

CSC4504 : Formal Languages & Applications

J Paul Gibson, D311

`paul.gibson@telecom-sudparis-edu.eu`

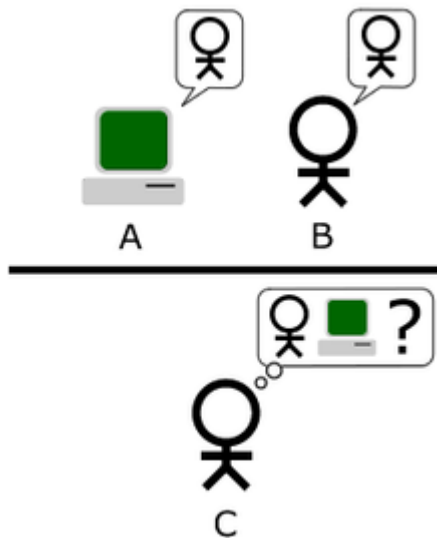
`/~gibson/Teaching/CSC4504/`

Natural Language Processing

`/~gibson/Teaching/CSC4504/Problem7-NLP.pdf`

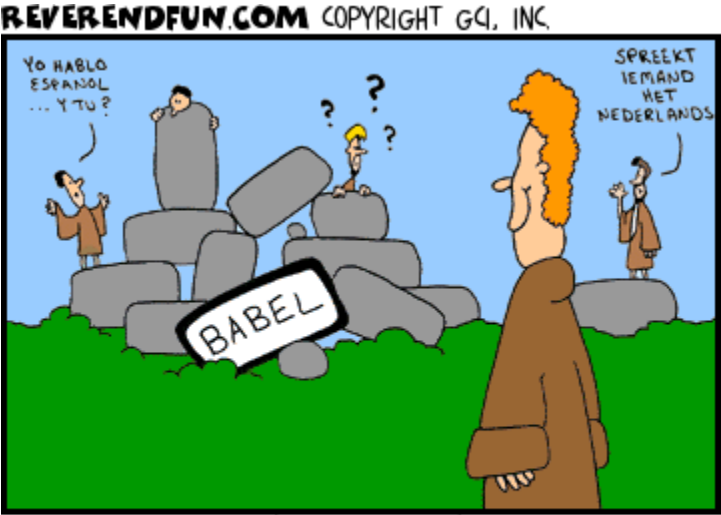


In 1950, Alan Turing "Computing Machinery and Intelligence" proposed a (Turing) test for *intelligence* : can a computer program/machine impersonate a human in a real-time (written) conversation sufficiently well that a typical human is unable to distinguish — through analysis of the conversation alone — between the program and a real human





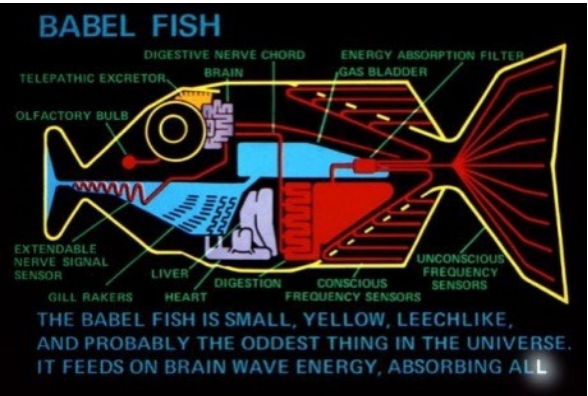
Unifying Language Understanding is integrated in myth and culture : but how it is done is magic rather than technology (until more modern times)



Thanks to Nick Hengeveld (See Genesis 11:1-9) 02-15-2000
SUDDENLY...RUPERT REALIZED THAT ALL THAT TIME SPENT EARNING A LANGUAGE DEGREE WAS FINALLY ABOUT TO PAY OFF



The Tower of Babel by Pieter Bruegel the Elder (1563)



The Hitchhiker's Guide to the Galaxy.

