— IN2110 —
Methods in Language Technology

Summing up
Exam preparations

Pierre Lison & Erik Velldal & Lilja Øvrelid

Language Technology Group (LTG)

May 25, 2022

Topics for today

- Practical details regarding the final exam
- High-level summary
- Sample exam questions (though not a sample exam)
- Oblig winners!

Exam preparation

- Both the lecture notes (slides) and the background reading specified in the lecture schedule (at the course page) are obligatory reading.
- We also expect that you have looked at the provided model solutions for the exercises.
- 2020 exam: much larger in scope (1 week home exam) but useful preparation
- 2021 exam: slightly larger in scope (6 hours home exam), but also useful
- Make use of group sessions this week!

Exam

When / where:

- 1 June at 15:00 (4 hours)
- Silurveien 2, Sal 4C and 4D (check on StudentWeb)
- Digital written exam on Inspera

Exam

When / where:

- 1 June at 15:00 (4 hours)
- Silurveien 2, Sal 4C and 4D (check on StudentWeb)
- Digital written exam on Inspera

- Each section will have points attached (summing to 100) to give you an idea of how they will be weighted in the grading.
- No support material allowed during the exam.
- A in-build calculator will be available in Inspera.

After IN2110

- Please remember to participate in the course evaluation hosted by FUI.
  - Even if this means just repeating the comments you already gave for the midterm evaluation.
  - While the midterm evaluation was only read by us, the FUI course evaluation is distributed department-wide.

After IN2110

- Please remember to participate in the course evaluation hosted by FUI.
  - Even if this means just repeating the comments you already gave for the midterm evaluation.
  - While the midterm evaluation was only read by us, the FUI course evaluation is distributed department-wide.
- Some other courses of potential interest:
- IN3120/IN4120 – Search technology
  - Fall 2022
  - Also based on the book by Manning, Raghavan, & Schütze (2008), *Introduction to Information Retrieval*
- IN3050/IN4050 – Introduction to AI and machine learning
  - Spring 2023

Inventory

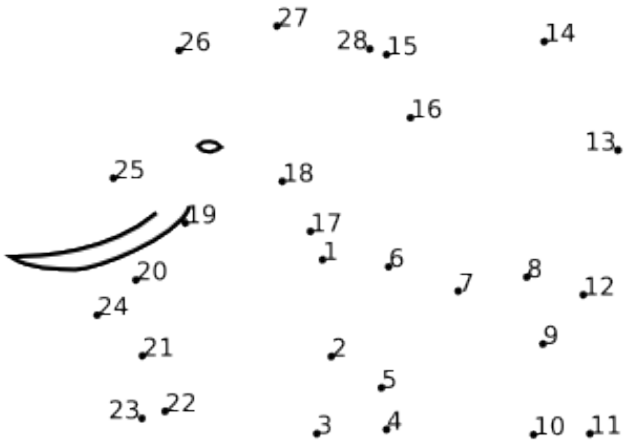
Main areas

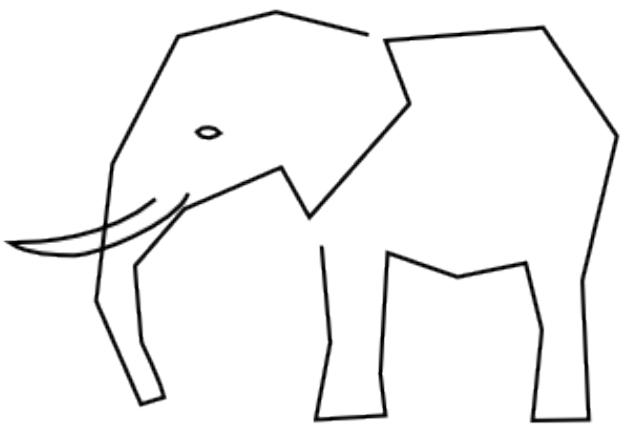
- Vector space models
- Classification
- Clustering
- Sequence modeling
- Dependency syntax and parsing
- MT
- Dialog
- Evaluation methodology

Progression

- Representation
- From geometric to probabilistic models
- From 'point-wise' to sequential and hierarchical modeling

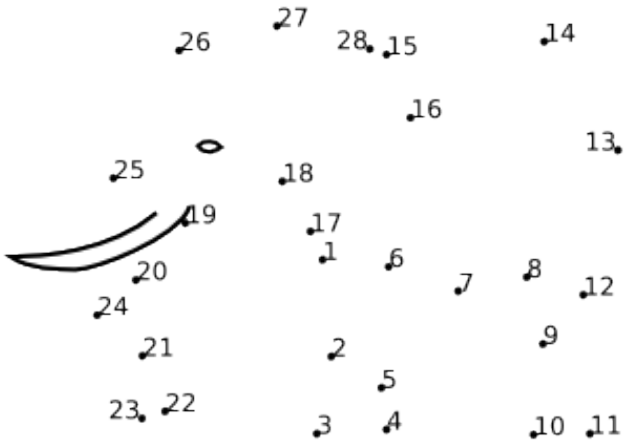
Connecting the dots...

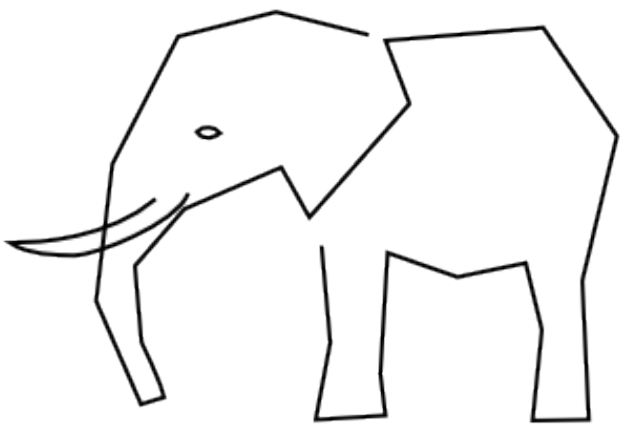




What have we been doing?

Connecting the dots...

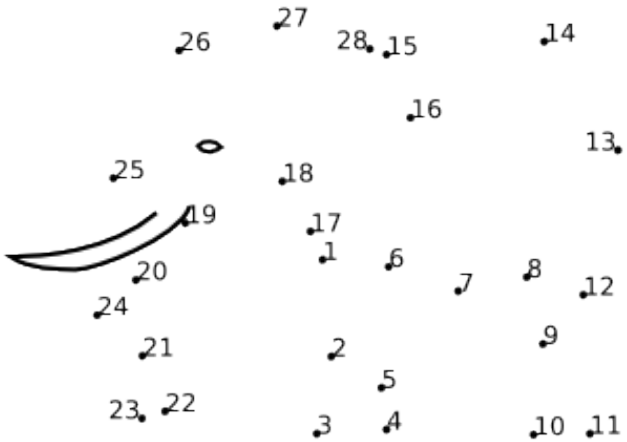


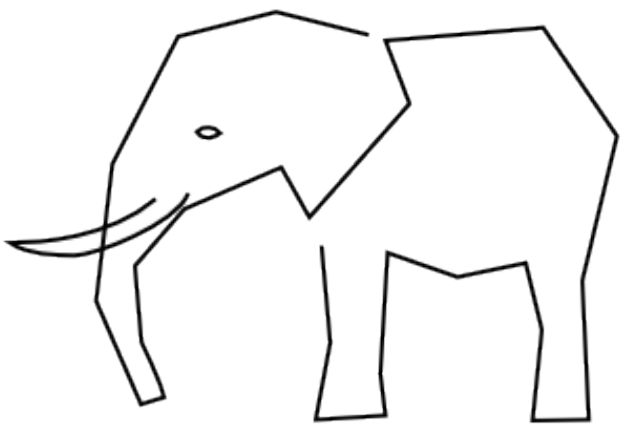


What have we been doing?

