Introduction to Natural Language Processing

a course taught as B4M36NLP at Open Informatics

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
OPEN INFORMATICS
FACULTY OF ELECTRICAL ENGINEERING

by members of the Institute of Formal and Applied Linguistics

ÚFAL

FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

Today:	**Week 1, lecture**
Today's topic:	**Introduction & Probability & Information theory**
Today's teacher:	**Jan Hajič**
E-mail:	hajic@ufal.mff.cuni.cz
WWW:	http://ufal.mff.cuni.cz/jan-hajic

Intro to NLP

- Instructor: Jan Hajič
  - ÚFAL MFF UK, office: 420 / 422 MS
  - Hours: J. Hajic: Mon 9:00-10:00
  - preferred contact: `hajic@ufal.mff.cuni.cz`
- Room & time:
  - lecture: Wed, 9:15-10:45
  - seminar [cvičení] follows (Zdenek Zabokrtsky)
  - Oct 5, 2016 – Jan 4, 2017
  - Final written exam date: Jan 11, 2017

Textbooks you need

- Manning, C. D., Schütze, H.:
  - *Foundations of Statistical Natural Language Processing*. The MIT Press. 1999. ISBN 0-262-13360-1. **[available at least at MFF / Computer Science School library, Malostranske nam. 25, 11800 Prague 1]**
- Jurafsky, D., Martin, J.H.:
  - *Speech and Language Processing.* Prentice-Hall. 2000. ISBN 0-13-095069-6 and **newer editions**. **[recommended].**
- Cover, T. M., Thomas, J. A.:
  - *Elements of Information Theory.* Wiley. 1991. ISBN 0-471-06259-6.
- Jelinek, F.:
  - *Statistical Methods for Speech Recognition*. The MIT Press. 1998. ISBN 0-262-10066-5

Other reading

- Journals:
  - Computational Lingusitics
  - Transactions on Computational Linguistics
- Proceedings of major conferences:
  - ACL (Assoc. of Computational Linguistics)
  - EACL (European Chapter of ACL)
  - EMNLP (Empirical Methods in NLP)
  - CoNLL (Natural Language Learning in CL)
  - IJCNLP (Asian cahpter of ACL)
  - COLING (Intl. Committee of Computational Linguistics)

Course segments (first three lectures)

- Intro & Probability & Information Theory
  - The very basics: definitions, formulas, examples.
- Language Modeling
  - n-gram models, parameter estimation
  - smoothing (EM algorithm)
- Hidden Markov Models
  - background, algorithms, parameter estimation

Probability

Experiments & Sample Spaces

- Experiment, process, test, ...
- Set of possible basic outcomes: sample space Ω
  - coin toss (Ω = {head,tail}), die (Ω = {1..6})
  - yes/no opinion poll, quality test (bad/good) (Ω = {0,1})
  - lottery (| Ω | ≅ 10^7 .. □□^{12})
  - # of traffic accidents somewhere per year (Ω = N)
  - spelling errors (Ω = Z^*), where Z is an alphabet, and Z^* is a set of possible strings over such and alphabet
  - missing word (| Ω | ≅ vocabulary size)

Events

- Event A is a set of basic outcomes
- Usually $A \subset \Omega$, and all $A \in 2^{\Omega}$ (the event space)
  - Ω is then the certain event, $\oslash$ is the impossible event
- Example:
  - experiment: three times coin toss
    - **Ω = {HHH, HHT, HTH, HTT, THH, THT, TTH, TTT}**
  - count cases with exactly two tails: then
    - **A = {HTT, THT, TTH}**
  - all heads:
    - **A = {HHH}**

Probability

- Repeat experiment many times, record how many times a given event A occurred ("count" c_1).
- Do this whole series many times; remember all c_is.
- Observation: if repeated really many times, the ratios of $\mathbf{c_i/T_i}$ (where T_i is the number of experiments run in the *i-th* series) are close to some (unknown but) **constant** value.
- Call this constant a ***probability of A***. Notation: **p(A)**

Estimating probability

- Remember: ... close to an *unknown* constant.
- We can only estimate it:
  - from a single series (typical case, as mostly the outcome of a series is given to us and we cannot repeat the experiment), set

    $$p(A) = c_1/T_1.$$

  - otherwise, take the weighted average of all c_i/T_i (or, if the data allows, simply look at the set of series as if it is a single long series).
- This is the **best** estimate.

