1 Information Theory

Information Content (Surprisal)

$-log_2 p(x)$: the more frequent the word, the lower its information content. e.g. the word type "blue" occurs ca. 3750 times in 10000 tokens, and its information content is $-log_2(3750/10000) \approx$1.42 bits.

Shannon Entropy

$H(X) = -\sum p(x) log_2 p(x)$: entropy as probability, the average information content of information encoding units in the language. Measure of information encoding potential of a symbol system. The higher the uncertainty, the larger the entropy. e.g. $H_{char}(Morse) = -(\frac{86}{136} * log_2(\frac{86}{136}) + \frac{50}{136} * log_2(\frac{50}{136})) \approx 0.949$ bits per character.

Joint Entropy, Conditional Entropy

Joint Entropy: $H(X,Y) = -\sum\sum p(x,y) log_2 p(x,y)$
Conditional Entropy: $H(Y|X) = -\sum p(x) \sum p(y|x) log_2 p(y|x)$ The more ambiguity in language (uncertainty), the higher conditional entropy. No ambiguity -> 0.

Probability Estimation

Maximum Likelihood (ML)
Problems: unit problem, sample size problem, interdependence problem, extrapolation problem
Methods: frequency-based, language models, experiments with humans

Mutual Information

$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$
reduction in the uncertainty of X given Y. compromise between minimum learning cost H(Y) and maximum expliciteness I(X; Y).
Entropy is the upper bound on the mutual information between forms and meanings
Is the entropy rate zero? -> asymptotic determinism of human utterances.

2 Propositional Logic

Why formal logic? overcome ambiguity, determine relationships between meanings of sentences, determine meanings of setences, model compositionality, recursive system.

Definition

Proposition: The meaning of a simple declarative sentence
Extensions: real-world situations they refer to.
Frege's Generalization: The extension of a sentence S is its truth value
The proposition expressed by a sentence is the set of possible cases [situations] of which that sentence is true.
propositional variables: p, q, r... propositional operators: $\neg$, $conjunction \wedge$, $disjunction \vee$, XOR, $\rightarrow$, $\leftrightarrow$

Syntax: Recursive Definition

1. Propositional letters in the vocabulary of L are formulas in L.
2. If ϕ is a formula in L, then $\neg\phi$ is too.
3. If ϕ and ψ are formulas in L, then $(\phi \wedge \psi), (\phi \vee \psi), (\phi \rightarrow \psi), (\phi \leftrightarrow \psi)$ are too.
4. Only that which can be generated by the clauses (i)-(iii) in a finite number of steps is a formula in L.

invalid: $\neg(\neg\neg p), \neg((p \wedge q))$

Construction Trees

- $(\neg(p \vee q) \rightarrow \neg\neg q) \leftrightarrow r$ $(iii. \leftrightarrow)$
  - $(\neg(p \vee q) \rightarrow \neg\neg q)$ $(iii. \rightarrow)$
    - $\neg(p \vee q)$ (ii)
      - $p \vee q$ $(iii. \vee)$
        - p (i)
        - q (i)
    - $\neg\neg q$ (ii)
      - $\neg q$ (ii)
        - q (i)
  - $r (i)$

Valuation Functions

For every valuation V and for all formulas ϕ:
$V(\phi \leftrightarrow \psi) = 1 iff V(\phi) = V(\psi)$.

3 Predicate Logic

Definition

constant symbols: a, b, c, ...
variable symbols: x, y, z, ...
n-ary/n-place predicate symbols: A, B, C, reflect relations between n elements (n>0)
connectives:$\neg, \wedge, \vee, \rightarrow, ...$
quantifiers: $\forall, \exists$
round brackets (), equal sign =

Syntax: Recursive Definition

1. If A is an n-ary predicate letter in the vocabulary of L, and each of t1,..., tn is a constant or a variable in the vocabulary of L, then At1,..., tn is a formula in L.
2. If ϕ is a formula in L, then $\neg\phi$ is too.
3. If ϕ and ψ are formulas in L, then $(\phi \wedge \psi), (\phi \vee \psi), (\phi \rightarrow \psi), (\phi \leftrightarrow \psi)$ are too.
4. If ϕ is a formula in L and x is a variable, then $\forall x\phi$ and $\exists x\phi$ is too.
5. Only that which can be generated by the clauses (i)-(iv) in a finite number of steps is a formula in L.