- Data-driven learning

Connecting the dots...

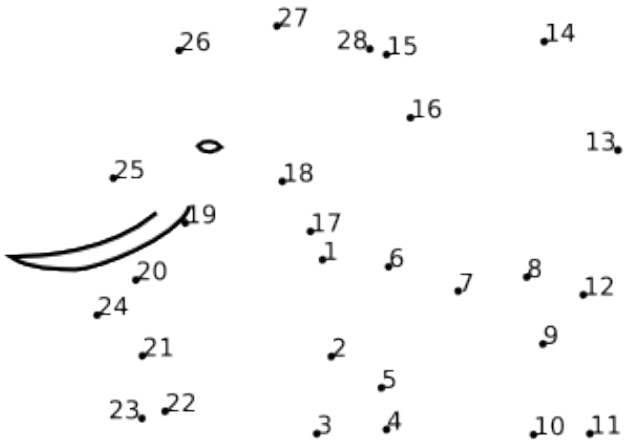


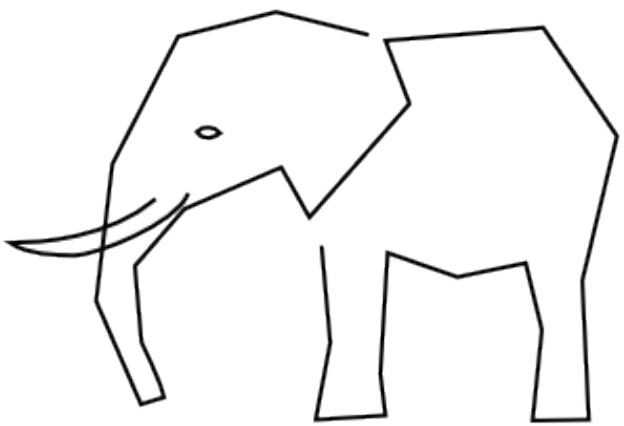


What have we been doing?

- Data-driven learning
- by counting observations

Connecting the dots...

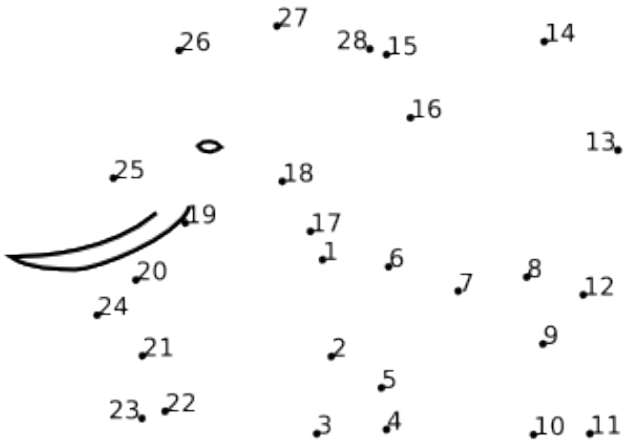


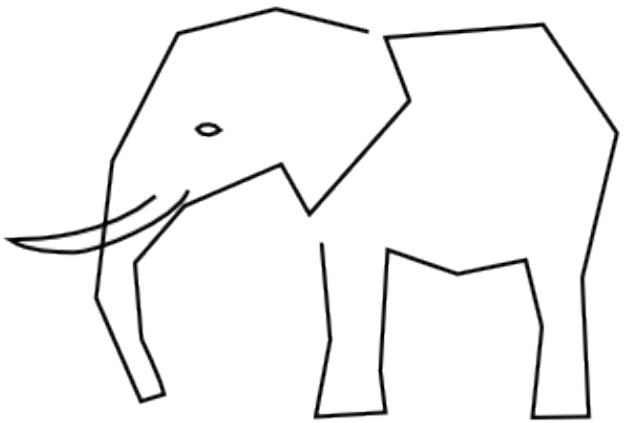


What have we been doing?

- Data-driven learning
- by counting observations
- in context;

Connecting the dots...



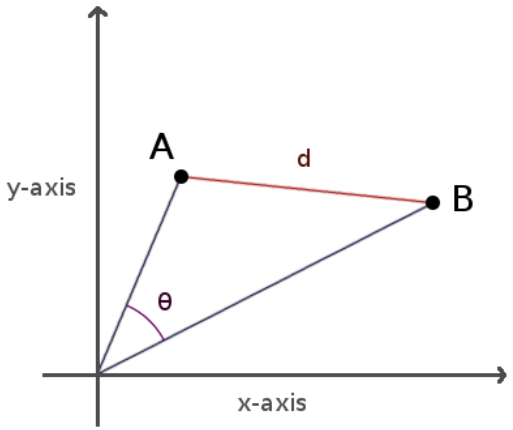


What have we been doing?

- Data-driven learning
- by counting observations
- in context;
  - context words in vector space models; bag-of-words, etc.
  - previous words in n-gram models
  - previous states in HMMs
  - features of configurations in dependency parsing
  - source/target words for MT
  - etc

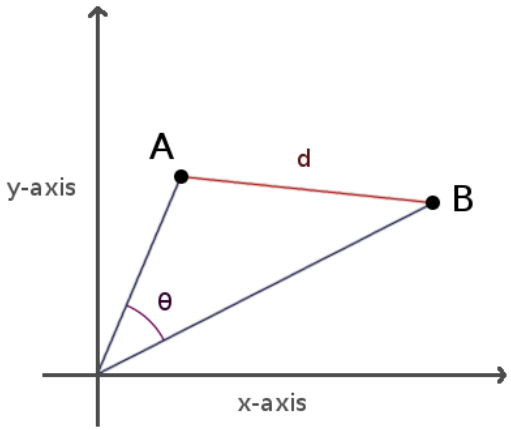
Vector space models

- General framework for representing data in terms of feature vectors.
- Can measure distance between objects in the geometrical model.
- Example: representing documents and words