Example

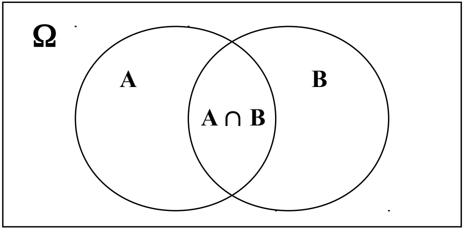
- Recall our example:
  - experiment: three times coin toss
    - **Ω = {HHH, HHT, HTH, HTT, THH, THT, TTH, TTT}**
  - count cases with exactly two tails: A = **{HTT, THT, TTH}**
- Run an experiment 1000 times (i.e. 3000 tosses)
- Counted: 386 cases with two tails (**HTT, THT,** or **TTH**)
- estimate: p(A) = 386 / 1000 = .386
- Run again: 373, 399, 382, 355, 372, 406, 359
  - p(A) = .379 (weighted average) or simply 3032 / 8000
- *Uniform* distribution assumption: p(A) = 3/8 = .375

Basic Properties

- Basic properties:
  - $p: 2^{\Omega} \rightarrow [0,1]$
  - $p(\Omega) = 1$
  - Disjoint events: $p(\cup A_i) = \sum_i p(A_i)$
- [NB: *axiomatic definition* of probability: take the above three conditions as axioms]
- Immediate consequences:
  - $p(\emptyset) = 0$, $p(\overline{A}) = 1 - p(A)$, $A \subseteq B \Rightarrow p(A) \leq p(B)$
  - $\sum_{a \in \Omega} p(a) = 1$

Joint and Conditional Probability

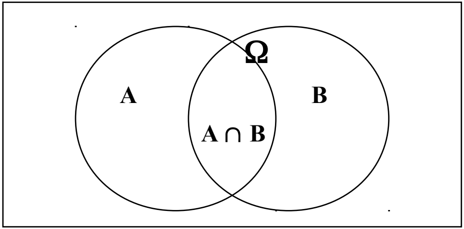
- $p(A,B) = p(A \cap B)$
- $p(A|B) = p(A,B) / p(B)$
  - Estimating form counts:
    - $\mathbf{p(A|B) = p(A,B) / p(B) = (c(A \cap B) / T) / (c(B) / T) =}$
      $\mathbf{= c(A \cap B) / c(B)}$



Bayes Rule

- p(A,B) = p(B,A) since p(A ∩ B) = p(B ∩ A)
  - therefore: p(A|B) p(B) = p(B|A) p(A), and therefore

$$\boxed{p(A|B) = p(B|A)\ p(A) / p(B)}\ !$$



Independence

- Can we compute p(A,B) from p(A) and p(B)?
- Recall from previous foil:

$$p(A|B) = p(B|A)\ \ p(A) / p(B)$$

$$p(A|B)\ p(B) = p(B|A)\ \ p(A)$$

$$p(A,B) = p(B|A)\ \ p(A)$$

... we're almost there: how p(B|A) relates to p(B)?

  - p(B|A) = P(B) iff A and B are ***independent***

- Example: two coin tosses, weather today and weather on March 4th 1789;
- Any two events for which p(B|A) = P(B)!

Chain Rule

$$p(A_1, A_2, A_3, A_4, \ldots, A_n) = \qquad \textbf{!}$$

$$p(A_1|A_2,A_3,A_4,\ldots,A_n) \times\ p(A_2|A_3,A_4,\ldots,A_n) \times$$
$$\times\ p(A_3|A_4,\ldots,A_n) \times\ \ldots\ p(A_{n-1}|A_n) \times p(A_n)$$

- this is a direct consequence of the Bayes rule.