1950s - Machine Translation Experiments started and funding was significant (in the USA) for 10+ years

The Automatic Language Processing Advisory Committee (ALPAC 1966) ended the substantial funding of MT research (in the United States) for 20+ years following.

Their **report** appeared to suggest that there was little need for more research!

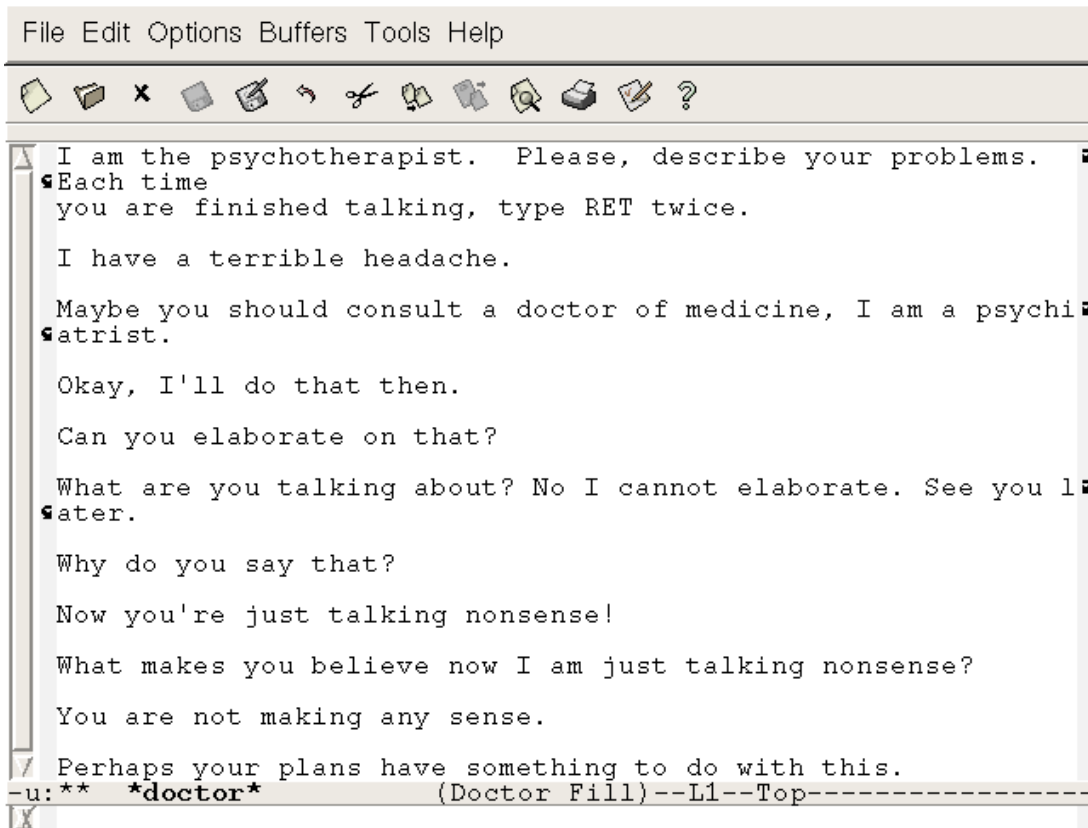
John R. Pierce and John B. Carroll. 1966. *Language and Machines: Computers in Translation and Linguistics*. National Academy of Sciences, Washington, DC, USA.

“...how much should be spent on research and development toward improving translation? It would be unreasonable to spend extravagantly on a relatively small business that is doing the job satisfactorily.”

But, some progress continued to be made in different areas of **Natural Language Processing** after the report was published and research funding reduced!