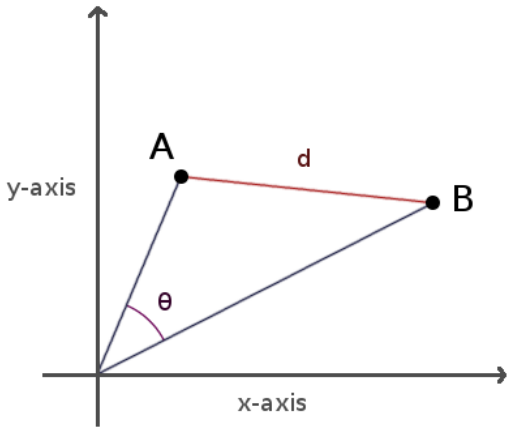
Vector space models

- General framework for representing data in terms of feature vectors.
- Can measure distance between objects in the geometrical model.
- Example: representing documents and words
- Distributional semantics



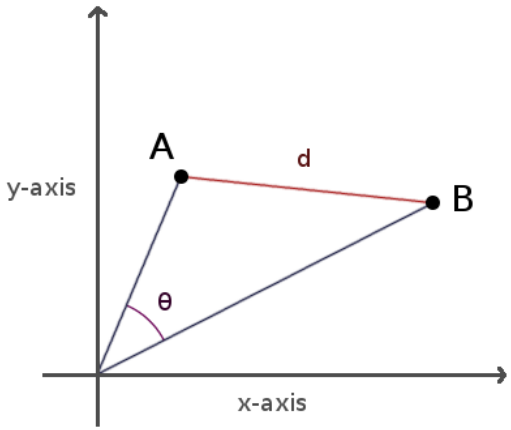
Vector space models

- General framework for representing data in terms of feature vectors.
- Can measure distance between objects in the geometrical model.
- Example: representing documents and words
- Distributional semantics
- Related issues: How to define words and contexts, various levels of text pre-processing, weighting, evaluating semantic vectors, challenges with distributional representations, and more.



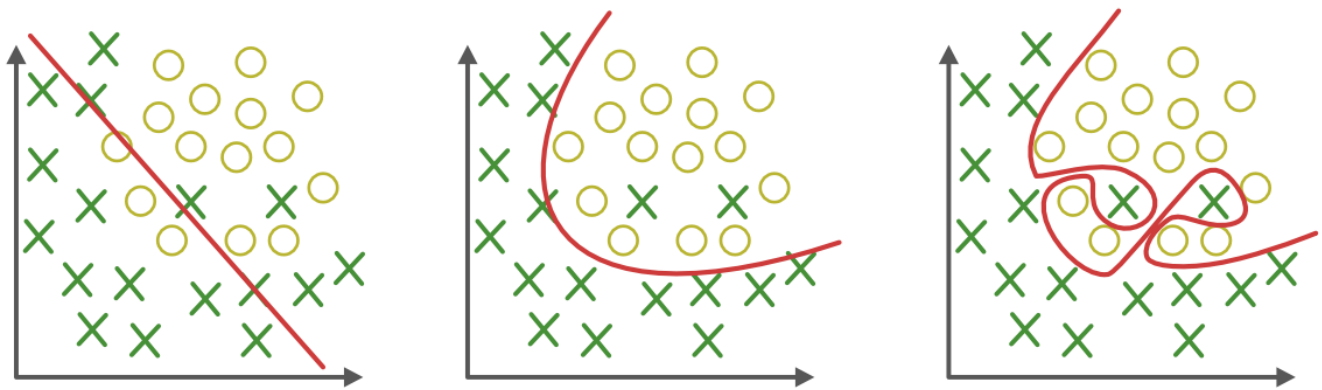
Vector space models

- General framework for representing data in terms of feature vectors.
- Can measure distance between objects in the geometrical model.
- Example: representing documents and words
- Distributional semantics
- Related issues: How to define words and contexts, various levels of text pre-processing, weighting, evaluating semantic vectors, challenges with distributional representations, and more.
- Embeddings



