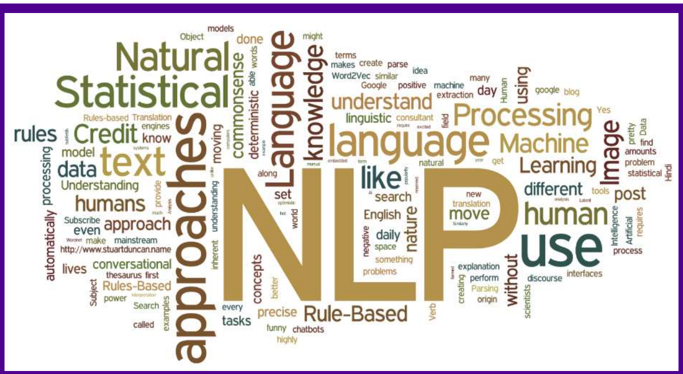
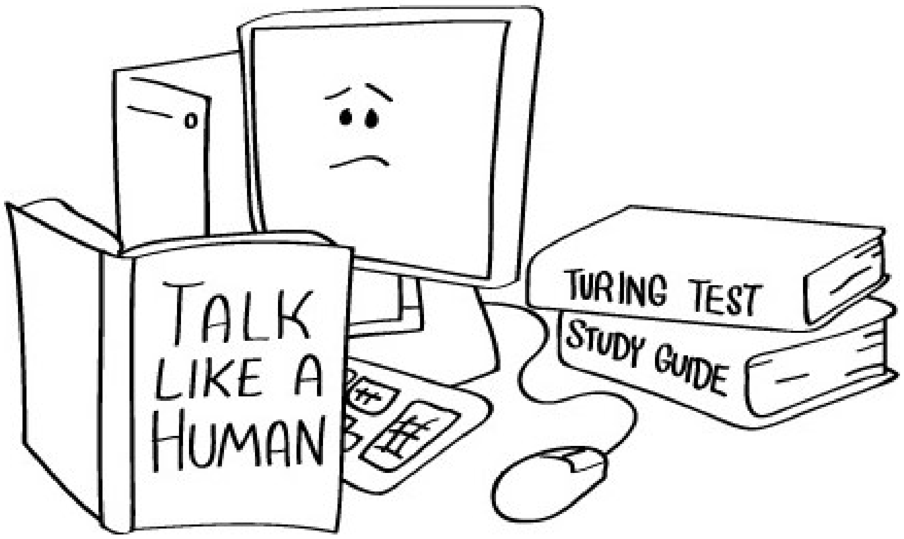


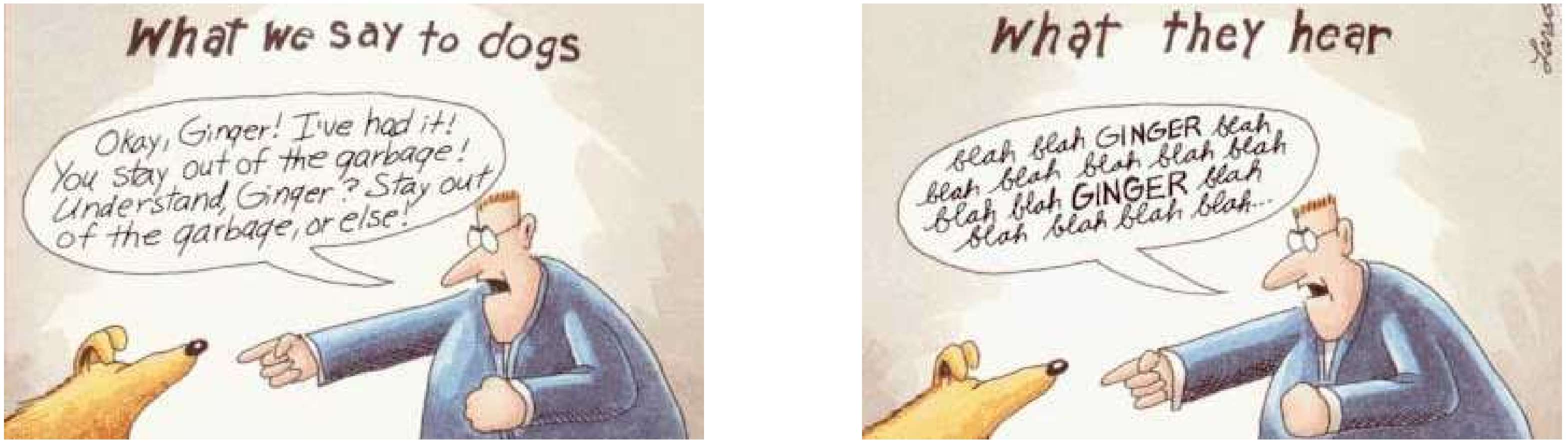
Natural Language Processing

CHAPTER 22 IN THE TEXTBOOK

- Human speech ~100,000 years
- Writing ~ 7,000 years
- Huge numbers of pages of information on the Web, almost all of it in natural language
- For knowledge acquisition an agent needs to understand (partially) the ambiguous, messy languages that humans use
- Specific information-seeking tasks:
  - **text classification**,
  - **information retrieval**, and
  - **information extraction**
- Requires to use language models, which predict the probability distribution of language expressions



What is NLP?


What we say to dogs
Okay, Ginger! I've had it! You stay out of the garbage! Understand, Ginger? Stay out of the garbage, or else!
what they hear
blah blah GINGER blah blah blah blah blah blah blah blah blah GINGER blah blah blah blah blah blah...

- Fundamental goal: analyze and process human language, broadly, robustly, accurately…

- End systems that we want to build:
  - Ambitious: speech recognition, machine translation, information extraction, dialog interfaces, question answering…
  - Modest: spelling correction, text categorization…

Tampere University

Problem: Ambiguities

- Headlines:
  - Enraged Cow Injures Farmer With Ax
  - Hospitals Are Sued by 7 Foot Doctors
  - Ban on Nude Dancing on Governor’s Desk
  - Iraqi Head Seeks Arms
  - Local HS Dropouts Cut in Half
  - Juvenile Court to Try Shooting Defendant
  - Stolen Painting Found by Tree
  - Kids Make Nutritious Snacks

- Why are these funny?

Language models

- Formal languages, such as the programming languages Java or Python, have precisely defined language models
- A **language** can be defined as a set of strings;
  - "`print(2 + 2)`" is a legal program in the language Python, whereas
  - "`2)+(2 print`" is not
- Infinitely many legal programs: they cannot be enumerated
  - they are specified by a set of rules called a **grammar**
- Formal languages also have rules that define the meaning of **semantics** of a program
  - the rules say that the "meaning" of "`2 + 2`" is 4, and
  - the meaning of "1/0" is that an error is signaled
- Natural languages, such as English or Spanish, cannot be characterized as a definitive set of sentences

Characteristics of natural languages

"Not to be invited is sad"

- is a sentence of English, but what about the grammaticality of

"To be not invited is sad"

- A natural language model = probability distribution over sentences rather than a definitive set
- Rather than asking if a string of *words* is (not) a member of the set defining the language, we ask for $P(S = words)$ — what is the probability that a random sentence would be *words*

Natural languages are also ambiguous

- "He saw her duck" can mean either that he saw a waterfowl belonging to her, or that he saw her move to evade something
- There is no single meaning for a sentence, but rather of a probability distribution over possible meanings

Natural languages are very large

- and hence difficult to deal with, and constantly changing
- Thus, our language models are, at best, an approximation

N-gram character models

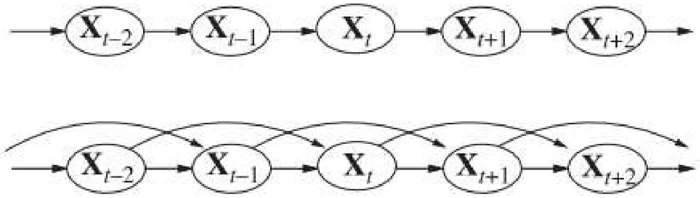
- Ultimately, a written text is composed of **characters** — letters, digits, punctuation, and spaces in English
- One of the simplest language models is a probability distribution over sequences of characters

- We write $P(c_{1:N})$ for the probability of a sequence of N characters, c_1 through c_N
- In one Web collection, *P*(“the”) = 0.027 and *P*(“zgq”) = 0.000000002
- A sequence of written symbols of length n is called an ***n*-gram**

- A model of the probability distribution of n-letter sequences is called an n-gram model
- We can also have n-gram models over sequences of words, syllables, or other units

Tampere University

Markov chain



- A *transition model* specifies a probability distribution over the latest state variables, given the previous values, that is,
$$\mathbf{P}(\mathbf{X}_t | \mathbf{X}_{0:t-1})$$