Early NLP ‘successes’ : ELIZA(1964-)

Joseph Weizenbaum. 1966. ELIZA—a computer program for the study of natural language communication between man and machine. *Commun. ACM* 9, 1 (January 1966), 36-45. DOI=10.1145/365153.365168



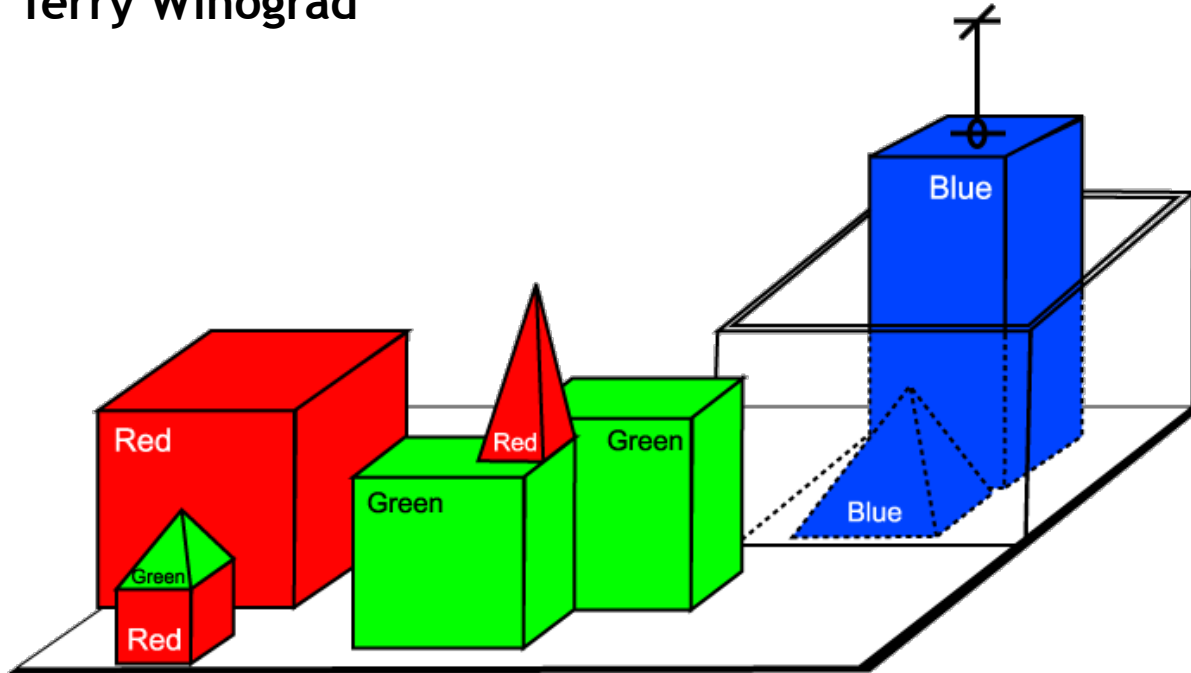
```
File Edit Options Buffers Tools Help
I am the psychotherapist. Please, describe your problems.
Each time
you are finished talking, type RET twice.
I have a terrible headache.
Maybe you should consult a doctor of medicine, I am a psychi
atrist.
Okay, I'll do that then.
Can you elaborate on that?
What are you talking about? No I cannot elaborate. See you l
ater.
Why do you say that?
Now you're just talking nonsense!
What makes you believe now I am just talking nonsense?
You are not making any sense.
Perhaps your plans have something to do with this.
-u:** *doctor* (Doctor Fill)--L1--Top-----
```

This led to many other **chatterbots** being developed in the 1970s.