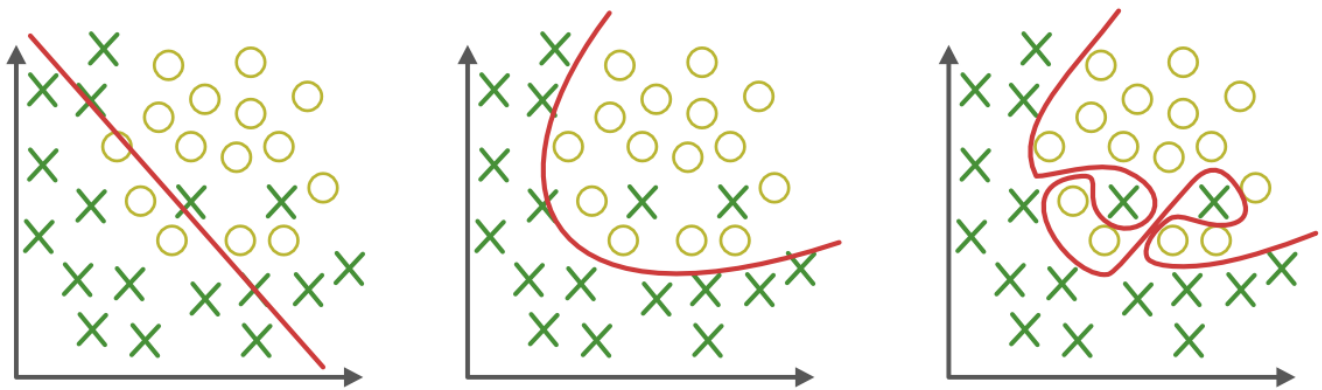
Classification and clustering

- Supervised vs unsupervised learning
- Labeled vs unlabeled data



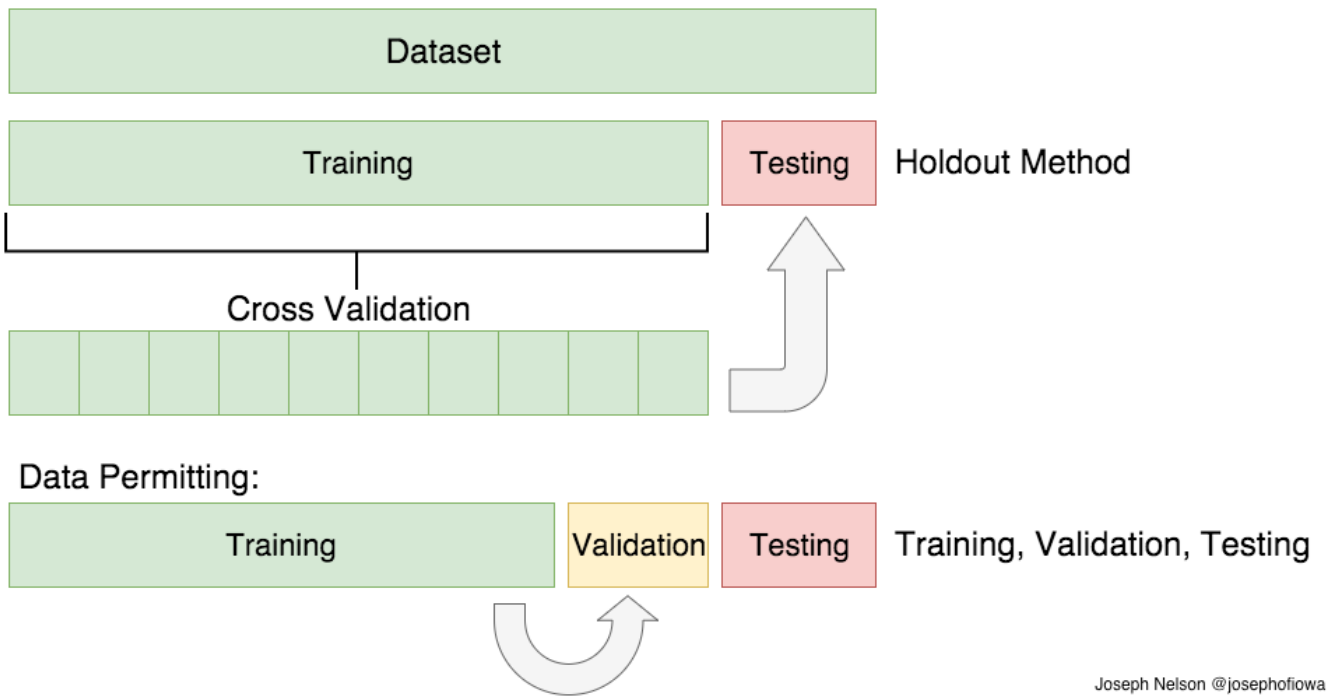
Classification and clustering

- Supervised vs unsupervised learning
- Labeled vs unlabeled data
- Classification: kNN, Rocchio, logistic regression
- Clustering: k-means



Evaluating classifiers

- How to test models and the use of data splits
- Counting correct and incorrect predictions: true positives, false negatives, etc.
- Metrics; accuracy, precision, recall, f-score, and ways of averaging.



Vektorrom; eksempler på spørsmål

- Hva ligger i begrepet distribusjonell semantikk?

Vektorrom; eksempler på spørsmål

- Hva ligger i begrepet distribusjonell semantikk?
- Når vi skal representere dokumenter som trekkvektorer basert på bag-of-words er det vanlig å ikke bare bruke rå frekvens-tellinger som trekkverdier, men å vekte disse med en funksjon som f.eks. TF-IDF. Hvilke trekk (ord) vil generelt bli vektet mest opp og mest ned ved bruk av TF- IDF? Forklar med kun et par setninger.

Vektorrom; eksempler på spørsmål

- Hva ligger i begrepet distribusjonell semantikk?
- Når vi skal representere dokumenter som trekkvektorer basert på bag-of-words er det vanlig å ikke bare bruke rå frekvens-tellinger som trekkverdier, men å vekte disse med en funksjon som f.eks. TF-IDF. Hvilke trekk (ord) vil generelt bli vektet mest opp og mest ned ved bruk av TF- IDF? Forklar med kun et par setninger.
- Når vi jobber med vektorrom-representasjoner av dokumenter benytter vi oss ofte av lengde-normalisering. Forklar hva dette innebærer og hvilken praktisk nytte det kan ha.

Vektorrom; eksempler på spørsmål

- Hva ligger i begrepet distribusjonell semantikk?
- Når vi skal representere dokumenter som trekkvektorer basert på bag-of-words er det vanlig å ikke bare bruke rå frekvens-tellinger som trekkverdier, men å vekte disse med en funksjon som f.eks. TF-IDF. Hvilke trekk (ord) vil generelt bli vektet mest opp og mest ned ved bruk av TF- IDF? Forklar med kun et par setninger.
- Når vi jobber med vektorrom-representasjoner av dokumenter benytter vi oss ofte av lengde-normalisering. Forklar hva dette innebærer og hvilken praktisk nytte det kan ha.
- Hva mener vi med henholdsvis stemming og lemmatisering? Hvilken effekt vil det ha for trekk-representasjonene våre (f.eks BoW) dersom vi gjør slik pre-prosessering av teksten, sammenliknet med å bruke ords fullformer?

Classification and clustering: sample questions

- Explain the difference between supervised and unsupervised learning. For both approaches, mention examples of models that we've touched on throughout the course.

Classification and clustering: sample questions

- Explain the difference between supervised and unsupervised learning. For both approaches, mention examples of models that we've touched on throughout the course.
- Imagine that we are working on a binary classification problem where there are many more examples in the negative than the positive class (assume a ratio of 9:10). Discuss whether or not Accuracy is a suiteable evaluation measure for this problem.

Classification and clustering: sample questions

- Explain the difference between supervised and unsupervised learning. For both approaches, mention examples of models that we've touched on throughout the course.
- Imagine that we are working on a binary classification problem where there are many more examples in the negative than the positive class (assume a ratio of 9:10). Discuss whether or not Accuracy is a suiteable evaluation measure for this problem.
- What are the differences and similarities between Rocchio and kNN?

Classification and clustering: sample questions

- Explain the difference between supervised and unsupervised learning. For both approaches, mention examples of models that we've touched on throughout the course.
- Imagine that we are working on a binary classification problem where there are many more examples in the negative than the positive class (assume a ratio of 9:10). Discuss whether or not Accuracy is a suiteable evaluation measure for this problem.
- What are the differences and similarities between Rocchio and kNN?
- In the context of model evaluation, briefly describe what 'micro-averaging' and 'macro-averaging' means, including their differences.

Logistisk regresjon

En lineær modell for klassifikasjon:

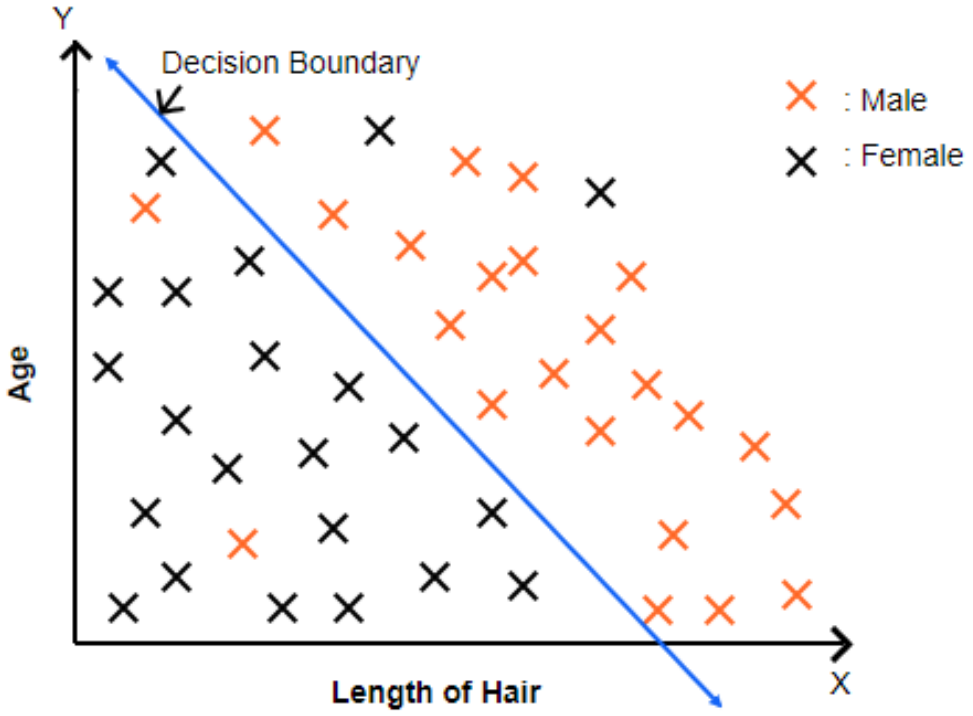
$$y = \sigma\left(\mathbf{w} \cdot \mathbf{x} + b\right) \tag{1}$$