The Golden Rule
(of Classic Statistical NLP)

- Interested in an event A given B (when it is not easy or practical or desirable to estimate $p(A|B)$):
- take Bayes rule, max over all As:
- $\mathrm{argmax}_A\ p(A|B) = \mathrm{argmax}_A\ p(B|A)\ .\ p(A) \ /\ p(B) \xcancel{=}$

$$\boxed{\mathrm{argmax}_A\ p(B|A)\ p(A)\ !}$$

- ... as $p(B)$ is constant when changing As

Random Variable

- is a function $X: \Omega \rightarrow Q$
  - in general: $Q = R^n$, typically R
  - easier to handle real numbers than real-world events
- random variable is *discrete* if Q is countable (i.e. also if finite)
- Example: *die*: natural "numbering" [1,6], *coin*: {0,1}
- Probability distribution:
  - $p_X(x) = p(X=x) =_{df} p(A_x)$ where $A_x = \{a \in \Omega : X(a) = x\}$
  - often just p(x) if it is clear from context what X is

Expectation
Joint and Conditional Distributions

- is a mean of a random variable (weighted average)
  - $E(X) = \Sigma_{x \in X(\Omega)} x \, . \, p_X(x)$
- Example: one six-sided die: 3.5, two dice (sum) 7
- Joint and Conditional distribution rules:
  - analogous to probability of events
- Bayes: $p_{X|Y}(x,y) =_{\text{notation}} p_{XY}(x|y) =_{\text{even simpler notation}}$

  $\mathbf{p(x|y) = p(y|x) \, . \, p(x) \, / \, p(y)}$
- Chain rule: $\mathbf{p(w,x,y,z) = p(z).p(y|z).p(x|y,z).p(w|x,y,z)}$

Essential Information Theory

The Notion of Entropy

- Entropy ~ "chaos", fuzziness, opposite of order, ...
  - you know it:
    - **it is much easier to create "mess" than to tidy things up...**
- Comes from physics:
  - Entropy does not go down unless energy is applied
- Measure of ***uncertainty:***
  - if low... low uncertainty; the higher the entropy, the higher uncertainty, but the higher "surprise" (information) we can get out of an experiment

The Formula

- Let $p_X(x)$ be a distribution of random variable X
- Basic outcomes (alphabet) Ω

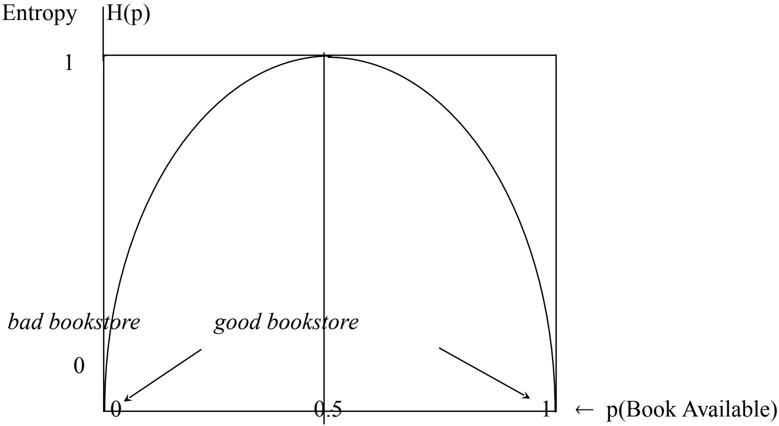
$$\boxed{H(X) = - \sum_{x \in \Omega} p(x) \log_2 p(x) \qquad !}$$

- Unit: bits ($\log_{10}$: nats)
- Notation: $H(X) = H_p(X) = H(p) = H_X(p) = H(p_X)$

Using the Formula: Example

- Toss a fair coin: $\Omega = \{head, tail\}$
  - p(head) = .5, p(tail) = .5
  - **H(p)** = $- 0.5 \log_2(0.5) + (- 0.5 \log_2(0.5)) = 2 \times ((-0.5) \times (-1)) = 2 \times 0.5 =$ **1**
- Take fair, 32-sided die: p(x) = 1 / 32 for every side x
  - **H(p)** = $-\sum_{i = 1..32} p(x_i) \log_2 p(x_i) = - 32 (p(x_1) \log_2 p(x_1)$ (since for all *i* $p(x_i) = p(x_1) = 1/32$) $= -32 \times ((1/32) \times (-5)) =$ **5** *(now you see why it's called **bits**?)*
- Unfair coin:
  - p(head) = .2 ... **H(p) = .722**; p(head) = .01 ... **H(p) = .081**

Example: Book Availability



The Limits

- When H(p) = 0?
  - if a result of an experiment is ***known*** ahead of time:
  - necessarily:

    $\exists x \in \Omega; p(x) = 1 \,\&\, \forall y \in \Omega; y \neq x \Rightarrow p(y) = 0$
- Upper bound?
  - none in general
  - for $|\Omega| = n$: $H(p) \leq \log_2 n$
    - **nothing can be more uncertain than the uniform distribution**

Perplexity: motivation

- Recall:
  - 2 equiprobable outcomes: H(p) = 1 bit
  - 32 equiprobable outcomes: H(p) = 5 bits
  - 4.3 billion equiprobable outcomes: H(p) ~= 32 bits
- What if the outcomes are not equiprobable?
  - 32 outcomes, 2 equiprobable at .5, rest impossible:
    - **H(p) = 1 bit**
  - Any measure for comparing the entropy (i.e. uncertainty/difficulty of prediction) (also) for random variables with *different number of outcomes*?