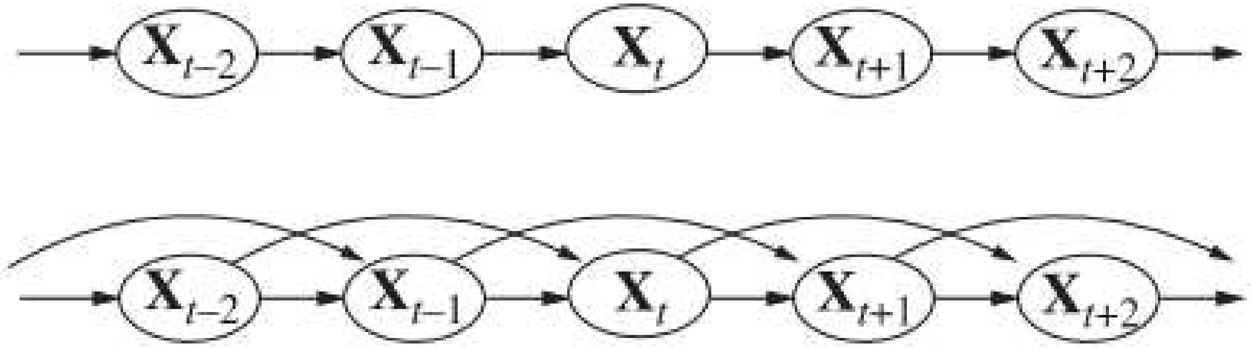
- **Problem:** the set $\mathbf{X}_{0:t-1}$ is unbounded in size as t increases

- Solve by making a **Markov assumption** —the current state depends on only a finite fixed number of previous states

- **Markov processes** or **Markov chains** first studied in depth by the Russian statistician Andrei Markov (1856–1922)

- The simplest model is the **first-order Markov process**, in which the current state depends only on the previous state and not on any earlier states
- I.e., a state provides enough information to make the future conditionally independent of the past, and we have
$$\mathbf{P}(\mathbf{X}_t | \mathbf{X}_{0:t-1}) = \mathbf{P}(\mathbf{X}_t | \mathbf{X}_{t-1})$$

- In a first-order Markov process, the transition model is the conditional distribution $\mathbf{P}(\mathbf{X}_t|\mathbf{X}_{t-1})$
- The transition model for a second-order Markov process is the conditional distribution $\mathbf{P}(\mathbf{X}_t|\mathbf{X}_{t-2}, \mathbf{X}_{t-1})$



- An n-gram model is defined as a **Markov chain** of order $n-1$
- In a Markov chain the probability of character c_i depends only on the immediately preceding characters, not on any other characters
- So in a **trigram** model (Markov chain of order 2) we have

$$P(c_i|c_{1:i-1}) = P(c_i|c_{i-2:i-1})$$

- We define the probability of a sequence of characters $P(c_{1:N})$ under the trigram model by first factoring with the chain rule and then using the Markov assumption:

$$P(c_{1:N}) = \prod_{i=1}^{N} P(c_i|c_{1:i-1}) = \prod_{i=1}^{N} P(c_i|c_{i-2:i-1})$$

- For a trigram model in a language with 100 characters, $P(c_i|c_{i-2:i-1})$ has a million (100^3) entries
- It can be accurately estimated by counting character sequences in a body of text of 10 million characters or more
- A body of text is called **corpus**, from the Latin word for body

Model evaluation

- There are many possible n-gram models—unigram, bigram, trigram, interpolated smoothing with different values of λ
- We can evaluate a model with **cross-validation**
  - Split the corpus into a training corpus and a validation corpus
  - Determine the parameters of the model from the training data
  - Then evaluate the model on the validation corpus
- The evaluation can be a **task-specific metric**
  - E.g., measuring accuracy on language identification
- Alternatively we can have a **task-independent model** of language quality:
  - calculate the probability assigned to the validation corpus by the model;
  - the higher the probability the better
- The probability of a large corpus will be a very small number, and floating-point underflow becomes an issue

Tampere University

N-gram word models

- In n-gram models over words all the same mechanisms apply as in character models
- The **vocabulary** is larger

- There are only about 100 characters in most languages, and sometimes we build character models that are even more restrictive, e.g.,
  - by treating “A” and “a” as the same symbol or
  - by treating all punctuation as the same symbol

- With word models we have tens of thousands of symbols, and sometimes millions
- It is not clear what constitutes a word
- English: a sequence of letters surrounded by spaces is a word
- Chinese: words are not separated by spaces
- Even in English many decisions must be made to have a clear policy on word boundaries: how many words are in

“ne’er-do-well”?

Or in “(Tel:1-800-960-5660x123)”?

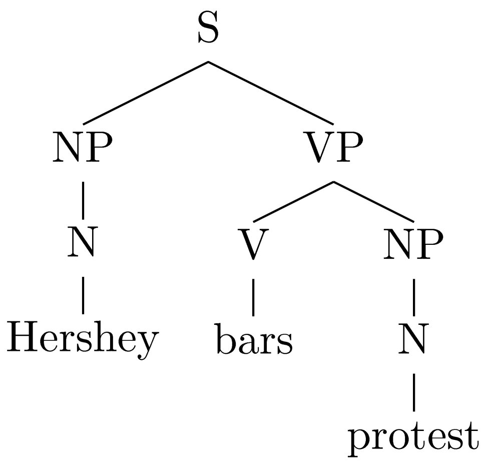
Unknown words

- Word n-gram models need to deal with **out of vocabulary words**
- Nobody is inventing a new letter of the alphabet
- With word models there is always the chance of a new word that was not seen in the training corpus
- Add just one new word to the vocabulary: $\langle\mathrm{UNK}\rangle$, standing for the unknown word

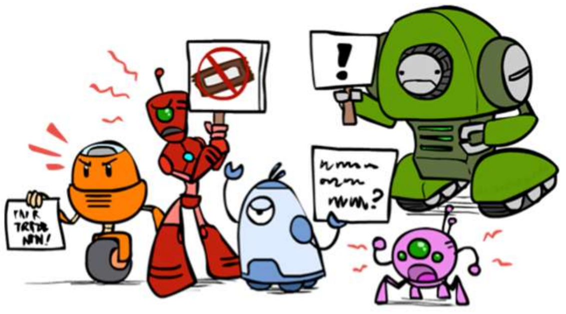
- We can estimate n-gram counts for $\langle\mathrm{UNK}\rangle$: go through the training corpus, and the first time any individual word appears it is previously unknown, so replace it with $\langle\mathrm{UNK}\rangle$
- All subsequent appearances of the word remain unchanged
- Then compute n-gram counts as usual, treating $\langle\mathrm{UNK}\rangle$ just like any other word
- When an unknown word appears in a test set, we look up its probability under $\langle\mathrm{UNK}\rangle$
- Sometimes multiple unknown-word symbols for different classes:
  - string of digits might be replaced with $\langle\mathrm{NUM}\rangle$,
  - email address with $\langle\mathrm{EMAIL}\rangle$

- Build models over the words in the AIMA textbook and then randomly sample sequences of words from the them
  - **Unigram**: *logical are as are confusion a may right tries agent goal the was…*
  - **Bigram**: *systems are very similar computational approach would be represented…*
  - **Trigram**: *planning and scheduling are integrated the success of naive Bayes model…*
- The unigram model (perplexity 891) is a poor approximation of either English or the content of an AI textbook
- The bigram (perplexity 142) and trigram (perplexity 91) models are much better