- **x** er en featurevektor som representer datapunktet å klassifisere
- y er modellprediksjonen (dvs. sansynnligheten for at punktet **x** tilhører den positive klassen i følge modellen).
- **w** (vektene) og b (skjæringspunktet) er parametrene som estimeres ut fra treningsdata, for eksempel via stochastic gradient descent.
- Sigmoid-funksjonen $\sigma(z) = \frac{1}{1+e^{-z}}$ begrenser resultatet fra den vektede summen $\mathbf{w} \cdot \mathbf{x} + b$ til å være en sansynnlighet, dvs. en tall i $[0, 1]$.
- For > 2 klasser bruker vi *softmax*-funksjonen i stedet for sigmoid

Logistisk regresjon

En lineær modell for klassifikasjon:

$$y = \sigma\left(\mathbf{w} \cdot \mathbf{x} + b\right) \tag{2}$$



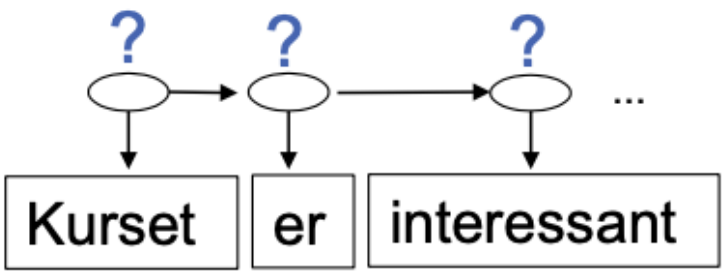
- Logistisk regresjon deler datapunktene med en (rett) linje.
- Linjen defineres gjennom parametrene $\mathbf{w}$ og b.
- Estimering av $\mathbf{w}$ og b (=trening av modellen) sikter på å *minimere tapet* på treningspunktene.

Logistisk regresjon: øvelse

- Se på koden i: `https://github.uio.no/IN2110/v21/tree/master/gruppetimer/ekstra/logistisk_regresjon.py`
- Koden er en liten utvidelse av språkidentifikatoren fra oblig1b hvor vi bruker forekomst av *delsekvenser av IPA symboler* (med lengder fra 1 til N) som features i stedet for isolerte IPA symboler.
- Hvis vi øker N fra 1 til 2 (altså med bigrams over IPA symboler) ser vi at *accuracy* blir bedre.
- Men deretter skjer det noe merkelig: *accuracy* blir faktisk **verre** med $N > 2$!
- Hva er det som skjer? Forklar hva er som er problemet.
- Hva kan vi endre i modellen slik at vi unngår dette problemet?

Sekvensmodeller

- Mange oppgaver i NLP kan beskrives som *sekvenslabellering*.
- Sekvensene kan bestå av ord, bokstaver, orddeler, symboler, akustiske observasjoner, osv.
- Vi har sett nærmere på en viktig modell for sekvenslabellering: **Hidden Markov Models** (HMMs).



HMMs består av:

1. Et sett med mulige merkelapper
2. Et vokabular over observasjoner
3. En emisjonsmodell $P(o_t|s_t)$
4. En transisjonsmodell $P(s_t|s_{t-1})$
5. Og en sansynnlighetsfordeling $P(s_0)$ over starttilstander.[a]

[a] Kan lemmes i transisjonsmodellen ved å anta en "dummy" starttilstand.

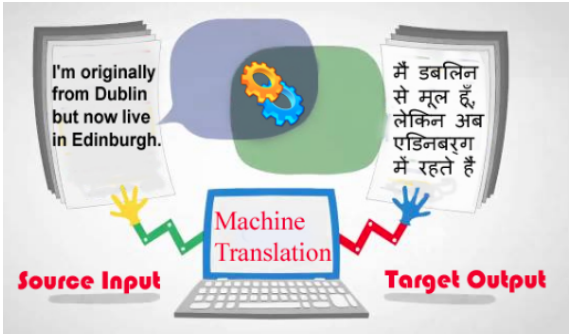
Sekvensmodeller : øvelser

- La oss si at vi ønsker å utvikle en modell for Named Entity Recognition med 4 labeller: PERS, ORG, LOC og MISC. Vi antar en vokabular på 10 tusen ord.
- Hva er størrelsen (*shape*) på transisjonsmatrisen?
- Hva er størrelsen på emisjonsmatrisen?

- Hvor mange sansynnligheter må beregnes når vi kjører Viterbi på en setning som består av 10 ord?

Maskinoversettelse (MT)

- MT handler om å oversette tekst eller tale fra et *kildespråk* til et *målspråk*.
- Mange språklige utfordringer: fleretydighet, faste utrykk, morfologi, ordstilling, *m.fl.*



- Maskinoversettelsesystemer kan utvikles ved å skrive mange oversettelseregler, eller via *datadrevne metoder*
- Datadrevne metoder baserer seg på parallelkorpora, og kan bestå av enten statistiske modeller eller (som vanligst i dag) nevrale *sequence-to-sequence* modeller.
- Evaluering av oversettelsekvaliteet: menneskelige vurderinger (*fluency*, *adequacy*) eller automatiserte metoder (f.eks. BLEU).

Maskinoversettelse: øvelser

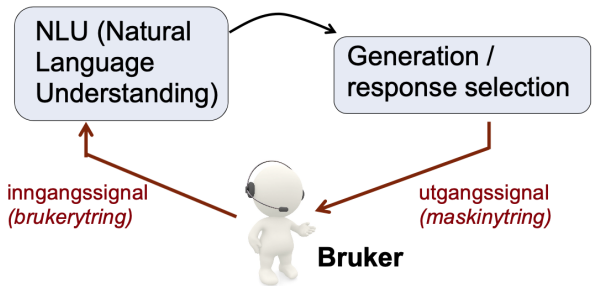
- Hvorfor er det vanskelig å trene en *ordbasert* (i motsetning til f.eks. *bokstavebasert*) maskinoversettelsemodell når kilde- eller målspråket har en veldig rik morfologi?
- Forklar hvorfor BLEU ikke er en perfekt måling av oversettelsekvalitet. Finn noen konkrete eksempler for å støtte dine forklaringer.
- MT-portalen *Apertium*[1] tilbyr oversettelser mellom bokmål og nynork basert på en "shallow transfer" regelbasert modell[2]. Hva er fordelene og ulempene ved å bruke en regelbasert tilnærming når man skal oversette mellom bokmål og nynorsk?