Some were based on **conceptual ontologies** - structuring real world data in a computer-understandable format

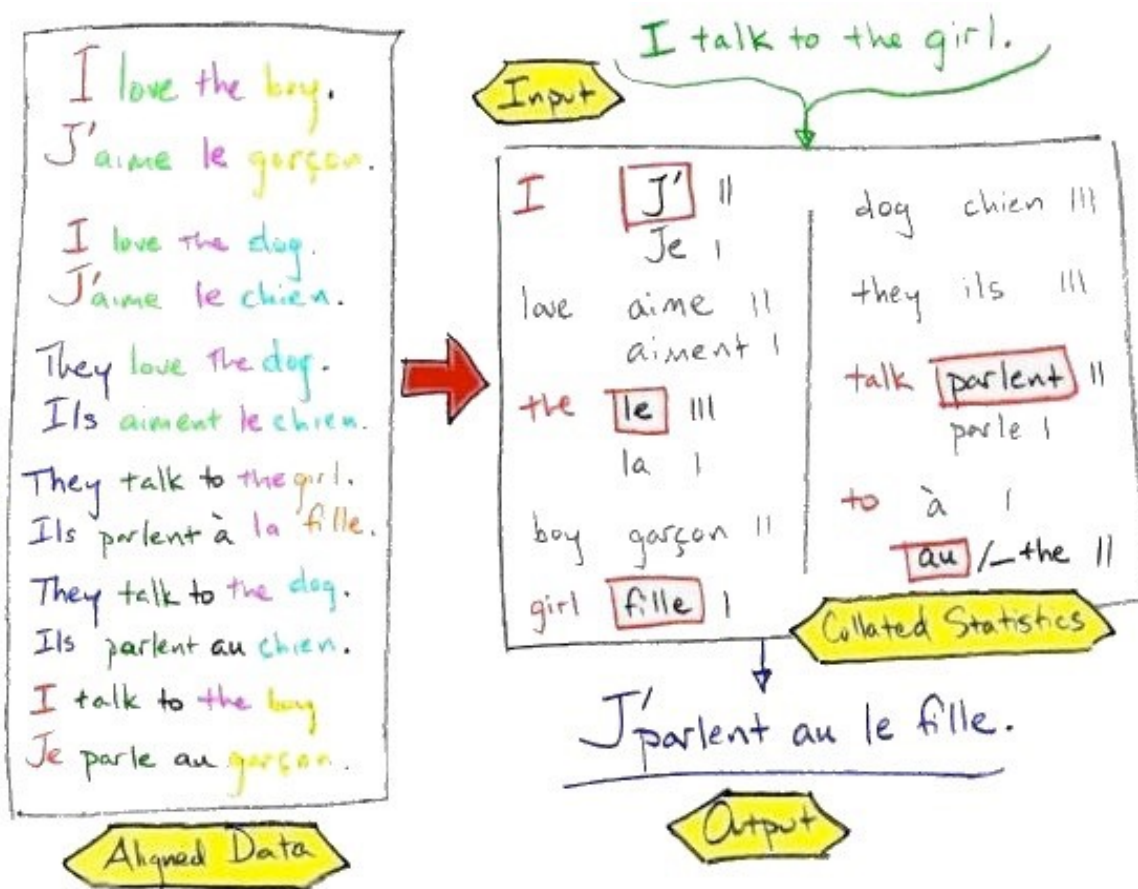
Early NLP 'successes' : SHRDLU (1968-)

Procedures as a Representation for Data in a Computer Program for Understanding Natural Language, PhD Thesis MIT TR235, (February 1971) by Terry Winograd



In the 1970S NLP systems - such as SHRDLU - were based on complex sets of hand-written rules

Machine Translation: machine learning/statistical methods were first proposed in the 1980s



These methods were built upon information theory and probabilistic language models:

- Word-based
- Phrase-based
- Syntax-Based
- Hierarchical

Take advantage of multilingual textual corpora

A subfield of NLP is devoted to learning approaches - Natural Language Learning (NLL)

NLP For Software Engineers - Requirements Engineering

Ryan, Kevin. "The role of natural language in requirements engineering." *Requirements Engineering, 1993., Proceedings of IEEE International Symposium on.* IEEE, 1993.