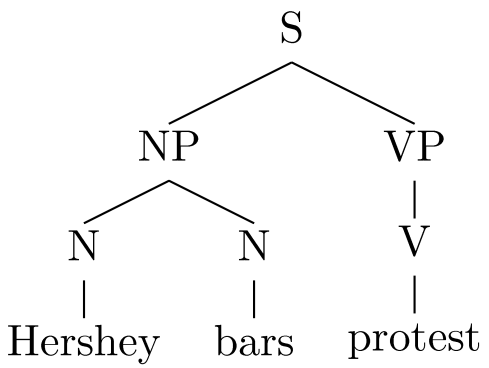
Parsing as Search



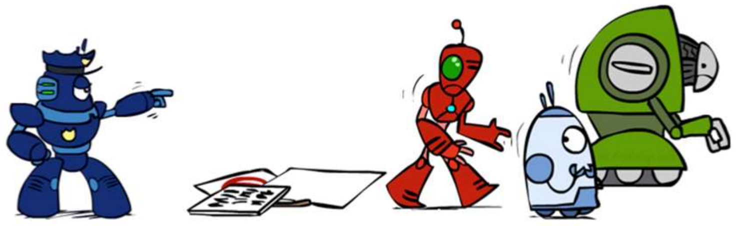
S
NP
VP
N
V
NP
Hershey
bars
N
protest



Hershey bars protest



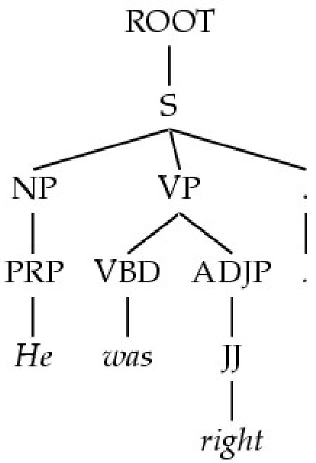
S
NP
VP
N
N
V
Hershey
bars
protest



Sentence
Noun Phrase
Verb Phrase
Noun
Verb

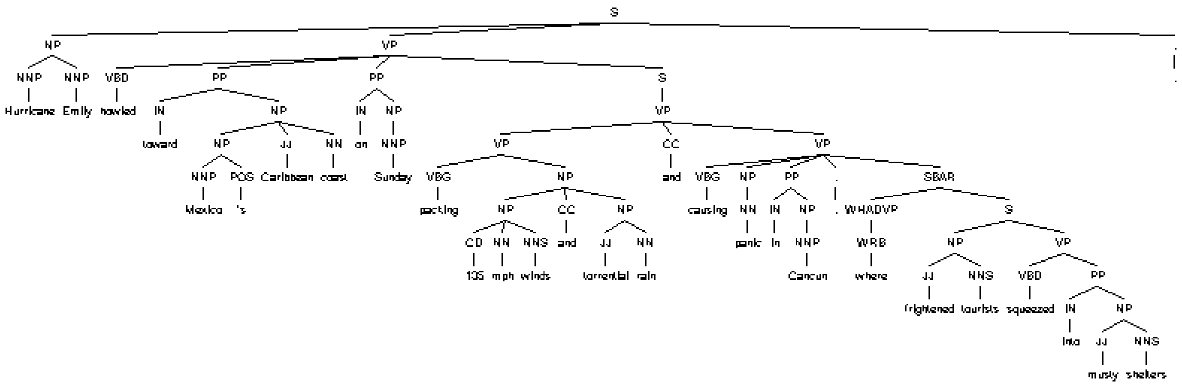
Grammar: PCFGs

- Natural language grammars are very ambiguous!
- PCFGs are a formal probabilistic model of trees
  - Each “rule” has a conditional probability (like an HMM)
  - Tree’s probability is the product of all rules used
- Parsing: Given a sentence, find the best tree – search!



ROOT → S	375/420
S → NP VP .	320/392
NP → PRP	127/539
VP → VBD ADJP	32/401
…..	

Syntactic Analysis

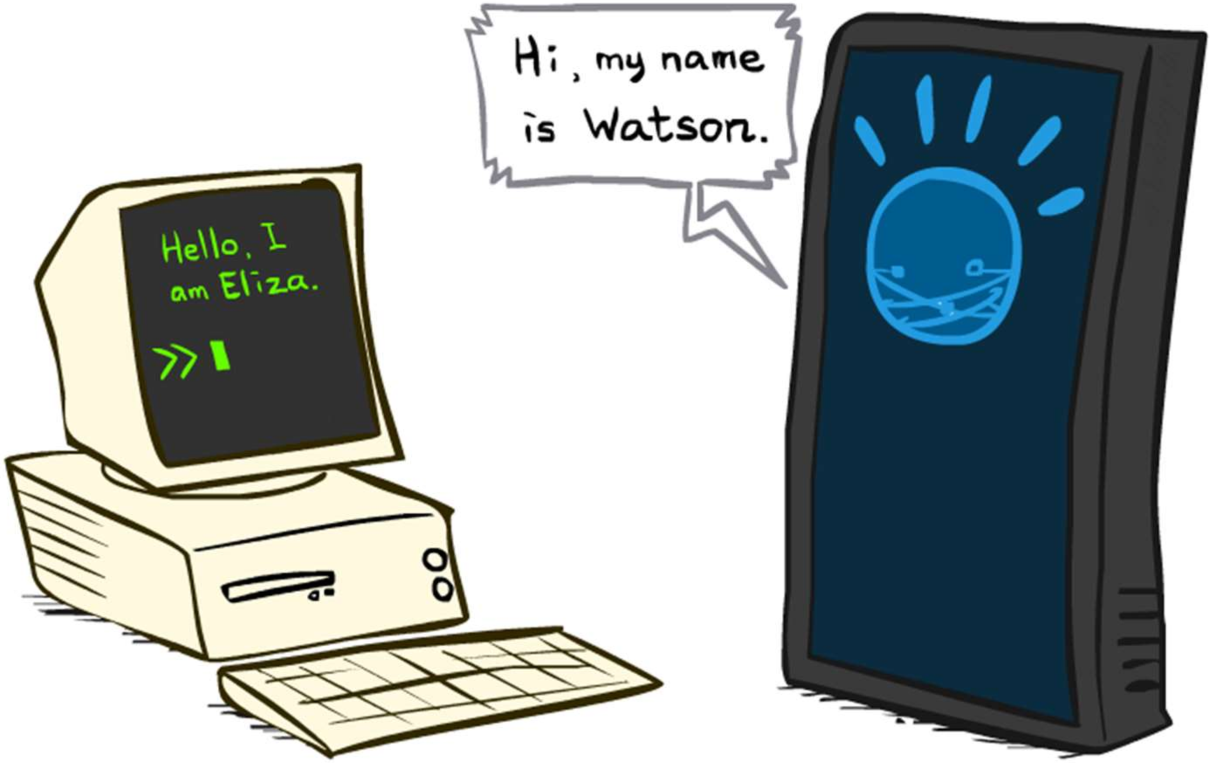

S
NP VP .
NNP NNP VBD PP PP S
Hurricane Emily howled IN NP IN NP VP
toward NP JJ NN on NNP VP CC VP
NNP POS Caribbean coast Sunday VBG NP and VBG NP PP , SBAR
Mexico 's packing NP CC NP causing NN IN NP WHADVP S
CD NN NNS and JJ NN panic in NNP WRB NP VP
135 mph winds torrential rain Cancun where JJ NNS VBD PP
frightened tourists squeezed IN NP
into JJ NNS
musty shelters

Hurricane Emily howled toward Mexico 's Caribbean coast on Sunday packing 135 mph winds and torrential rain and causing panic in Cancun, where frightened tourists squeezed into musty shelters.