[1] `https://www.apertium.org/index.nob.html?dir=nob-nno`

[2] For mer detaljer: `http://rua.ua.es/dspace/handle/10045/12025`

Interaktive systemer

- I vår siste forelesning snakket vi om *interaktive systemer*, slik som prateroboter eller dialogsystemer.



- Spontane samtaler mellom mennesker er *samarbeidsaktiviteter*:
  - Vi signaliserer til hverandre når vi ønsker å gi eller ta ordert (turtaking)
  - Vi uttrykker *dialogakter*, blant annet ulike *grounding*-signaler for å forsikre seg at vi forstår hverandre riktig.
  - Vi tolker hverandres ytringer på en samarbeidsvillig måte (*conversational implicatures*)
- Interaktive systemer kan være enten regelbaserte eller datadrevne
- Dialogstyring er nødvendig for å handtere mer kompliserte interaksjoner

Interaktive systemer: øvelser

- Tegn en endelig tilstandsautomat for en praterobot som tar imot pizzabestillinger.

En kort, oppfunnet samtale:

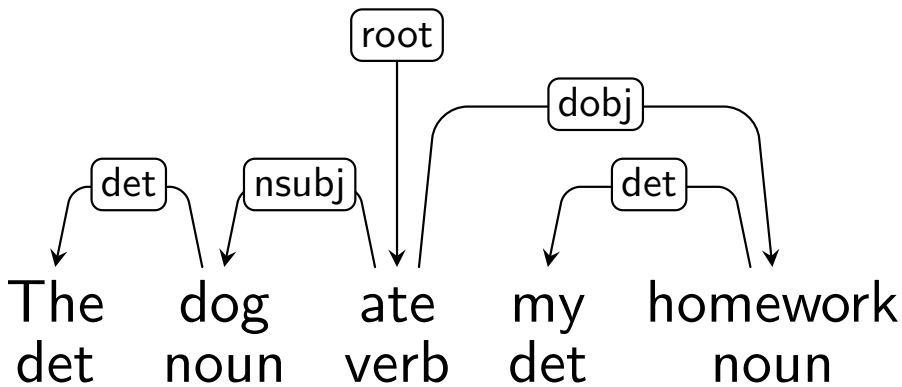
A: Hi there! So, how was your day?
B: Well, my boss put one more big pile of work on my desk today, as if I didn't have anything else to do...
A: Poor you... Oh, while you're up, could you help me carry these groceries in the kitchen?
B: Sure will do. Oh, did you see Mark's email today?
A: Which email?
B: Mark's. He sent it this afternoon I think
A: Mm no I didn't check my email this afternoon
B: Oh I see
A: So what was he saying?
B: Uh?
A: What was he saying?
B: Well he was asking whether we had something planned on the 28th
A: The 28th? But we already hav/
B: Yes I know, we'll be in Trondheim that day. **A**: Too bad, we haven't seen him for quite some time!
B: Yeah... you know what, why don't you tell him that we're busy on the 28th, but that we have nothing planned for the following weekend?

A: Good idea!

- Hvilke dialogakter (i følge taksonomien fra Searle) kan du finne i samtalen?
- Hva slags *grounding*-signaler kan du finne i samtalen?
- Er det også noen *conversational implicatures*?

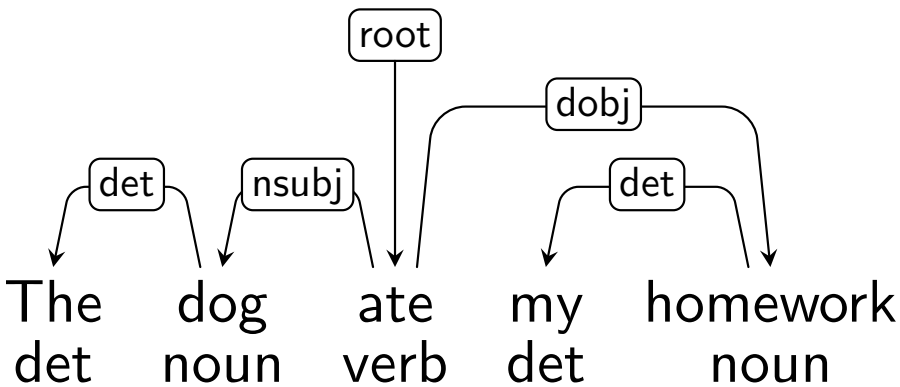
Dependency Grammar (DG)

- An alternative to phrase structure representations
- Syntactic functions are central
- Claimed to be closer to semantic analysis
- The basic idea:
  - Syntactic structure consists of lexical items, linked by binary asymmetric relations called dependencies.
  - Represented as dependency graphs.



Dependency Grammar (DG)

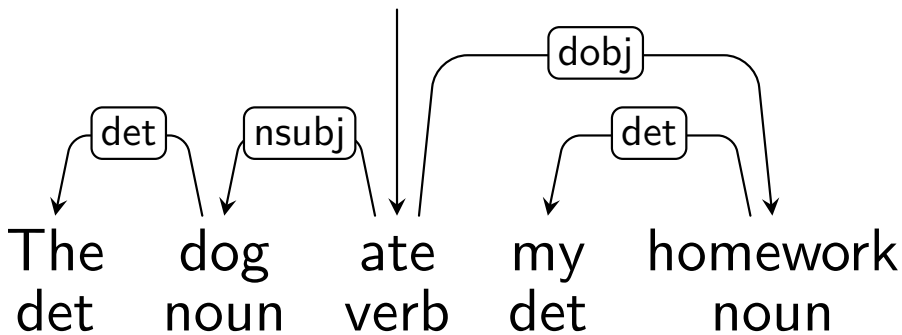
- An alternative to phrase structure representations
- Syntactic functions are central
- Claimed to be closer to semantic analysis
- The basic idea:
  - Syntactic structure consists of lexical items, linked by binary asymmetric relations called dependencies.
  - Represented as dependency graphs.



- Criteria for heads and dependents

Formal Conditions on Dependency Graphs

- Principles:
  - Syntactic structure is complete (Connectedness).
  - Syntactic structure is hierarchical (Acyclicity).
  - Every word has at most one syntactic head (Single-Head).
- Connectedness is enforced by adding a special root node (node 0).