Perplexity

- Perplexity:
  - $G(p) = 2^{H(p)}$
- ... so we are back at 32 (for 32 eqp. outcomes), 2 for fair coins, etc.
- it is easier to imagine:
  - NLP example: vocabulary size of a vocabulary with uniform distribution, which is equally hard to predict
- the “wilder” (biased) distribution, the better:
  - lower entropy, lower perplexity

Joint Entropy and Conditional Entropy

- Two random variables: X (space Ω),Y (Ψ)
- Joint entropy:
  - no big deal: ((X,Y) considered a single event):

$$H(X,Y) = - \sum_{x \in \Omega} \sum_{y \in \Psi} p(x,y) \log_2 p(x,y)$$

- Conditional entropy:

$$H(Y|X) = - \sum_{x \in \Omega} \sum_{y \in \Psi} \underline{p(x,y)} \log_2 p(y|x)$$

recall that $H(X) = E(\log_2(1/p_X(x)))$

(weighted average: <u>weights are not conditional</u>)

Properties of Entropy I

- Entropy is non-negative:
  - $H(X) \geq 0$
  - proof: (recall: $H(X) = - \Sigma_{x \in \Omega} p(x) \log_2 p(x)$)
    - **log(p(x)) is negative or zero for x ≤ 1,**
    - **p(x) is non-negative; their product p(x)log(p(x) is thus negative;**
    - **sum of negative numbers is negative;**
    - **and -*f* is positive for negative *f***
- Chain rule:
  - $H(X,Y) = H(Y|X) + H(X)$, as well as
  - $H(X,Y) = H(X|Y) + H(Y)$ (since $H(Y,X) = H(X,Y)$)

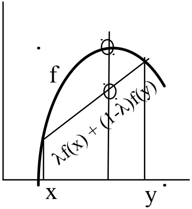
Properties of Entropy II

- Conditional Entropy is better (than unconditional):
  - $H(Y|X) \leq H(Y)$
- $H(X,Y) \leq H(X) + H(Y)$ (follows from the previous (in)equalities)
  - **equality iff X,Y independent**
  - **[recall: X,Y independent iff $p(X,Y) = p(X)p(Y)$]**
- H(p) is concave (remember the book availability graph?)
  - concave function f over an interval (a,b):

$$\forall x,y \in (a,b),\ \forall \lambda \in [0,1]:$$

$$f(\lambda x + (1-\lambda)y) \geq \lambda f(x) + (1-\lambda)f(y)$$

  - **function f is convex if -f is concave**



“Coding” Interpretation of Entropy

- The least (average) number of bits needed to encode a message (string, sequence, series,...) (each element having being a result of a random process with some distribution p): = H(p)
- Remember various compressing algorithms?
  - they do well on data with repeating (= easily predictable = low entropy) patterns
  - their results though have high entropy ⇒ compressing compressed data does nothing

Coding: Example

- How many bits do we need for ISO Latin 1?
  - ⇒ the trivial answer: 8
- Experience: some chars are more common, some (very) rare:
  - **...so what if we use more bits for the rare, and less bits for the frequent? [be careful: want to decode (easily)!]**
  - **suppose: p('a') = 0.3, p('b') = 0.3, p('c') = 0.3, the rest: p(x)≅ .0004**
  - **code: 'a' ~ 00, 'b' ~ 01, 'c' ~ 10, rest: $11b_1b_2b_3b_4b_5b_6b_7b_8$**
  - **code acbbécbaac:** `0010010111000011111001000010`

```
a c b b     é     c b a a c
```

  - **number of bits used: 28 (vs. 80 using "naive" coding)**
- code length ~ 1 / probability; conditional prob OK!

