - the set of terminals
- R - the set of rules of the form $X \to Y_1, y_2, \ldots, Y_n$ where $n \geq 0, X \in V, Y_i \in V \cup \Sigma$
- S is a dedicated start symbol

A CFG defines both a set of strings (a language), and structures used to represent sentences (constituent trees)

4.2.2 Ambiguity

Some sentences have more than one parse: **structural ambiguity**.

- The structure ambiguity can be caused by **PoS ambiguity** (both are types of syntactic ambiguity)
- Some sentences have structural ambiguity even without PoS ambiguity. This is called **attachment ambiguity**
  - Depends on where different phrases attach in the tree
  - Different attachments have different meanings

Key problems

- **Recognition problem**: does the sentence belong to the language defined by the CFG
- **Parsing problem**: what is a (most plausible) derivation (tree) corresponding the sentence? (parsing encompasses recognition)

4.2.3 Chomsky Normal form (chomp chomp)

Converting to CNF form

- Get rid of empty (i.e. ϵ) productions: $C \to \epsilon$
- Get rid of unary rules: $C \to C_1$
- N-ary rules: $C \to C_1 C_2 \ldots C_n \quad (n > 2)$

e.g. $NP \to DT \;\; NNP \;\; VBG \;\; NN$ becomes

- $NP \to DT \;\; @NP|DT$
- $@NP|DT \to NNP \;\; @NP|DT\_NNP$
- $@NP|DT\_NNP \to VBG \;\; NN$

Definition 4.2.4: Probabilistic CFG

CFG where all the states have probabilities associated with them
Not all PCFGs give rise to a proper distribution over trees i.e. the sum over probabilities of all trees the grammar can generate may be less than 1 : $\sum_T P(T) < 1$
Good news: any PCFG estimated with the maximum likelihood procedure are always proper

4.3 Parsing

4.3.1 Speeding up algorithm (approximate search)

Basic Pruning (roughly)

- for each span (i, j) store only labels which have the probability at most N times smaller than the probability of the most probably label for this span
- Check not all rules but only rules yielding subtree labels having non-negligible probability

Course to fine pruning: parse with a smaller (simpler) grammar and pre-compute (posterior) probabilities for each spans, and use only the ones with non-negligible probability from the previous grammar

4.3.2 Parser Evaluation

Intrinsic evaluation

- Automatic: evaluate against annotation provided by human experts (gold standard) according to some predefined measure
- Manual: ... according to human judgement

Extrinsic Evaluation: score syntactic representation by comparing how well a system using this representation performs on some task. e.g. use syntactic representation as input ofr a semantic analyzer and compare results of the analyzer using syntax predicted by different parsers

4.3.3 Automatic Intrinsic Evaluation

the usual method, parsers are evaluated against gold standard provided by linguists

- Training set: used for estimation of model parameters
- Dev set: used for tuning the model (initial experiments)
- Test set: final experiments to compare against previous work

something else

- Exact match: percentage of trees predicted correctly
- Bracket score: scores how well individual phrases (and their boundaries) are identified (this is the usual method)
- Crossing brackets: percentage of phrases boundaries crossing

Definition 4.3.4: Bracket score

Regards a tree as a collection of brackets: $[\min, \max, C]$ The set of brackets predicted by a parser is compared against the set of brackets in the tree annotated by alinguist Precision, recall and F1 are used as scores

- $\mathrm{Pr} = \dfrac{\text{number of brackets the parser and annotation agree on}}{\text{number of brackets predicted by the parser}}$
- $\mathrm{Re} = \dfrac{\text{number of brackets the parser and annotation agree on}}{\text{number of brackets in annotation}}$
- $\mathrm{F1} = \dfrac{2 \times \mathrm{Pr} \times \mathrm{Re}}{\mathrm{Pr} + \mathrm{Re}}$