http://nlp.cs.berkeley.edu

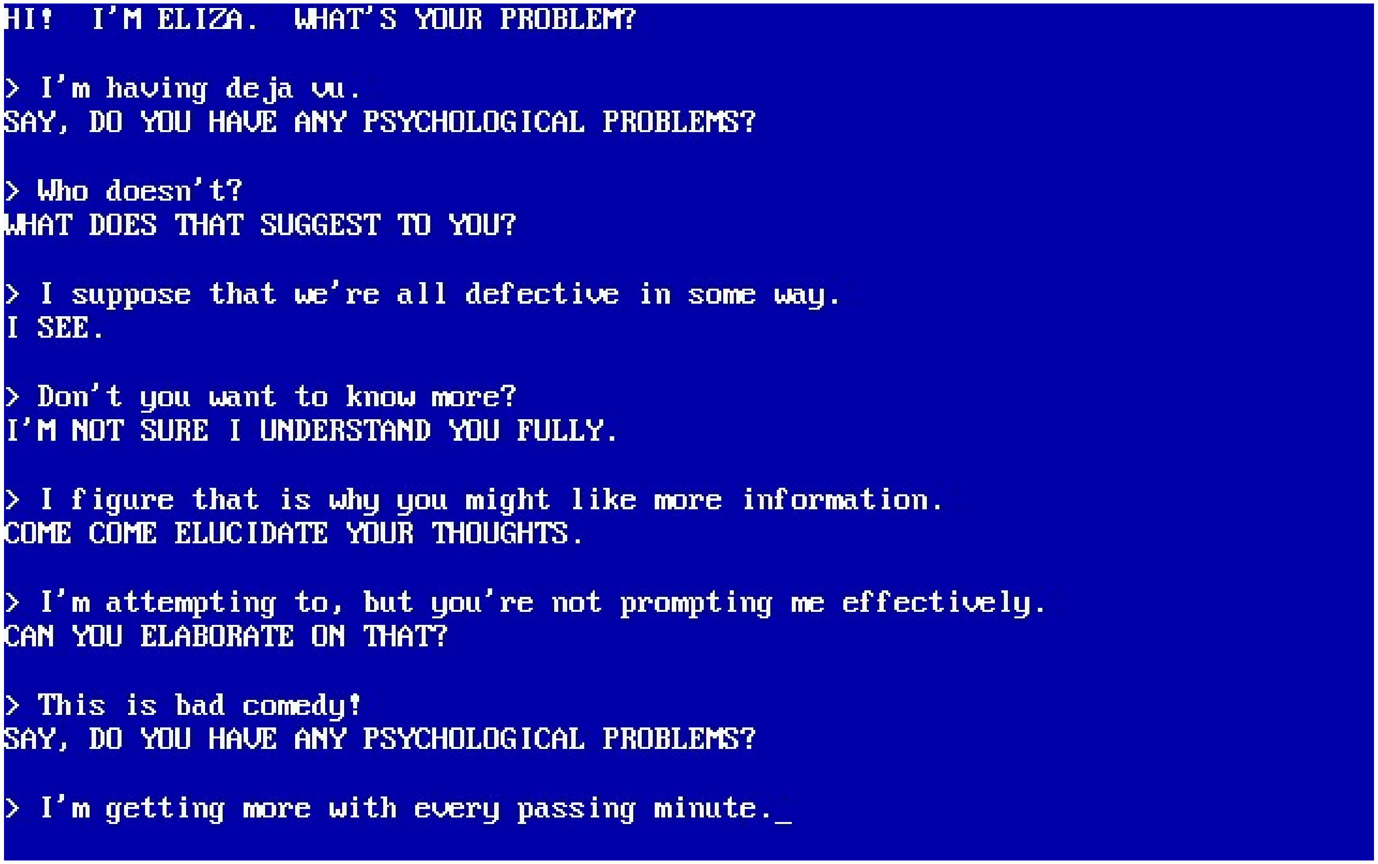
Tampere University

Dialog Systems



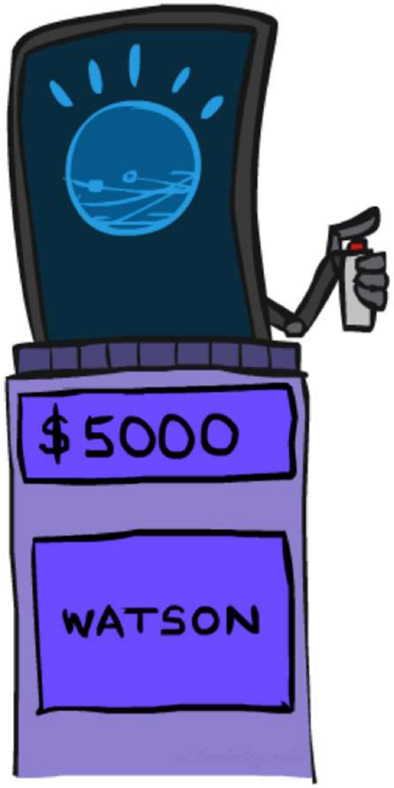
ELIZA

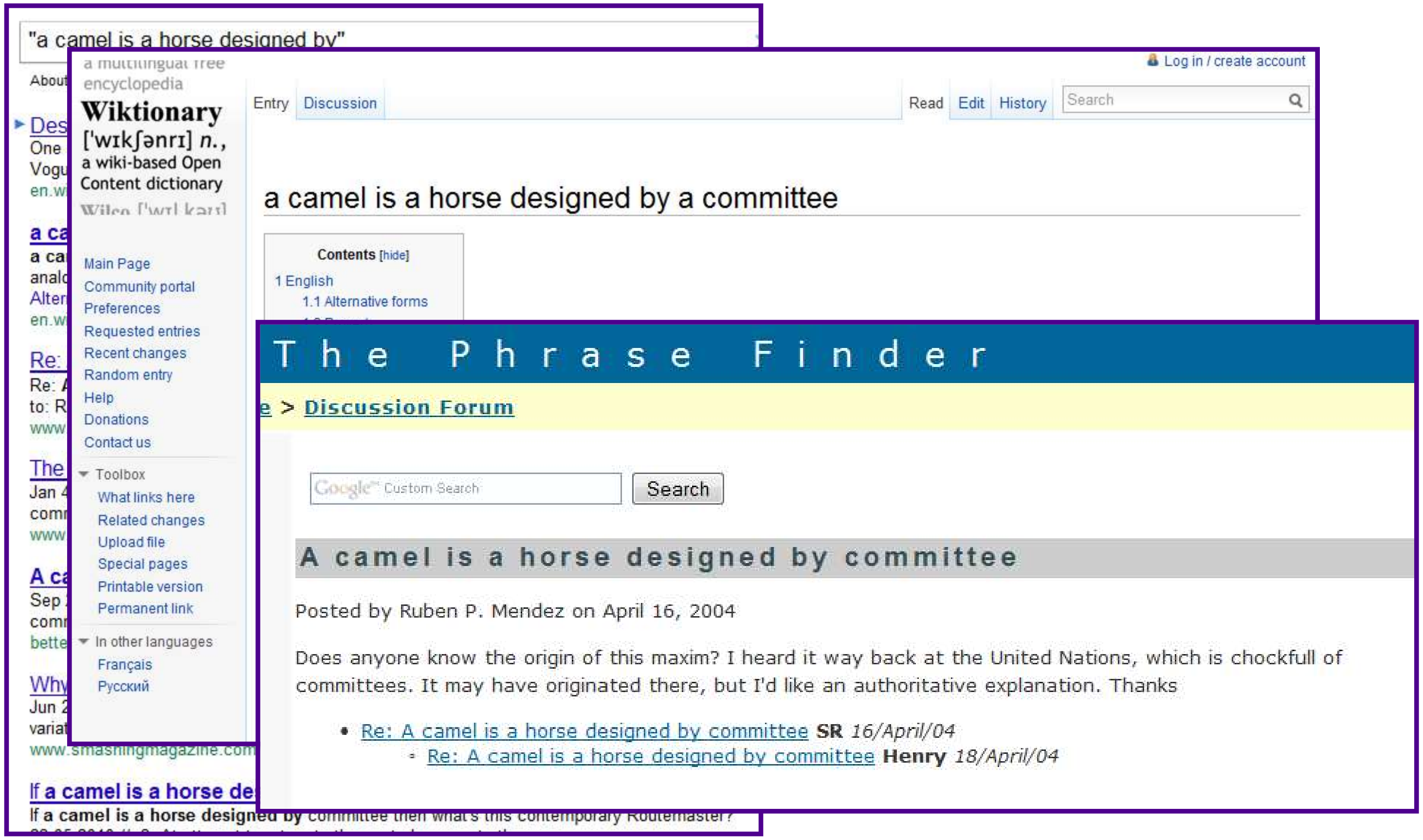
- A “psychotherapist” agent (Weizenbaum, ~1964)
- Led to a long line of chatterbots
- How does it work:
  - Trivial NLP: string match and substitution
  - Trivial knowledge: tiny script / response database
  - Example: matching “I remember __” results in “Do you often think of __”?
- Can fool some people some of the time?



```
HI!  I'M ELIZA.  WHAT'S YOUR PROBLEM?

> I'm having deja vu.
SAY, DO YOU HAVE ANY PSYCHOLOGICAL PROBLEMS?

> Who doesn't?
WHAT DOES THAT SUGGEST TO YOU?

> I suppose that we're all defective in some way.
I SEE.

> Don't you want to know more?
I'M NOT SURE I UNDERSTAND YOU FULLY.

> I figure that is why you might like more information.
COME COME ELUCIDATE YOUR THOUGHTS.

> I'm attempting to, but you're not prompting me effectively.
CAN YOU ELABORATE ON THAT?

> This is bad comedy!
SAY, DO YOU HAVE ANY PSYCHOLOGICAL PROBLEMS?

> I'm getting more with every passing minute._
```

Watson





Wiktionary ['wɪkʃənrɪ] *n.*, a wiki-based Open Content dictionary

a camel is a horse designed by a committee

If a camel is a horse de

If a camel is a horse designed by committee then what's this contemporary Routemaster?

The Phrase Finder

A camel is a horse designed by committee

Posted by Ruben P. Mendez on April 16, 2004

Does anyone know the origin of this maxim? I heard it way back at the United Nations, which is chockfull of committees. It may have originated there, but I'd like an authoritative explanation. Thanks

- Re: A camel is a horse designed by committee **SR** *16/April/04*
  - Re: A camel is a horse designed by committee **Henry** *18/April/04*

What’s in Watson?

- A question-answering system (IBM, 2011)
- Designed for the game of Jeopardy
- How does it work:
  - Sophisticated NLP: deep analysis of questions, noisy matching of questions to potential answers
  - Lots of data: onboard storage contains a huge collection of documents (e.g. Wikipedia, etc.), exploits redundancy
  - Lots of computation: 90+ servers
- Can beat all of the people all of the time?