Kullback-Leibler Distance (Relative Entropy)

- Remember:
  - long series of experiments... c_i/T_i oscillates around some number... we can only estimate it... to get a distribution q.
- So we get a distribution q; (sample space Ω, r.v. X)

  the true distribution is, however, p. (same Ω, X)

  ⇒ how big error are we making?
- D(p||q) (the Kullback-Leibler distance):

$$D(p\|q) = \sum_{x \in \Omega} \underline{p(x)} \log_2 (p(x)/q(x)) = E_p \log_2 (p(x)/q(x))$$

Comments on Relative Entropy

- Conventions:
  - $0 \log 0 = 0$
  - $p \log (p/0) = \infty$ (for $p > 0$)
- Distance? (less "misleading": Divergence)
  - not quite:
    - **not symmetric: D(p||q) ≠ D(q||p)**
    - **does not satisfy the triangle inequality**
  - but useful to look at it that way
- H(p) + D(p||q): bits needed for encoding <u>p</u> if <u>q</u> is used

Mutual Information (MI)
in terms of relative entropy

- Random variables X, Y; $p_{X\cap Y}(x,y)$, $p_X(x)$, $p_Y(y)$
- Mutual information (between two random variables X,Y):

$$I(X,Y) = D(p(x,y) \,\|\, p(x)p(y))$$

- I(X,Y) measures how much (our knowledge of) Y contributes (on average) to easing the prediction of X
- or, how <u>p(x,y)</u> deviates from (independent) <u>p(x)p(y)</u>

Mutual Information: the Formula

- Rewrite the definition: [recall: $D(r||s) = \sum_{v \in \Omega} r(v) \log_2 (r(v)/s(v))$; substitute $r(v) = p(x,y)$, $s(v) = p(x)p(y)$; $<v> \sim <x,y>$]

$$\boxed{\begin{aligned} I(X,Y) &= D(p(x,y) \,||\, p(x)p(y)) = \\ &= \sum_{x \in \Omega} \sum_{y \in \Psi} p(x,y) \log_2 (p(x,y)/p(x)p(y)) \quad ! \end{aligned}}$$

- Measured in bits (what else? :-)

From Mutual Information to Entropy

- by how many bits the knowledge of Y ***lowers*** the entropy H(X):

$$I(X,Y) = \sum_{x \in \Omega} \sum_{y \in \Psi} p(x,y) \log_2 (\underline{p(x,y)/p(y)}p(x)) =$$

...use p(x,y)/p(y) = p(x|y)

$$= \sum_{x \in \Omega} \sum_{y \in \Psi} p(x,y) \log_2 (\underline{p(x|y)/p(x)}) =$$

...use log(a/b) = log a - log b (a ~ p(x|y), b ~ p(x)), distribute sums

$$= \sum_{x \in \Omega} \sum_{y \in \Psi} p(x,y)\log_2 p(x|y) - \sum_{x \in \Omega} \sum_{y \in \Psi} p(x,y)\log_2 p(x) =$$

...use def. of H(X|Y) (left term)*, and $\sum_{y \in \Psi} p(x,y) = p(x)$* (right term)

$$= - H(X|Y) + (- \sum_{x \in \Omega} p(x)\log_2 p(x)) =$$

*...use def. of H(X) (*right term*), swap terms*

$= H(X) - H(X|Y)$ ...by symmetry, $= H(Y) - H(Y|X)$

Properties of MI vs. Entropy

- $I(X,Y) = H(X) \underline{- H(X|Y)}$ = number of bits the knowledge of Y lowers the entropy of X

  $= H(Y) - H(Y|X)$ (prev. foil, symmetry)

Recall: $H(X,Y) = H(X|Y) + H(Y) \Rightarrow -H(X|Y) = H(Y) - H(X,Y) \Rightarrow$

- $I(X,Y) = H(X) + \underline{H(Y) - H(X,Y)}$
- $I(X,X) = H(X)$ (since $H(X|X) = 0$)
- $I(X,Y) = I(Y,X)$ (just for completeness)
- $I(X,Y) \geq 0$ ... let’s prove that now (as promised).