Ambriola, Vincenzo, and Vincenzo Gervasi. "Processing natural language requirements." *Automated Formal Languages, 1997. Proceedings., 12th IEEE International Conference.* IEEE, 1997.



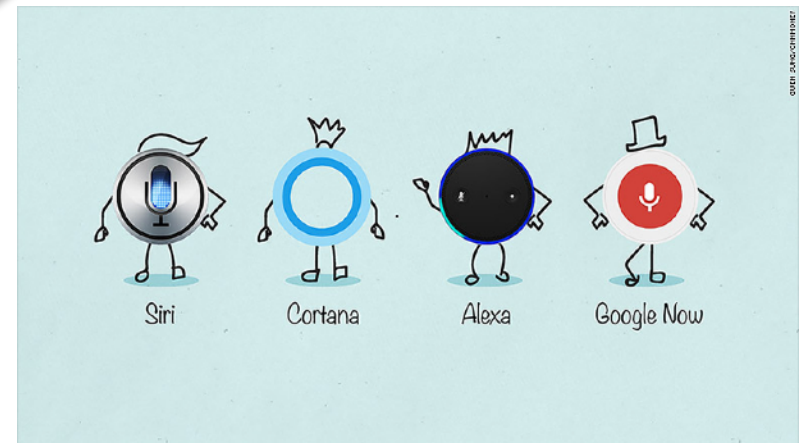
"I think you misunderstood me when I said I wanted our factory to go all green."

NLP For Software Engineers - HCI Design

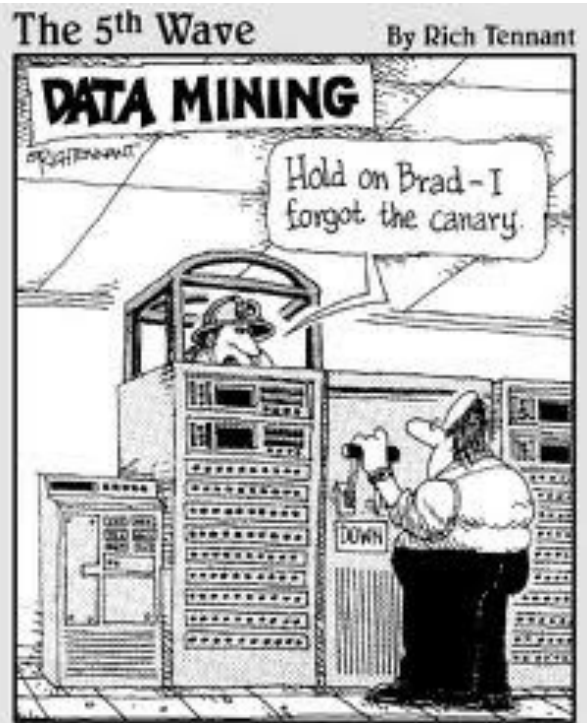
Allen, James F., et al. "Toward conversational human-computer interaction." *AI magazine* 22.4 (2001): 27.