invalid: $a, A, \forall(Axy)$

Construction Trees

- $\forall_x \forall_y ((A_{xy} \wedge B_y) \rightarrow \exists_x A_{xb})$ $(iv.\forall)$
  - $\forall_y ((A_{xy} \wedge B_y) \rightarrow \exists_x A_{xb})$ $(iv.\forall)$
    - $(A_{xy} \wedge B_y) \rightarrow \exists_x A_{xb}$ $(iii. \rightarrow)$
      - $A_{xy} \wedge B_y$ $(iii.\wedge)$
        - A_{xy} (i)
        - B_y (i)
      - $\exists_x A_{xb}$ $(iv.\exists)$
        - A_{xb} (i)

Semantics

domain (D): set of entities
interpretation functions I = $\{< m, e >, < s, e >, < v, e >\}$, I(m) = e, I(s) = e, I(v) = e.
model M: consists of a domain D and an interpretation function I which conforms to:

1. if c is a constant in L, then I(c) $\in$ D
2. if B is an n-ary prpedicate letter in L, then I(B) $\subset$ D

valuation function V_M:
If Aa1,...,an is an atomic sentence in L, then V_M(Aa1,...,an) = 1 iff <I(a1), ..., I(an)> $\in$ I(A).
...
$V_M(\forall x\phi)$ = 1 iff V_M([c/x]ϕ) = 1 for all constants c in L.
$V_M(\exists x\phi)$ = 1 iff V_M([c/x]ϕ) = 1 for at least one constant c in L.
If $V_M(\phi) = 1$, then ϕ is said to be true in model M.

Formula vs. Sentence

A sentence is a formula in L which lacks free variables.
Sentence: Aa, $\forall x(Fx)$, $\forall x(Ax \rightarrow \exists y By)$
Not a sentence (but Formula): Ax, Fx, $Ax \rightarrow \exists y By$

4 Second-Order Logic

e.g. CR (CX: X is a predicate with the property of being a color; Rx: x is red)
$\exists X(CX \wedge Xm)$: Mars has a color.
$\exists X(Xj \wedge Xp)$: John has at least one thing in common with Peter.

Vocalbulary extention

First-order predicate variables: X, Y, Z, ... Second-order predicate constants: A, B, C, ... e.g. AX: X is a property with the property of being an animal

Syntax: Recursive Definition

1. If A is an n-ary first-order predicate letter in the vocabulary of L, and each of t1,..., tn are individual terms in L, then At1,..., tn is an (atomic) formula in L.
2. If X is a [first-order] predicate variable and t is an individual term (both constants and variables) in L, then Xt is an atomic formula in L;
3. If A is an n-ary second-order predicate letter/constant in L, and T1,...,Tn are first-order **unary** predicate constants, or predicate variables, in L, then AT1,...,Tn is an (atomic) formula in L;
4. If ϕ is a formula in L, then $\neg\phi$ is too.
5. If ϕ and ψ are formulas in L, then $(\phi \wedge \psi), (\phi \vee \psi), (\phi \rightarrow \psi), (\phi \leftrightarrow \psi)$ are too.
6. If ϕ is a formula in L and x is a variable, then $\forall x\phi$ and $\exists x\phi$ is too.
7. If X is a [first-order] predicate variable, and ϕ is a formula in L, then $\forall X\phi$ and $\exists X\phi$ is too.
8. Only that which can be generated by the clauses (i)-(vii) in a finite number of steps is a formula in L.

invalid: $x, X, Xab, \forall(Xa)$

Semantics

just as a first-order predicate denotes a set of entities, a second-order predicate denotes a set of a set of entities.