Other (In)Equalities and Facts

- Log sum inequality: for r_i, $s_i \geq 0$

  $\sum_{i=1..n} (r_i \log(r_i/s_i)) \leq (\sum_{i=1..n} r_i) \log(\sum_{i=1..n} r_i / \sum_{i=1..n} s_i))$
- $D(p||q)$ is convex [in p,q] ($\Leftarrow$ log sum inequality)
- $H(p_X) \leq \log_2|\Omega|$, where Ω is the sample space of p_X

  Proof: uniform $u(x)$, same sample space Ω: $\sum p(x) \log u(x) = -\log_2|\Omega|$;

  $\log_2|\Omega| - H(X) = -\sum p(x) \log u(x) + \sum p(x) \log p(x) = D(p||u) \geq 0$
- $H(p)$ is concave [in p]:

  Proof: from $H(X) = \log_2|\Omega| - D(p||u)$, $D(p||u)$ convex $\Rightarrow H(x)$ concave

Cross-Entropy

- Typical case: we've got series of observations

  $T = \{t_1, t_2, t_3, t_4, \ldots, t_n\}$(numbers, words, ...; $t_i \in \Omega$);

  estimate (simple):

  $\forall y \in \Omega$: $\tilde{p}(y) = c(y) / |T|$, def. $c(y) = |\{t \in T; t = y\}|$
- ...but the true p is unknown; every sample is too small!
- Natural question: how well do we do using $\tilde{p}$ [instead of p]?
- Idea: simulate actual p by using a different T'

  (or rather: by using different observation we simulate the insufficiency of T vs. some other data ("random" difference))

Cross Entropy: The Formula

- $H_{p'}(\tilde{p}) = H(p') + D(p' \| \tilde{p})$

$$\boxed{H_{p'}(\tilde{p}) = - \sum_{x \in \Omega} p'(x) \log_2 \tilde{p}(x) \quad !}$$

- p' is certainly not the true p, but we can consider it the "real world" distribution against which we test $\tilde{p}$
- note on notation (confusing...): p/p' ↔ $\tilde{p}$, also $H_{T'}(p)$
- (Cross)Perplexity: $G_{p'}(p) = G_{T'}(p) = 2^{H_{p'}(\tilde{p})}$

Conditional Cross Entropy

- So far: “unconditional” distribution(s) p(x), p’(x)...
- In practice: virtually always conditioning on context
- Interested in: sample space Ψ, r.v. Y, y ∈ Ψ;

  context: sample space Ω, r.v. X, x ∈ Ω;:

  “our” distribution p(y|x), test against p’(y,x),

  which is taken from some independent data:

$$H_{p'}(p) = - \sum_{y \in \Psi,\ x \in \Omega} p'(y,x) \log_2 p(y|x)$$

Sample Space vs. Data

- In practice, it is often inconvenient to sum over the sample space(s) Ψ, Ω (especially for cross entropy!)
- Use the following formula:

$$H_{p'}(p) = \boxed{- \sum_{y \in \Psi,\ x \in \Omega} p'(y,x) \log_2 p(y|x) = - 1/|T'| \sum_{i = 1..|T'|} \log_2 p(y_i|x_i) \quad !}$$

- This is in fact the normalized log probability of the "test" data:

$$H_{p'}(p) = - 1/|T'| \log_2 \prod_{i = 1..|T'|} p(y_i|x_i)$$

Computation Example

- $\Omega = \{a,b,..,z\}$, prob. distribution (assumed/estimated from data):

  $p(a) = .25$, $p(b) = .5$, $p(\alpha) = 1/64$ for $\alpha \in \{c..r\}$, $= 0$ for the rest: s,t,u,v,w,x,y,z
- Data (test): <u>barb</u> $p'(a) = p'(r) = .25$, $p'(b) = .5$
- Sum over Ω:

  α **a b c d e f g ... p q r s t ... z**

  $-p'(\alpha)\log_2 p(\alpha)$ **.5+.5+0+0+0+0+0+0+0+0+0+1.5+0+0+0+0+0 = <u>2.5</u>**
- Sum over data:

  i / s_i **1/b 2/a 3/r 4/b**

  $-\log_2 p(s_i)$ **1 + 2 + 6 + 1 = 10 (1/4) × 10 = <u>2.5</u>**

  $1/|T'|$

Cross Entropy: Some Observations

- $H(p)$?? <, =, > ?? $H_{p'}(p)$: ALL!
- Previous example:

  [$p(a) = .25$, $p(b) = .5$, $p(\alpha) = 1/64$ for $\alpha \in \{c..r\}$, $= 0$ for the rest: s,t,u,v,w,x,y,z]

  $$H(p) = 2.5 \text{ bits} = H(p') \text{ (\underline{barb})}$$

- Other data: <u>probable</u>: **(1/8)(6+6+6+1+2+1+6+6)= 4.25**

  $$H(p) < 4.25 \text{ bits} = H(p') \text{ (\underline{probable})}$$

- And finally: <u>abba</u>: **(1/4)(2+1+1+2)= 1.5**

  $$H(p) > 1.5 \text{ bits} = H(p') \text{ (\underline{abba})}$$

- But what about: <u>baby</u> $-p'(\text{‘y’})\log_2 p(\text{‘y’}) = -.25\log_2 0 = \infty$ (??)

Cross Entropy: Usage

- Comparing data??
  - *NO!* (we believe that we test on ***real*** data!)
- Rather: comparing distributions (***vs.*** real data)
- Have (got) 2 distributions: p and q (on some Ω, X)
  - which is better?
  - better: has lower cross-entropy (perplexity) on real data S
- “Real” data: S
- $H_S(p) = - 1/|S| \Sigma_{i = 1..|S|} \log_2 p(y_i|x_i) \boxed{??} H_S(q) = - 1/|S| \Sigma_{i = 1..|S|} \log_2 q(y_i|x_i)$

Comparing Distributions

Test data S: probable

- p(.) from prev. example: $\boxed{H_S(p) = 4.25}$

  p(a) = .25, p(b) = .5, p(α) = 1/64 for α ∈{c..r}, = 0 for the rest: s,t,u,v,w,x,y,z

- q(.|.) (conditional; defined by a table):

q(.|.)→ ↓	a	b	e	l	o	p	r	other
a	0	.5	0	0	0	.125	0	0
b	1	0	0	0	1	.125	0	0
e	0	0	0	1	0	.125	0	0
l	0	.5	0	0	0	.125	0	0
o	0	0	0	0	0	.125	1	0
p	0	0	0	0	0	.125	0	1
r	0	0	0	0	0	.125	0	0
other	0	0	1	0	0	.125	0	0

ex.: q(o|r) = 1

q(r|p) = .125

(1/8) (log(p|oth.)+log(r|p)+log(o|r)+log(b|o)+log(a|b)+log(b|a)+log(l|b)+log(e|l))

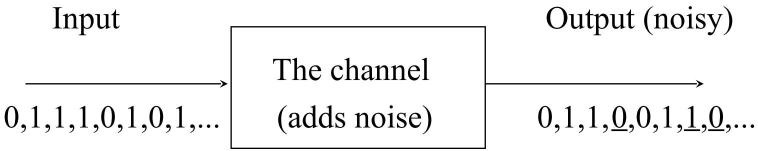
(1/8) (0 + 3 + 0 + 0 + 1 + 0 + 1 + 0)

$H_S(q) = .625$

Language Modeling
(and the Noisy Channel)

The Noisy Channel

- Prototypical case:



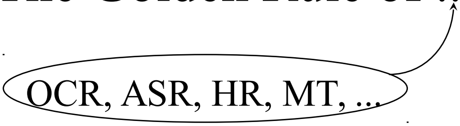
- Model: probability of error (noise):
- Example: p(0|1) = .3 p(1|1) = .7 p(1|0) = .4 p(0|0) = .6
- The Task:

known: the noisy output; want to know: the input (***decoding***)

Noisy Channel Applications

- OCR
  - straightforward: text → print (adds noise), scan → image
- Handwriting recognition
  - text → neurons, muscles (“noise”), scan/digitize → image
- Speech recognition (dictation, commands, etc.)
  - text → conversion to acoustic signal (“noise”) → acoustic waves
- Machine Translation
  - text in target language → translation (“noise”) → source language
- Also: Part of Speech Tagging
  - sequence of tags → selection of word forms → text

Noisy Channel: The Golden Rule of ...


OCR, ASR, HR, MT, ...