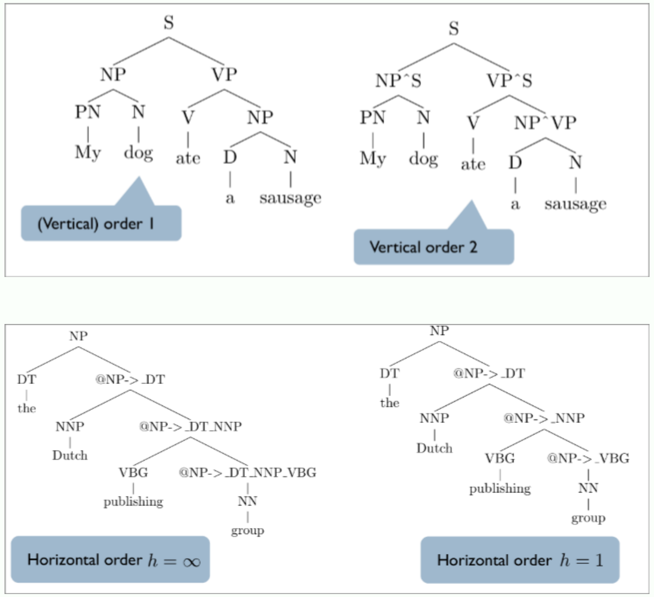
4.3.5 Treebank PCFG

just reading off the treebank Weaknesses:

- They do not encode lexical preferences
- They do not encode structural properties (beyond single rules)

e.g. subject and object NPs are very different

Definition 4.3.6: Vertical and Horizontal Markovization



4.4 Lexicalisation

4.4.1 random lexical things

Lexicalization: create new categories by adding the lexical head to patterns

---

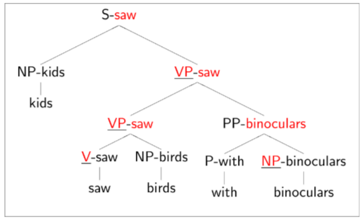
Projectivitiy: a sentence's dependency parse is projective if every subtree occupies a contiguous span of the sentence - doesn't cross any edges in the flat graph

4.4.2 Pros and Cons

- Adding category-splitting makes the grammar more specific (good)
- Leads to huge grammar blowup and very sparse data (bad)
- Ways to balance these issues
  - Complex smoothing schemes (similar to n-gram interpolation/backoff)
  - More recently, increasing emphasis on automatically learned subcategories

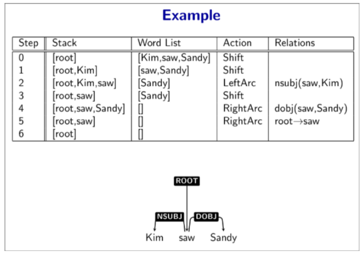
4.4.3 Head rules

Assign each rule a "head" to inherit from, and propage up the graph



4.4.4 Shift reduce

it's sure a thing



Example

Step	Stack	Word List	Action	Relations
0	[root]	[Kim,saw,Sandy]	Shift	
1	[root,Kim]	[saw,Sandy]	Shift	
2	[root,Kim,saw]	[Sandy]	LeftArc	nsubj(saw,Kim)
3	[root,saw]	[Sandy]	Shift	
4	[root,saw,Sandy]	[]	RightArc	dobj(saw,Sandy)
5	[root,saw]	[]	RightArc	root→saw
6	[root]	[]		

4.4.5 Critera for parser

- Target representation: constituency or dependency?
- Efficiency? in practice, both runtime and memory use
- Incrementability: parse the whole sentence at one, or obtain partial ltr analyses/expectations
- Retrainable system?

4.4.6 Graph-based vs Transition-based vs Conversion-based

- TB: Features in scoring function can look at any part of the stack; no optimality guarantees for search; linear time; (classically) projective only
- GB: Features in scoring function limited by factorization; optimal search within that model; quadratic time; no projectivity constraint
- CB: In terms of accuracy, sometimes best to first constituency-parse, then convert to dependencies. Slower than direct methods, some treebanks are available solely in dependency form

4.4.7 summary

while constituency parses give heierarchically nested phrases, dependency parses represent syntax with trees whose edges connect words in the sentence (no abstract phrases like NP). Edges often labelled with relations like subj. Head rules govern how a lexicalised constituency grammar can be extracted from a treebank, and how a constituency parse can be converted to a dependency parse. Two main paradigms, graph-based and transition-based, with diff kinds of models and search algos

5 First order logic and semantic stuff

we can lambda calc our ass off to resolve some syntactic ambiguities. This doesn't resolve *all* semantic ambiguity though - word sense, semantic scope, anaphoric expressions