Tampere University

Language identification

- n-gram character models are well suited for language identification: given a text, determine what natural language it is written in
- This is a relatively easy task; even with short texts such as "*Hello, world*" or "*Wie geht es dir*," it is easy to identify the language
- Computer systems identify languages with greater than 99% accuracy; closely related languages, e.g. Swedish and Norwegian, can get confused
- One approach is to first build a trigram character model of each candidate language, $P(c_i|c_{i-2:i-1}, \ell)$, where the variable ℓ ranges over languages
- For each ℓ the model is built by counting trigrams in a corpus of that language
- About 100,000 characters of each language are needed
- That gives us a model of $\mathbf{P}(Text\,|\,Language)$

- To select the most probable language ℓ^* given the text $c_{1:N}$ we apply Bayes' rule followed by the Markov assumption to get the most probable language:

$$\begin{aligned} \ell^* &= \underset{\ell}{\operatorname{argmax}} P(\ell | c_{1:N}) \\ &= \underset{\ell}{\operatorname{argmax}} P(\ell) P(c_{1:N} | \ell) \\ &= \underset{\ell}{\operatorname{argmax}} P(\ell) \prod_{i=1}^{N} P(c_i | c_{i-2:i-1}, \ell) \end{aligned}$$

- The trigram model learned from a corpus, what about the prior probability $P(\ell)$?
- If we are selecting a random Web page we know that English is the most likely language and that the probability of Macedonian will be less than 1%
- The exact number we select for these priors is not critical because the trigram model usually selects one language that is several orders of magnitude more probable than any other

Other tasks for character models include

1. *Spelling correction*
2. *Genre classification*
   - deciding if a text is a news story, a legal document, a scientific article, etc.
   - While many features help make this classification, counts of punctuation and other character n-gram features go a long way
3. *Named-entity recognition*
   - finding names of things in a document and deciding what class they belong to
   - E.g., in the text "*Mr. Sopersteen was prescribed aciphex*," we should recognize that "*Mr. Sopersteen*" is the name of a person and "*aciphex*" is the name of a drug
   - Character-level models are good for this task because they can associate the character sequence
     - "*ex*" followed by a space with a drug name and
     - "*steen*" with a person name, and

     thereby identify words that they have never seen before

Smoothing n-gram models

- The training corpus provides only an estimate of the true probability distribution
- For common character sequences such as "*th*" any English corpus will give a good estimate: about 1.5% of all trigrams
- On the other hand, "*ht*" is very uncommon—no dictionary words start with ht
- It is likely that the sequence would have a count of zero in a training corpus of standard English
- Should we assign P("*ht*") = 0?
- If we did, then the text "*The program issues an http request*" would have an English probability of zero
- This seems wrong

- We want our language models to generalize well to texts they haven’t seen yet
- Just because we have never seen “*http*” before does not mean that our model should claim that it is impossible
- Adjust the language model so that sequences that have a count of zero in the training corpus will be assigned a small nonzero probability
- The other counts will be adjusted downward slightly so that the probability still sums to 1
- The process of adjusting the probability of low-frequency counts is called **smoothing**

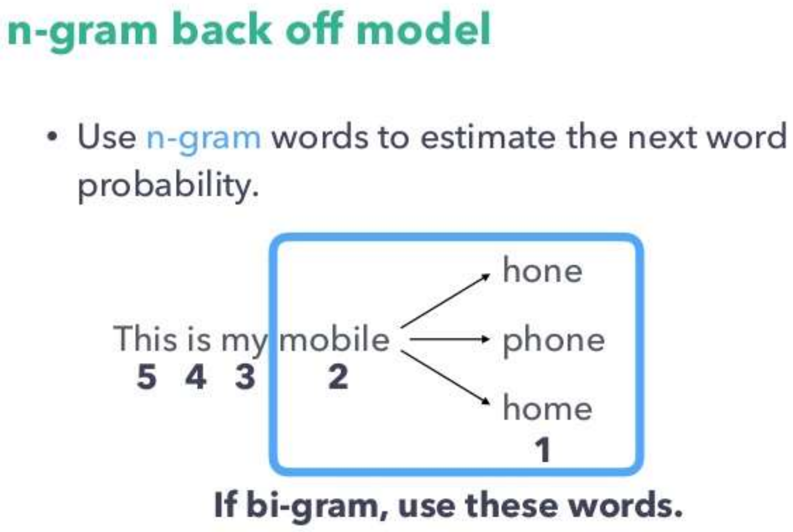
Laplace Smoothing

- Laplace (18th century):
  - In the lack of further information, if a random Boolean variable X has been **false** in all n observations so far then the estimate for
$$P(X = true)$$
    should be $1/(n+2)$

- He assumes that with two more trials, one might be **true** and one **false**
- Laplace smoothing performs relatively poorly

Tampere University

Backoff Model



n-gram back off model

- Use n-gram words to estimate the next word probability.

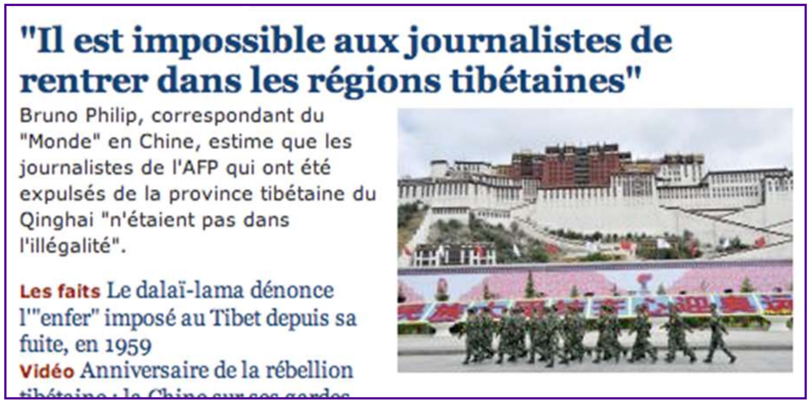
If bi-gram, use these words.

- A better approach is a **backoff** model
  - start by estimating n-gram counts
  - for any sequence that has a low or zero count, we back off to $(n-1)$-grams

- **Linear interpolation smoothing** is a backoff model that combines trigram, bigram, and unigram models by linear interpolation
$$\hat{P}(c_i|c_{i-2:i-1})$$
$$= \lambda_3 P(c_i|c_{i-2:i-1}) + \lambda_2 P(c_i|c_{i-1}) + \lambda_1 P(c_i),$$
- where $\lambda_1 + \lambda_2 + \lambda_3 = 1$

Machine Translation

Machine Translation

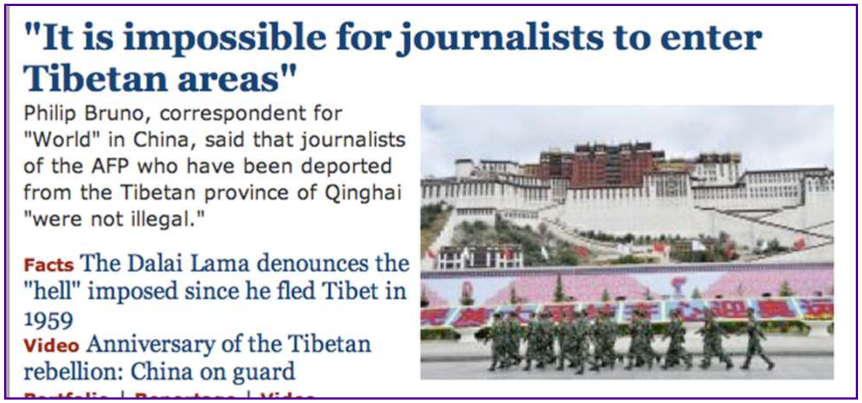


"Il est impossible aux journalistes de rentrer dans les régions tibétaines"

Bruno Philip, correspondant du "Monde" en Chine, estime que les journalistes de l'AFP qui ont été expulsés de la province tibétaine du Qinghai "n'étaient pas dans l'illégalité".

Les faits Le dalaï-lama dénonce l'"enfer" imposé au Tibet depuis sa fuite, en 1959

Vidéo Anniversaire de la rébellion tibétaine : la Chine sur ses gardes



"It is impossible for journalists to enter Tibetan areas"

Philip Bruno, correspondent for "World" in China, said that journalists of the AFP who have been deported from the Tibetan province of Qinghai "were not illegal."

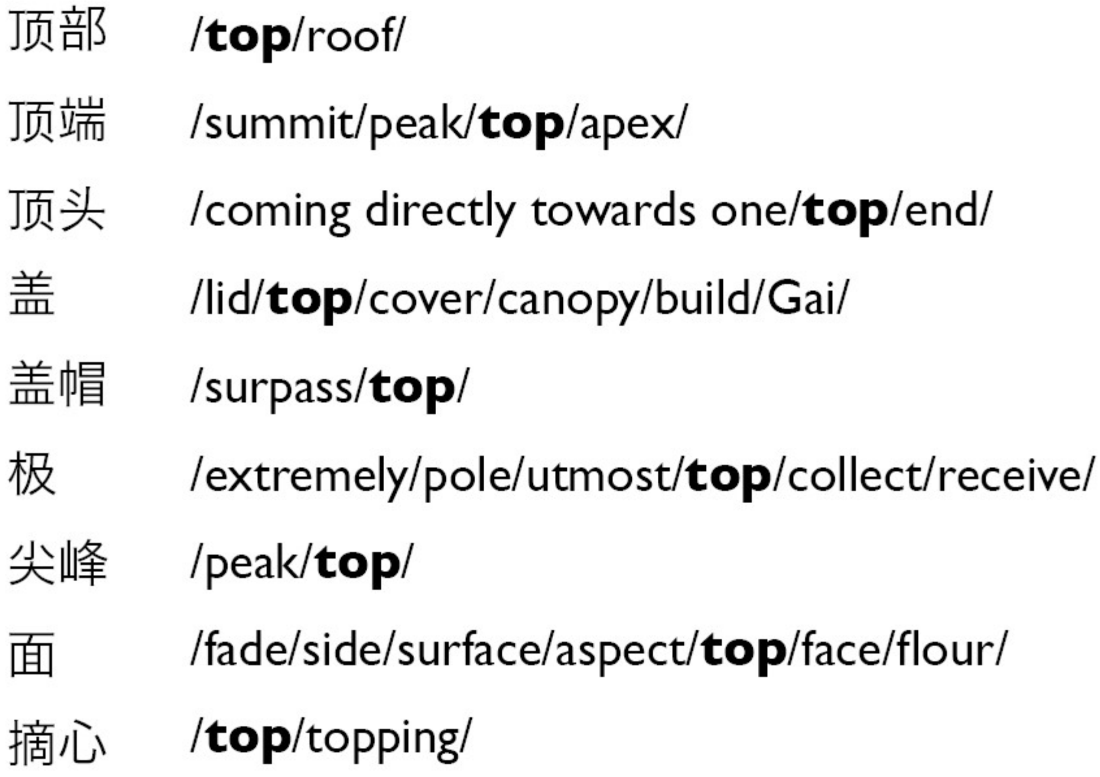
Facts The Dalai Lama denounces the "hell" imposed since he fled Tibet in 1959