5 Abbildungen

Eine Abbildung $f : X \rightarrow Y$ ist eine Vorschrift, die jedem $x \in X$ eindeutig ein bestimmtes $y = f(x) \in Y$ zuordnet. y ist das *Bild* von x und x das *Urbild* von y. Für eine Abbildung gilt, dass jedes Element der Urmenge X genau auf ein $y \in Y$ abbildet, es müssen aber nicht alle Elemente aus Y angenommen werden bzw. darf auch mehrfach angenommen werden (rechtseindeutig, linksvollständig).
Als Relation:
$f \subseteq A \times B$ mit $f = \{(a, f(a)) \mid a \in A \wedge f(a) \in B\}$

Funktionen

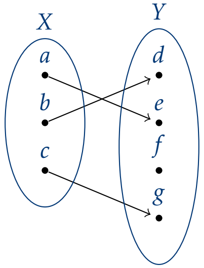
Sei $f \subseteq A \times B$ linkseindeutige und rechtsvollständige Relation.
F ist linksvollständig, wenn gilt $\forall a \in A \exists b \in B : (a,b) \in R$.
F ist rechtseindeutig, wenn gilt $\forall a \in A \forall b_1, b_2 \in B : (a, b_1) \in R \wedge (a, b_2) \in R \Rightarrow b_1 = b_2$.

Bild, Urbild

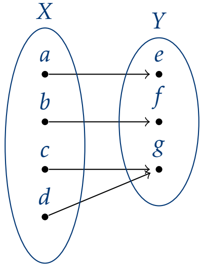
Sei $f : A \rightarrow B$ und $M \subseteq A$.
Das *Bild* von M unter f ist die Menge $f(M) := \{f(x) \mid x \in M\}$.
Das *Urbild* einer Teilmenge $N \subseteq B$ heißt $f^{-1}(N) := \{a \in A \mid f(a) \in N\}$.

Eigenschaften von Abbildungen

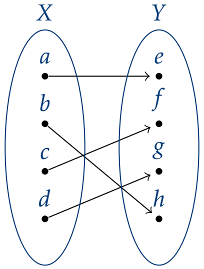
Injektivität:
$\forall x, y \in X : f(x) = f(y) \Rightarrow x = y$
Jedes $y \in Y$ wird höchstens einmal (oder garnicht) getroffen:



Surjektivität:
$\forall y \in Y \exists x \in X : f(x) = y$
Jedes $y \in Y$ wird mindestens einmal getroffen:



Bijektivität:
Jedem $x \in X$ wird genau ein $y \in Y$ zugeordnet und jedem $y \in Y$ genau ein $x \in X$:



Beispiel für Abbildung, die injektiv aber nicht surjektiv ist: Sei $f : \mathbb{N} \rightarrow \mathbb{N}$. Dann ist $f(n) = n + 1$ injektiv, da $f(x) = f(y) \Leftrightarrow x + 1 = y + 1$ gelten muss, was nur gilt, wenn $x = y$. f ist nicht surjektiv da 0 kein Urbild.

Komposition

Die *Komposition* (Hintereinanderausführung) zweier Abbildungen $f : A \rightarrow B$ und $g : B \rightarrow C$ ist $a \mapsto (g \circ f)(a) = g(f(a))$, $a \in A$

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[rr, bend right, "g \circ f"'] & B \arrow[r, "g"] & C \end{tikzcd}$$

Es gilt $(h \circ g) \circ f = h \circ (g \circ f)$. Außerdem gilt: Die Komposition von injektiven Abbildungen ist injektiv, die von surjektiven Abbildungen ist surjektiv und die von bijektiven Abbildungen ist bijektiv.

Identität, Umkehrabbildung

Die Abbildung $id_A : A \to A$ mit $id_A(a) = a$ heißt *Identität*.
Sei $f : A \to B$ bijektive Abbildung. Dann existiert zu f stets eine Abbildung g mit $g \circ f = id_A$ und $f \circ g = id_B$. g heißt die zu f *inverse Abbildung* (f^{-1}). Es gilt $f^{-1}(f(a)) = a$ und $f(f^{-1}(b)) = b$.