5.1 First order Logic

Example 5.1.1: First order logic examples

- Fred ate rice: $\exists e(\text{eat}(e, \text{fred}, \text{rice}) \land e \prec n)$
- Fred ate rice at midnight:

$$\exists e(\text{eat}(e, \text{fred}, \text{rice}) \land e \prec n \land \exists x(\text{with}(e, x) \\ \land \text{fork}(x)) \land \text{at}(e, \text{midnight}))$$

- Every dog had a bone

$$\forall x(\text{dog}(x) \to \exists y(\text{bone}(y) \land \text{have}(x, y)))$$
$$\exists y(\text{bone}(y) \land \forall x(\text{dog}(x) \to \text{have}(x, y)))$$

Ambiguity in semantics: dog bone or "every man loves a woman"

- $\forall x(\text{man}(x) \to \exists y(\text{woman}(y) \land \exists e(\text{love}(e, x, y) \land n \subseteq e)))$
  "for every man, there exists a woman where he loves her"
  $\implies$ Every man can love a different woman
- $\exists y(\text{woman}(y) \land \forall x(\text{man}(x) \to \exists e(\text{love}(e, x, y) \land n \subseteq e)))$
  "There exists a woman where every man loves her"
  $\implies$ Same woman loved by all men

5.1.2 Scope

The ambiguity arises because **every** and *a* each has its own scope:

- Interpretation 1: **every** has scope over *a*
- Interpretation 2: **a** has scope over **every**

Scope is not uniquely determined either by left-to-right order, or position in the parse tree. We therefore need more mechanisms to make sure it's reflected in the LF assigned to S. **Solution**: semantic underspecification (whatever that is)

Definition 5.1.3: Problems, Pros, Cons

Most general question: why is NLP Hard? tl;dr ambiguity

- Homophones: **blue** vs **blew**
- Word senses: **bank** (finance or river?)
- Part of Speech: **chair** (noun or verb?)
- Syntactic structure: **I saw a girl with a telescope**
- Quantifier scope: **Every child loves some movie**
- Multiple: **I saw her duck**
- Reference: John dropped the goblet onto the glass table and it broke
- Discourse: The meeting is cancelled. Nicholaas isn't coming to the office today

Sentiment Analysis - simple counting of a lexicon

- Hard to know whether words that seem pos/neg are actually used that way - sarcasm/irony, or maybe mentioning opposing viewpoints
- Opinion words may be describing the subject e.g. a characters actions/attitude
- Some words act as semantic modifiers of other opinion-bearing words/phrases

Human Annotation

- Gray areas for annotation schemes, and we want consistency
- Takes time and resources to get actual people
- Annotation quality problems
  - Simple error
  - Not reading full context
  - Not noticing an erroneous pre-annotation
  - Forgetting a detail from the guidelines
  - Cases not anticipated by/specified in guidelines

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetuer at, consectetuer sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

Morbi luctus, wisi viverra faucibus pretium, nibh est placerat odio, nec commodo wisi enim eget quam. Quisque libero justo, consectetuer a, feugiat vitae, porttitor eu, libero. Suspendisse sed mauris vitae elit sollicitudin malesuada. Maecenas ultricies eros sit amet ante. Ut venenatis velit. Maecenas sed mi eget dui varius euismod. Phasellus aliquet volutpat odio. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Pellentesque sit amet pede ac sem eleifend consectetuer. Nullam elementum, urna vel imperdiet sodales, elit ipsum pharetra ligula, ac pretium ante justo a nulla. Curabitur tristique arcu eu metus. Vestibulum lectus. Proin mauris. Proin eu nunc eu urna hendrerit faucibus. Aliquam auctor, pede consequat laoreet varius, eros tellus scelerisque quam, pellentesque hendrerit ipsum dolor sed augue. Nulla nec lacus.

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetuer odio sem sed wisi.

Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetuer eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

Etiam euismod. Fusce facilisis lacinia dui. Suspendisse potenti. In mi erat, cursus id, nonummy sed, ullamcorper eget, sapien. Praesent pretium, magna in eleifend egestas, pede pede pretium lorem, quis consectetuer tortor sapien facilisis magna. Mauris quis magna varius nulla scelerisque imperdiet. Aliquam non quam. Aliquam porttitor quam a lacus. Praesent vel arcu ut tortor cursus volutpat. In vitae pede quis diam bibendum placerat. Fusce elementum convallis neque. Sed dolor orci, scelerisque ac, dapibus nec, ultricies ut, mi. Duis nec dui quis leo sagittis commodo.