Video Anniversary of the Tibetan rebellion: China on guard

Portfolio | Reportage | Video

- Translate text from one language to another
- Recombines fragments of example translations
- Challenges:
  - What fragments? [learning to translate]
  - How to make efficient? [fast translation search]

The Problem with Dictionary Lookups



顶部	/**top**/roof/
顶端	/summit/peak/**top**/apex/
顶头	/coming directly towards one/**top**/end/
盖	/lid/**top**/cover/canopy/build/Gai/
盖帽	/surpass/**top**/
极	/extremely/pole/utmost/**top**/collect/receive/
尖峰	/peak/**top**/
面	/fade/side/surface/aspect/**top**/face/flour/
摘心	/**top**/topping/

Example from Douglas Hofstadter

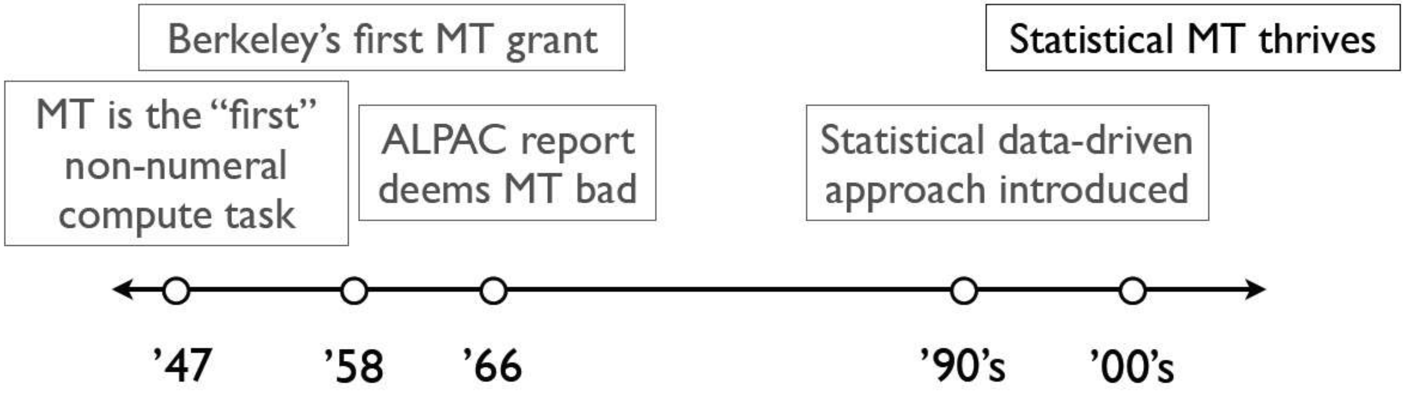
MT: 60 Years in 60 Seconds

Warren Weaver

When I look at an article in Russian, I say: "This is really written in English, but it has been coded in some strange symbols. I will now proceed to decode."

John Pierce

"Machine Translation" presumably means going by algorithm from machine-readable source text to useful target text... In this context, there has been no machine translation...



MT is the "first" non-numeral compute task

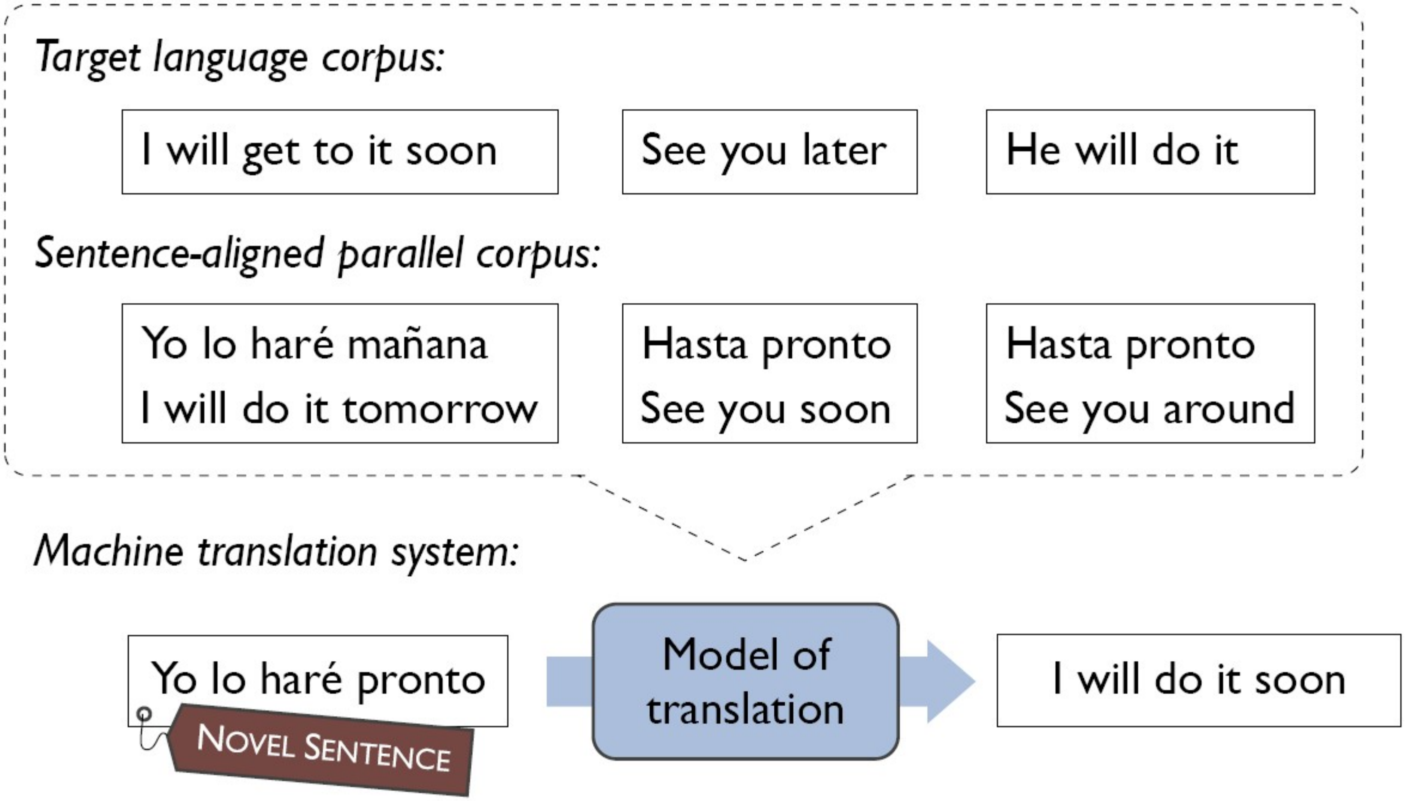
Berkeley's first MT grant

ALPAC report deems MT bad

Statistical data-driven approach introduced

Statistical MT thrives

Data-Driven Machine Translation



Target language corpus:

I will get to it soon

See you later

He will do it

Sentence-aligned parallel corpus:

Yo lo haré mañana
I will do it tomorrow

Hasta pronto
See you soon

Hasta pronto
See you around

Machine translation system:

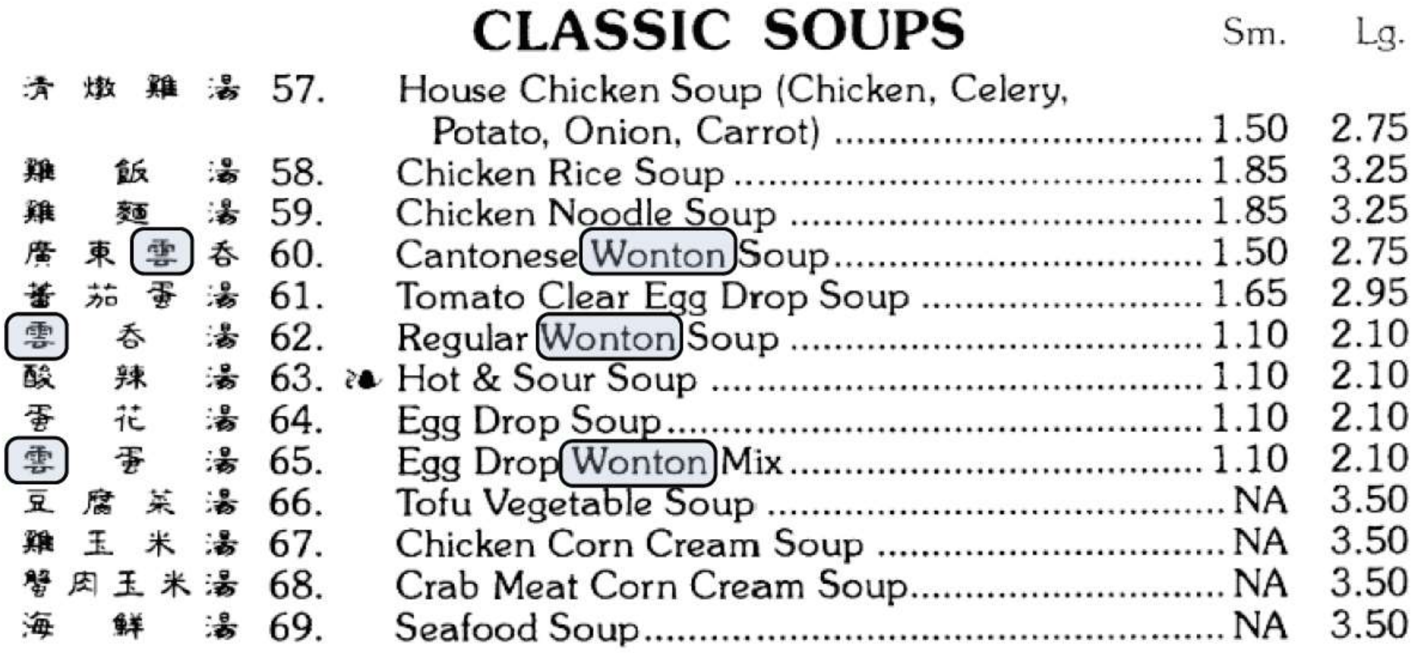
Yo lo haré pronto

NOVEL SENTENCE

Model of translation

I will do it soon

Learning to Translate

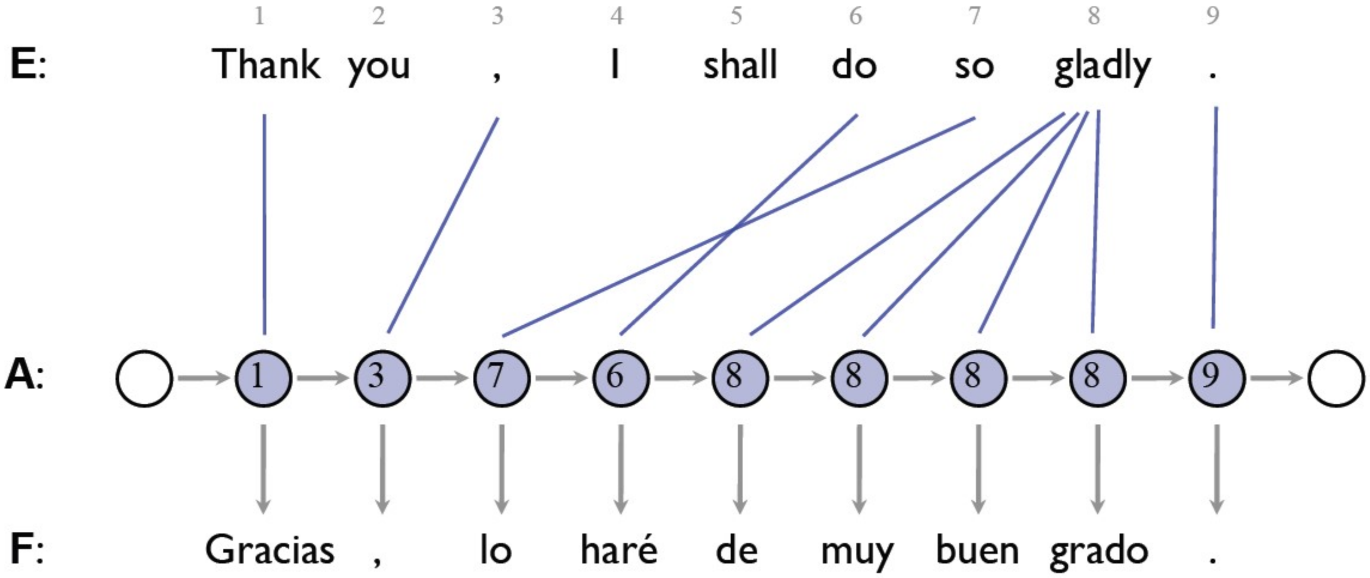


CLASSIC SOUPS

			Sm.	Lg.
清燉雞湯	57.	House Chicken Soup (Chicken, Celery, Potato, Onion, Carrot)	1.50	2.75
雞飯湯	58.	Chicken Rice Soup	1.85	3.25
雞麵湯	59.	Chicken Noodle Soup	1.85	3.25
廣東雲吞	60.	Cantonese Wonton Soup	1.50	2.75
蕃茄蛋湯	61.	Tomato Clear Egg Drop Soup	1.65	2.95
雲吞湯	62.	Regular Wonton Soup	1.10	2.10
酸辣湯	63.	🌶 Hot & Sour Soup	1.10	2.10
蛋花湯	64.	Egg Drop Soup	1.10	2.10
雲蛋湯	65.	Egg Drop Wonton Mix	1.10	2.10
豆腐菜湯	66.	Tofu Vegetable Soup	NA	3.50
雞玉米湯	67.	Chicken Corn Cream Soup	NA	3.50
蟹肉玉米湯	68.	Crab Meat Corn Cream Soup	NA	3.50
海鮮湯	69.	Seafood Soup	NA	3.50

Example from Adam Lopez

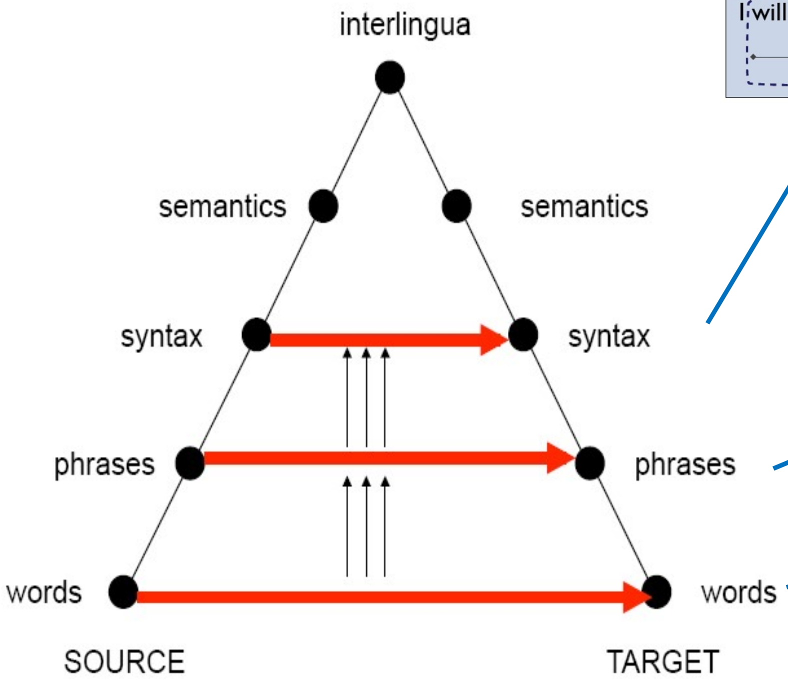
An HMM Translation Model

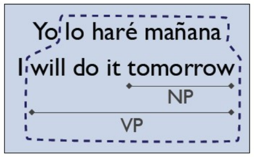


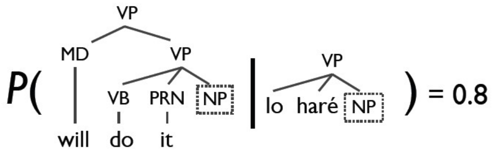
Model Parameters

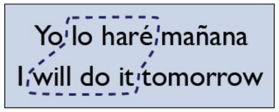
Emissions: $P(F_1 = \text{Gracias} \mid E_{A_1} = \text{Thank})$ *Transitions*: $P(A_2 = 3 \mid A_1 = 1)$