Mächtigkeit von Mengen, Abzählbarkeit

Gleichmächtige Mengen:
Seien M und N zwei Mengen. M und N heißen gleichmächtig, wenn es eine bijektive Abbildung $f : M \to N$ gibt ($M \cong N$).
Endliche Mengen:
Eine Menge M heißt endlich, wenn $M = \emptyset$ oder es für ein $n \in \mathbb{N}$ eine bijektive Abbildung $b : M \to \mathbb{N}_n$ gibt.
Unendliche Mengen:
Eine Menge M heißt unendlich, wenn M nicht endlich.
Abzählbare Mengen:
Eine Menge M heißt abzählbar, wenn M endlich oder es gibt bijektive Abbildung $b : M \to \mathbb{N}$.
Abzählbar unendliche Mengen:
Eine Menge M heißt abzählbar unendlich, wenn M abzählbar und M unendlich.
Überabzählbare Mengen:
Eine Menge M heißt überabzählbar, wenn M nicht abzählbar.
Spezielle endliche Mengen:
Sei $n \in \mathbb{N}$. Dann ist $\mathbb{N}_n := [n] := \{1, \ldots, n\}$ die Menge der ersten n natürlichen Zahlen.
Beispiele:
$|\{a, b, c\}| = 3 = |\{x, y, 11\}|$
$|\mathbb{N}| = |\mathbb{R}| = |\mathbb{N} \times \mathbb{N}|$

Kardinalität

Anzahl der Elemente einer Menge. Zwei Mengen haben gleiche Kardinalität, wenn sie gleichmächtig sind.

Beispielbeweis

Zu zeigen: $|\mathbb{N}| = |\mathbb{N} \times \mathbb{N}|$
Beweis. Wir betrachten $f : \mathbb{N} \to \mathbb{N} \times \mathbb{N}$ mit $f(n) := (1, n)$ und $g : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ mit $g(n, m) := 2^n \cdot 3^m$. Beide sind injektiv, also $\mathbb{N} \cong \mathbb{N} \times \mathbb{N}$, also $|\mathbb{N}| = |\mathbb{N} \times \mathbb{N}|$.

6 Graphentheorie

Gerichteter Graph

$G = (V, E)$ wobei V Menge aller Knoten z.B. $V = \{v_0, v_1, v_2, \ldots, v_n\}$ und $E \subseteq V \times V$ Menge aller Kanten mit $e = (v, u)$. Hierbei steht v für den Startknoten von e und u ist Endknoten von e.
Hinweis:
Ist die Kantenmenge E symmetrisch ($(u, v) \in E \wedge (v, u) \in E$) sprechen wir von einem ungerichteten Graphen. In solchen werden keine Schlingen betrachtet.

Adjazente Knoten

Zwei Knoten, die in einem Graphen durch eine Kante verbunden sind, heißen *adjazent* oder *benachbart*.

Endlicher Graph

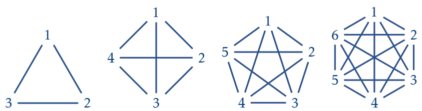
Ein Graph G heißt endlich, wenn die Knotenmenge V endlich ist.

Nullgraph (vollst. unverbunden)

$G = (V, \emptyset) \Rightarrow$ ohne Kanten

Vollständiger Graph

$G = (V, V \times V)$ ist vollständig (heißt auch K_n wegen n Knoten) und hat Maximalzahl von n^2 Kanten $\Rightarrow$ gerichtet und mit Schlingen. Der Ungerichtete K_n hat $\frac{n \cdot (n-1)}{2}$ Kanten, wobei n die Zahl der Knoten ist.
Beispiel:



Ungerichteter Graph

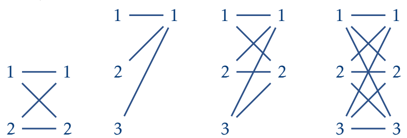
Ein Graph $G = (V, E)$ ist ungerichtet $\Leftrightarrow E$ ist symmetrisch $\Leftrightarrow (u, v) \in E \wedge (v, u) \in E$. Da hier die Kanten ungerichtet, kann Mengenschreibweise verwendet werden.

Schlinge

Kante mit gleichem Start- und Endknoten. Wird bei ungerichteten Graphen nicht betrachtet.

Bipartite Graphen

Ein ungerichteter Graph ist bipartit, wenn die Knotenmenge V in zwei disjunkte Teilmengen U, W zerlegbar ist, sodass alle Kanten $e \in E$ einen Endpunkt in U und einen anderen in W haben.
Beispiel:



Eulersche Graphen

G heißt eulerscher Graph, falls es in G einen geschlossenen Weg gibt, der jede Kante von G enthält.
G ist eulerscher Graph $\Leftrightarrow$ jede Ecke von G hat geraden Grad und G ist zusammenhängend.