- Projectivity
- Treebanks: Universal Dependencies

Transition-based dependency parsing

- Basic idea:
  - define a transition system for mapping a sentence to its dependency graph
  - Learning: induce a model for predicting the next state transition, given the transition history
  - Parsing: Construct the optimal transition sequence, given the induced model

Transition-based dependency parsing

- Basic idea:
  - define a transition system for mapping a sentence to its dependency graph
  - Learning: induce a model for predicting the next state transition, given the transition history
  - Parsing: Construct the optimal transition sequence, given the induced model
- Transition system (arc eager):

SHIFT	move from front of buffer to top of stack
REDUCE	pop the top of stack (requires existing head)
LEFT-ARC(K)	leftward dependency of type k; reduce
RIGHT-ARC(K)	rightward dependency of type k; shift

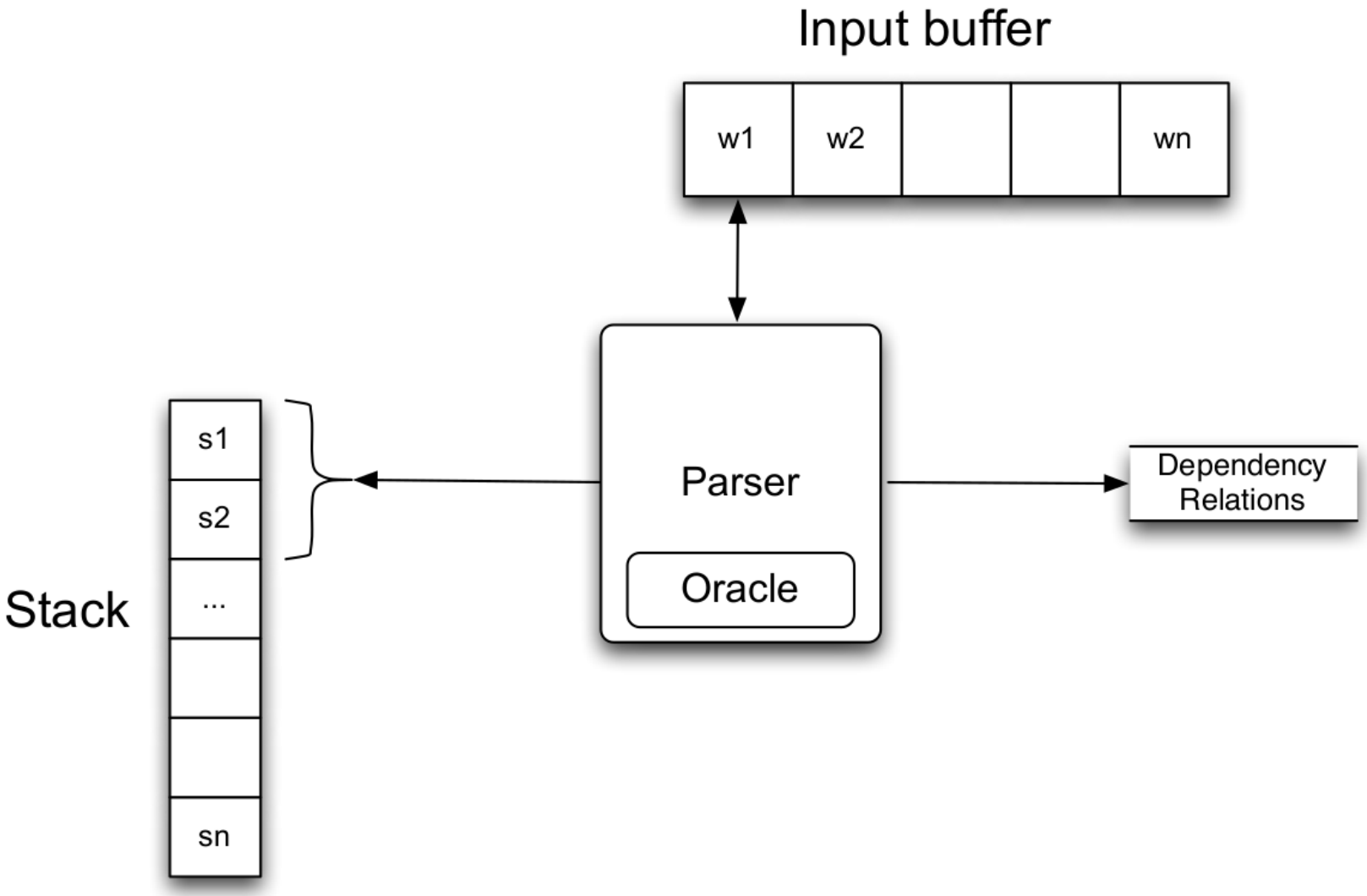
Transition-based dependency parsing

- Basic idea:
  - define a transition system for mapping a sentence to its dependency graph
  - Learning: induce a model for predicting the next state transition, given the transition history
  - Parsing: Construct the optimal transition sequence, given the induced model
- Transition system (arc eager):

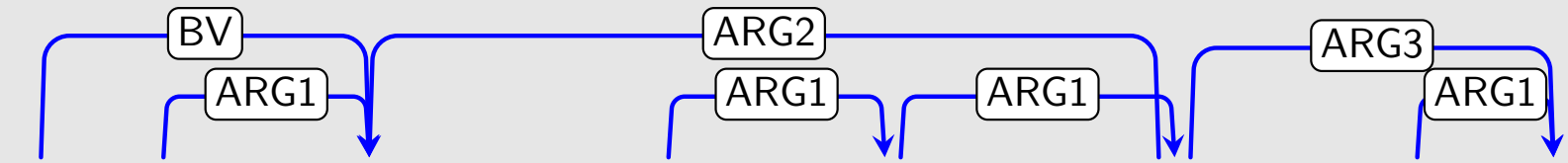
SHIFT	move from front of buffer to top of stack
REDUCE	pop the top of stack (requires existing head)
LEFT-ARC(K)	leftward dependency of type k; reduce
RIGHT-ARC(K)	rightward dependency of type k; shift

- Learning can be performed using a classifier
- Features over input, stack and dependency relations: e.g. $word(S1), dep(s0)$

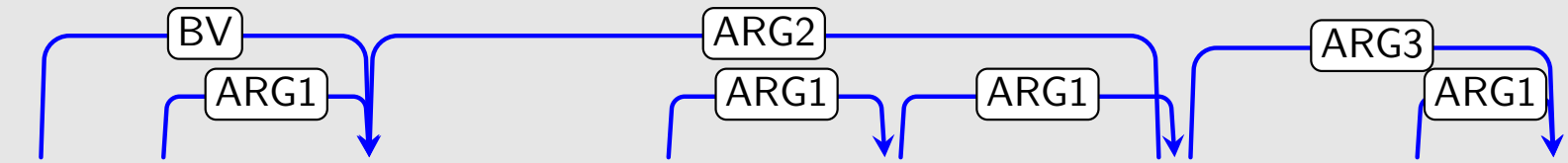
Architecture: Stack and Buffer Configurations