Levels of Transfer

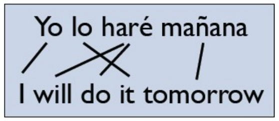






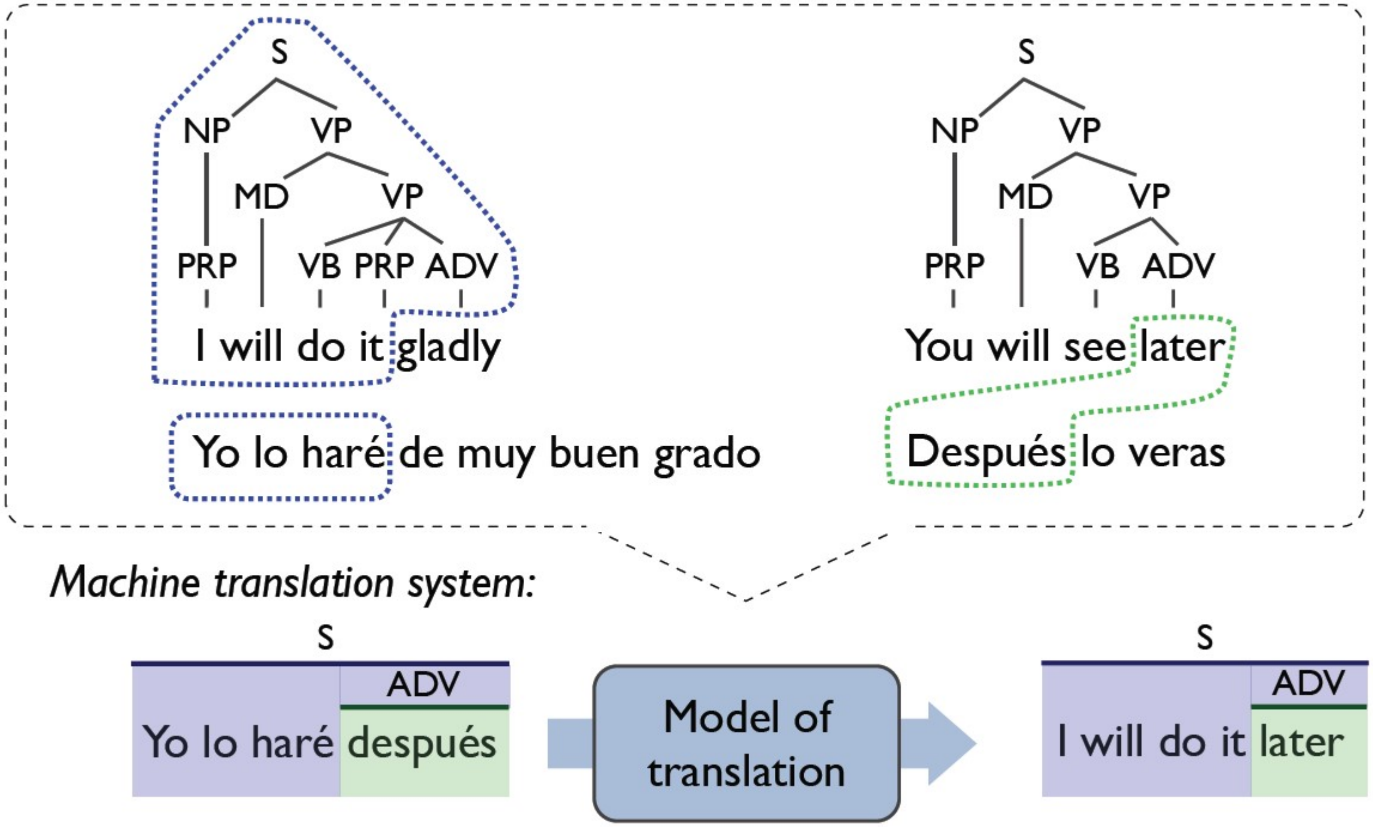


English (E)	*P*(E | lo haré)
will do it	0.8
will do so	0.2



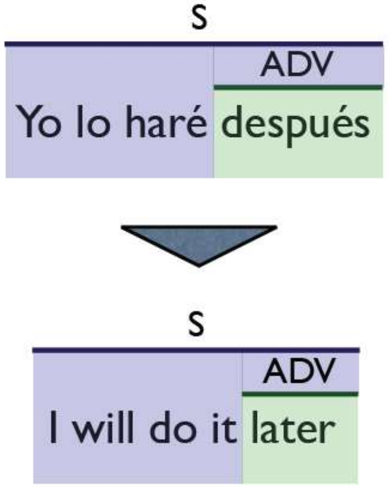
English (E)	*P*(E | mañana)
tomorrow	0.7
morning	0.3

Syntactic Translation



A Syntactic MT System

Synchronous Derivation



Synchronous Grammar Rules

S → ⟨ Yo lo haré ADV ; I will do it ADV ⟩

ADV → ⟨ después ; later ⟩

A Statistical Model

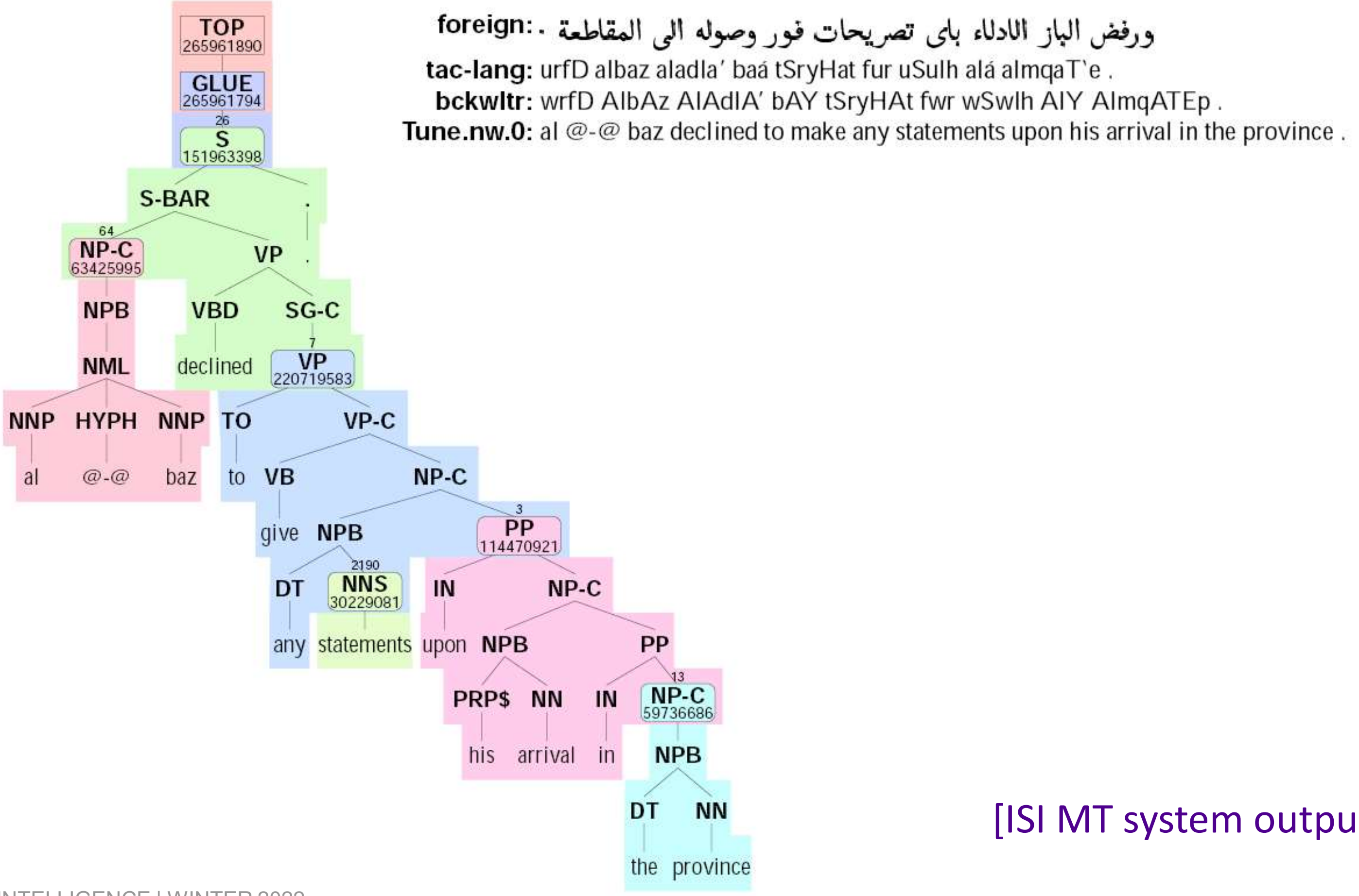
Translation model components factor over applied rules

How well are these rules supported by the data?

Language model factors over n-grams

How well is this output sentence supported by the data?

Example: Syntactic MT Output



foreign: ورفض الباز الادلاء باى تصريحات فور وصوله الى المقاطعة .

tac-lang: urfD albaz aladla’ baá tSryHat fur uSulh alá almqaT`e .

bckwltr: wrfD AlbAz AlAdlA’ bAY tSryHAt fwr wSwlh AlY AlmqATEp .

Tune.nw.0: al @-@ baz declined to make any statements upon his arrival in the province .

[ISI MT system output]