Untergraph

Seien $G = (V_G, E_G)$, $H = (V_H, E_H)$ zwei Graphen. H heißt Teilgraph von G, wenn $V_H \subseteq V_G$ und $E_H \subseteq E_G$ (wenn also jede Kante von H auch zu G gehört.)
Hinweis:
Der Nullgraph O_n ist Teilgraph jedes Graphen mit n Knoten. Außerdem ist jeder Graph Teilgraph des vollständigen Graphen K_n.

Induzierte Teilgraphen

Sei $G = (V, E)$ ein Graph. Ist $V' \subseteq V$ eine Teilmenge der Knotenmenge V von G, dann ist der Untergraph oder der durch V' induzierte Teilgraph von G der Graph $G[V'] = (V', E')$ mit $E' = \{(u, v) \mid u, v \in V' \wedge (u, v) \in E\}$
Beispiel:
Ist G ein Graph hat $G[\{2, 3, 4\}]$ nur die Knoten 2, 3 und 4 und die entsprechenden Kanten.

Grad eines Knoten

Der Ausgrad von v ist die Zahl der Kanten, die v als Startknoten besitzen. Der Ingrad von v ist die Zahl der Kanten, die in v enden. Ist G ungerichtet stimmen Ausgrad von v und Ingrad von v überein und wird Grad von v genannt.
Hinweis:
Sei $G = (V, E)$ gerichteter Graph, dann gilt $\sum_{v \in V} indeg(v) = \sum_{v \in V} outdeg(v) = |E|$. Ist G ungerichtet, dann gilt $\sum_{v \in V} deg(v) = 2 \cdot |E|$.

Wege

Ein Weg von den Knoten u nach v ist eine Folge benachbarter Knoten. Die Länge eines Weges ist die Anzahl der Kanten. Ein Weg der Länge 0 wird als trivialer Weg bezeichnet und besteht nur aus einem Knoten.
Hinweis:
Ein Weg heißt geschlossen, wenn seine beiden Endknoten gleich sind.

Graphzusammenhang

Knoten u und v eines ungerichteten Graphen heißen zuammenhängend, wenn es einen Weg in G von u nach v gibt.

Zusammenhangskomponente

Ein Graph G heißt zusammenhängend wenn jedes Knotenpaar aus G zusammenhängend ist.
Hinweis:
Die Äquivalenzklassen (zusammenhängende Teilgraphen) einer Zusammenhangsrelation Z über einem ungerichteten Graphen G heißen Zusammenhangskomponenten (ZK) von G.

Pfade, Kreise

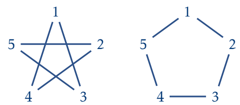
Als *Pfad* werden Wege in einem Graphen bezeichnet, bei denen keine Kante zweimal durchlaufen wird. Ein geschlossener Pfad heißt *Kreis*. Bei einem *einfachen Pfad* wird kein Knoten mehrfach durchlaufen. Ein geschlossener Pfad, der mit Ausnahme seines Ausgangspunktes einfach ist, heißt *einfacher Kreis*. Ein einfacher Kreis durch sämtliche Knoten des Graphen, heißt *Hamiltonscher Kreis*.

Hamiltonscher Kreis

Kann der Zusammenhang eines Graphen G durch die Entnahme eines einzigen Knotens (und sämtlicher mit diesem Knoten benachbarter Kanten) zerstört werden, dann besitzt G keinen Hamiltonschen Kreis.

Isomorphe Graphen

Zwei Graphen heißen isomorph zueinander, wenn sie strukturell gleich sind.
Beispiel:



v	1	2	3	4	5
$\phi(v)$	1	4	2	5	3

Komplementäre Graphen

Sei $G = (V, E)$ ein Graph. Dann ist $\overline{G} = (V, (V \times V) \setminus E)$ der Komplementärgraph von G.
Hinweis:
Ein Graph heißt selbstkomplementär wenn G und $\overline{G}$ isomorph sind.