Example question: Properties of Dependency Graphs

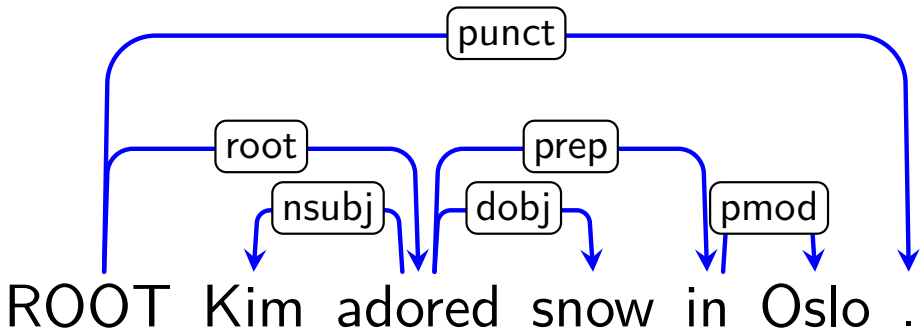


Example question: Properties of Dependency Graphs

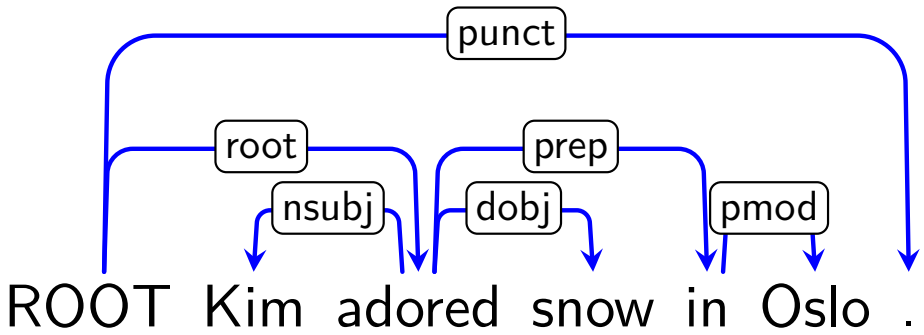


Which of the following formal properties hold for this dependency graph:
(a) Connectedness, (b) Acyclicity,
(c) Single-Headedness
Explain your answers.

Example question: Transition-based dependency parsing

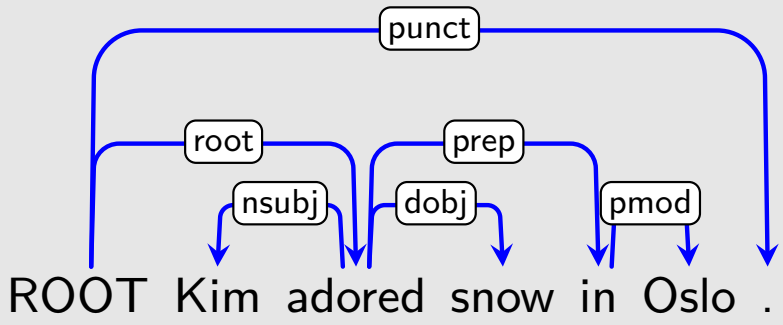


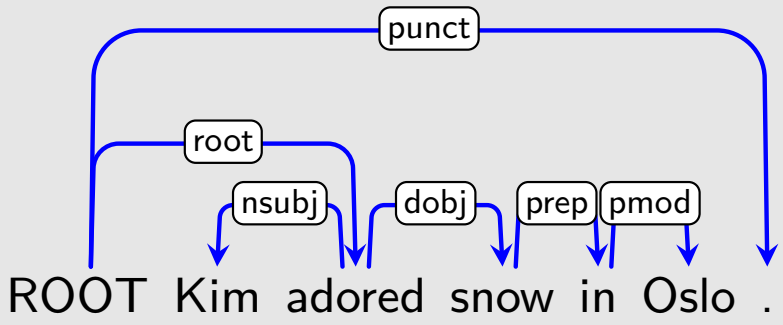
Example question: Transition-based dependency parsing



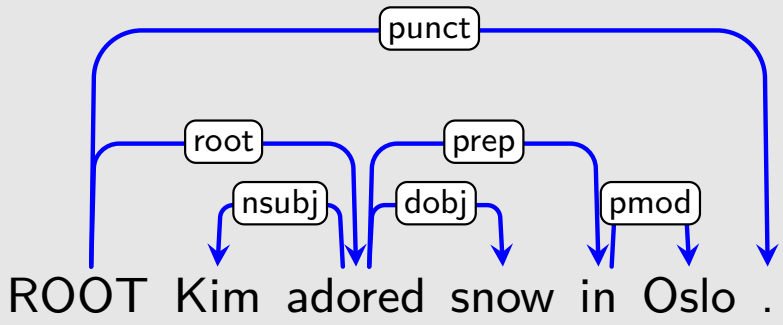
Provide the transition sequence corresponding to the dependency graph above using the arg eager algorithm. How does it differ from the arc standard approach?

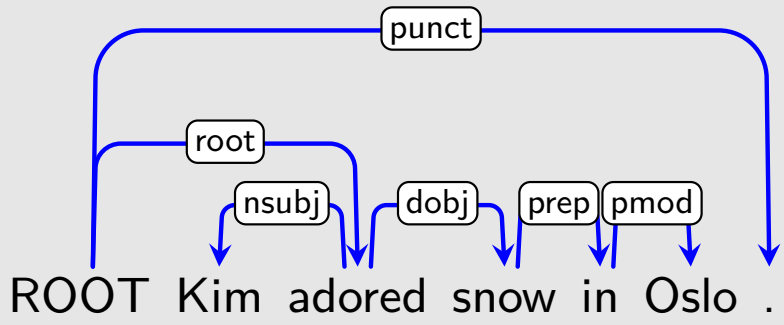
Example question:: Dependency Evaluation





Example question:: Dependency Evaluation





(5) How is dependency parsing often evaluated and what are the scores for the two trees above?

Gold standard on the left, system prediction on the right.

Finally, the 2022 oblig champions are . . .

- 50 submitted and passed and passed all obligatory assignments . . .

Finally, the 2022 oblig champions are . . .

- 50 submitted and passed and passed all obligatory assignments . . .
- all survivors qualified for the final exam . . .

Finally, the 2022 oblig champions are . . .

- 50 submitted and passed and passed all obligatory assignments . . .
- all survivors qualified for the final exam . . .
- . . . some with a larger margin than others

Finally, the 2022 oblig champions are . . .

- 50 submitted and passed and passed all obligatory assignments . . .
- all survivors qualified for the final exam . . .
- . . . some with a larger margin than others
- **For a total of 40.5 points**:
  - hansihag
  - jonassf
  - chrisfeg
  - joannasv
  - rubycp

Finally, the 2022 oblig champions are . . .

- 50 submitted and passed and passed all obligatory assignments . . .
- all survivors qualified for the final exam . . .
- . . . some with a larger margin than others
- **For a total of 40.5 points**:
  - hansihag
  - jonassf
  - chrisfeg
  - joannasv
  - rubycp
- **For a total of 41 points**:
  - leseeger
  - vlhandfo
  - haastr
  - amandusb
  - mosial


That's all Folks!