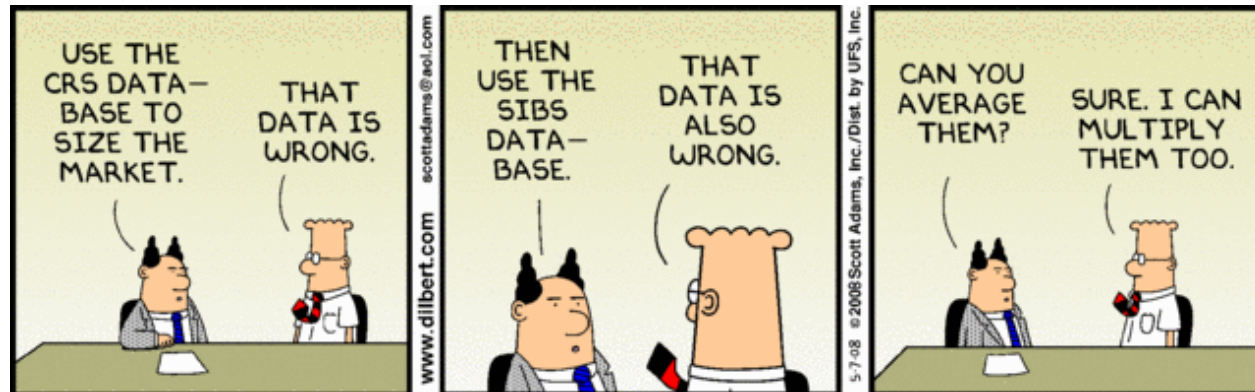
15 years later



NLP For Software Engineers - Text Mining



Sebastiani, Fabrizio. "Machine learning in automated text categorization." *ACM computing surveys (CSUR)* 34.1 (2002): 1-47.

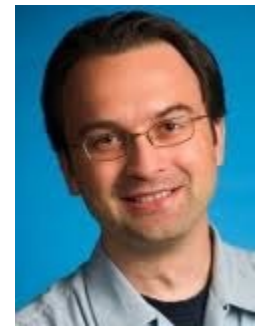


Formal Languages for NLP

Jochen L. Leidner. 2003. Current issues in Formal Languages for Natural Language Processing. In *Proceedings of the HLT-NAACL 2003 workshop on Formal Languages and architecture of language technology systems - Volume 8 (SEALTS '03)*, Vol. 8. Association for Computational Linguistics, Stroudsburg, PA, USA, 45-50. DOI=10.3115/1119226.1119233



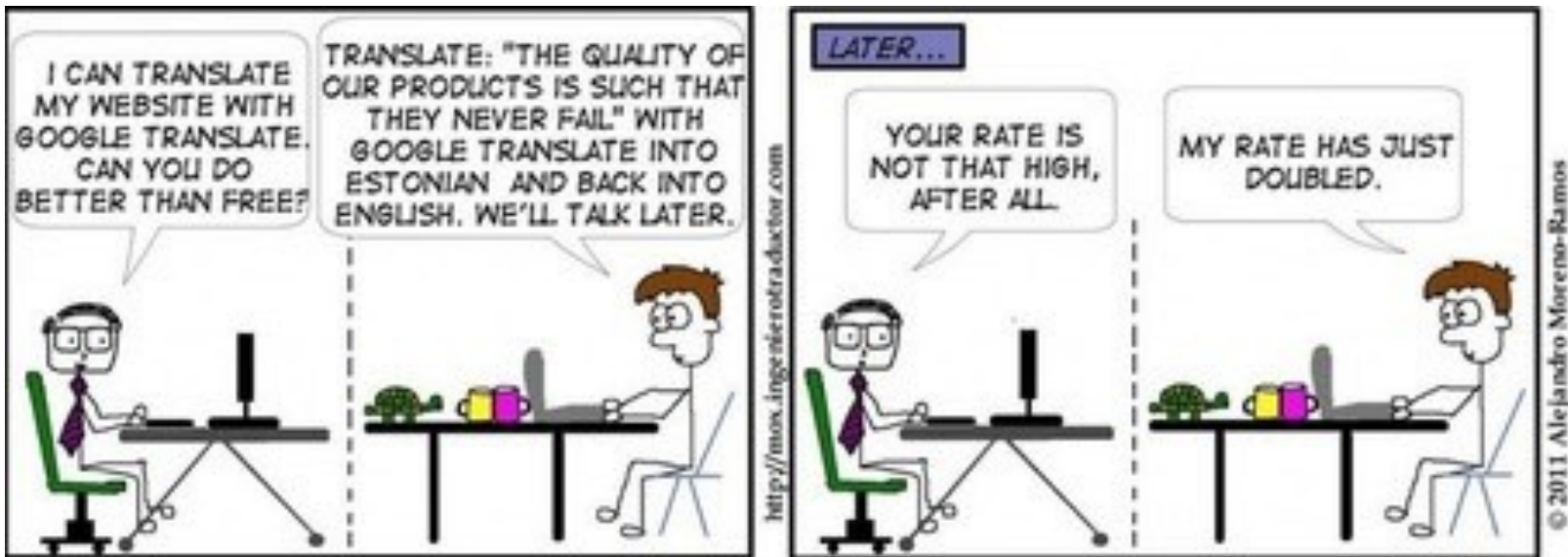
Google Translate



What have we learned in the last 30 years?