- Recall:

  $p(A|B) = p(B|A)\ p(A) / p(B)$ (Bayes formula)

  $A_{best} = \text{argmax}_A\ p(B|A)\ p(A)$ (The Golden Rule)

- $p(B|A)$: the acoustic/image/translation/lexical model
  - application-specific name
  - will explore later
- $p(A)$: ***<u>the language model</u>***

The Perfect Language Model

- Sequence of word forms [forget about tagging for the moment]
- Notation: $A \sim W = (w_1,w_2,w_3,\ldots,w_d)$
- The big (modeling) question:

$$p(W) = ?$$

- Well, we know (Bayes/chain rule →):

$$p(W) = p(w_1,w_2,w_3,\ldots,w_d) =$$

$$= p(w_1) \times\ p(w_2|w_1)\ \times\ p(w_3|w_1,w_2)\ \times\ldots\times\ p(w_d|w_1,w_2,\ldots,w_{d-1})$$

- Not practical (even short W → too many parameters)

Markov Chain

- Unlimited memory (cf. previous foil):
  - for w_i, we know all its predecessors $w_1,w_2,w_3,\ldots,w_{i-1}$
- Limited memory:
  - we disregard “too old” predecessors
  - remember only *k* previous words: $w_{i-k},w_{i-k+1},\ldots,w_{i-1}$
  - called “k^{th} order Markov approximation”
- + stationary character (no change over time):

$$p(W) \cong \Pi_{i=1..d} p(w_i|w_{i-k},w_{i-k+1},\ldots,w_{i-1}),\ d = |W|$$

n-gram Language Models

- (n-1)th order Markov approximation → n-gram LM:

$$\boxed{p(W) =_{df} \prod_{i=1..d} p(\overset{\text{prediction}}{w_i}|\overbrace{w_{i-n+1},w_{i-n+2},\ldots,w_{i-1}}^{\text{history}})\ \textbf{!}}$$

- In particular (assume vocabulary |V| = 60k):
  - **0-gram LM: uniform model, $p(w) = 1/|V|$, 1 parameter**
  - **1-gram LM: unigram model, $p(w)$, 6×10^4 parameters**
  - **2-gram LM: bigram model, $p(w_i|w_{i-1})$ 3.6×10^9 parameters**
  - **3-gram LM: trigram model, $p(w_i|w_{i-2},w_{i-1})$ 2.16×10^{14} parameters**

Maximum Likelihood Estimate

- MLE: Relative Frequency...
  - ...best predicts the data at hand (the "training data")
- Trigrams from Training Data T:
  - count sequences of three words in T: $c_3(w_{i-2},w_{i-1},w_i)$

    [NB: notation: just saying that the three words follow each other]
  - count sequences of two words in T: $c_2(w_{i-1},w_i)$:
    - **either use $c_2(y,z) = \Sigma_w\, c_3(y,z,w)$**
    - **or count differently at the beginning (& end) of data!** **$p(w_i|w_{i-2},w_{i-1})$**

$$\boxed{=_{est.}\ c_3(w_{i-2},w_{i-1},w_i)\ /\ c_2(w_{i-2},w_{i-1})\ !}$$

LM: an Example

- Training data:

  <s> <s> He can buy the can of soda.

  - Unigram: $p_1(\text{He}) = p_1(\text{buy}) = p_1(\text{the}) = p_1(\text{of}) = p_1(\text{soda}) = p_1(.) = .125$
    $p_1(\text{can}) = .25$
  - Bigram: $p_2(\text{He}|\text{<s>}) = 1$, $p_2(\text{can}|\text{He}) = 1$, $p_2(\text{buy}|\text{can}) = .5$,
    $p_2(\text{of}|\text{can}) = .5$, $p_2(\text{the}|\text{buy}) = 1$,...
  - Trigram: $p_3(\text{He}|\text{<s>},\text{<s>}) = 1$, $p_3(\text{can}|\text{<s>},\text{He}) = 1$,
    $p_3(\text{buy}|\text{He},\text{can}) = 1$, $p_3(\text{of}|\text{the},\text{can}) = 1$, ..., $p_3(.|\text{of},\text{soda}) = 1$.
  - Entropy: $H(p_1) = 2.75$, $H(p_2) = .25$, $H(p_3) = 0$ ← Great?!

LM: an Example (The Problem)

- Cross-entropy:
- S = <s> <s> It was the greatest buy of all.
- Even $H_S(p_1)$ fails (= $H_S(p_2) = H_S(p_3) = \infty$), because:
  - all unigrams but p_1(the), p_1(buy), p_1(of) and p_1(.) are 0.
  - all bigram probabilities are 0.
  - all trigram probabilities are 0.
- We want: to make all (theoretically possible*) probabilities non-zero.

*in fact, all: remember our graph from day 1?