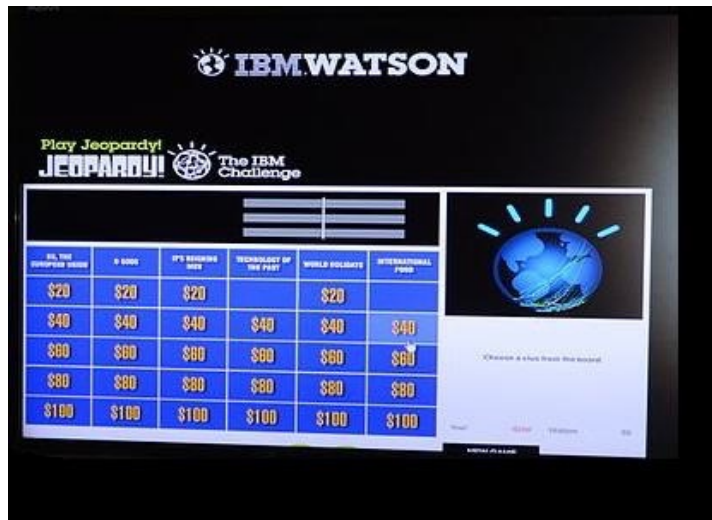
Perhaps the head of machine translation at Google (since 2004) may know ??

Och, Franz Josef, and Hermann Ney. "A systematic comparison of various statistical alignment models." *Computational linguistics* 29.1 (2003): 19-51.

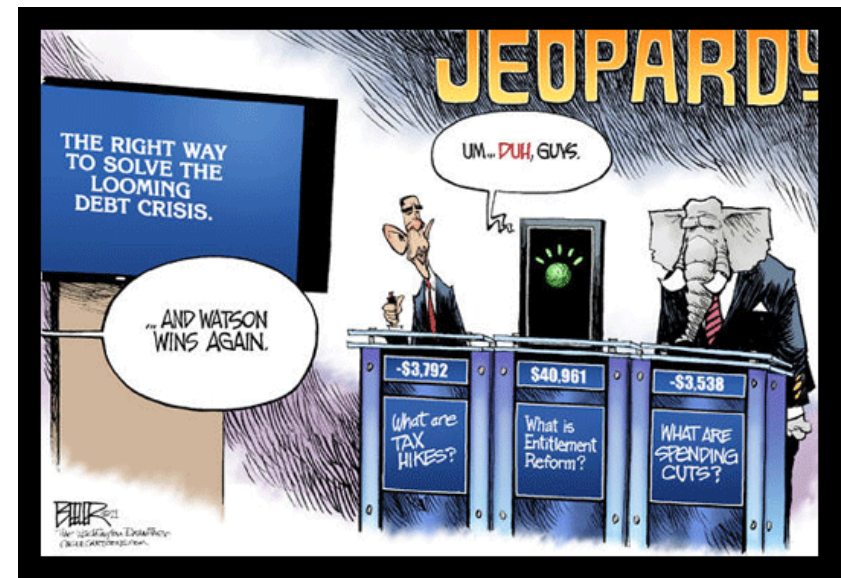


Watson

Building Watson: An Overview of the DeepQA Project, David Ferrucci et al., 2010



Ferrucci, David, et al. "Watson: Beyond Jeopardy." *Artificial Intelligence* (2012).



Personal Assistants



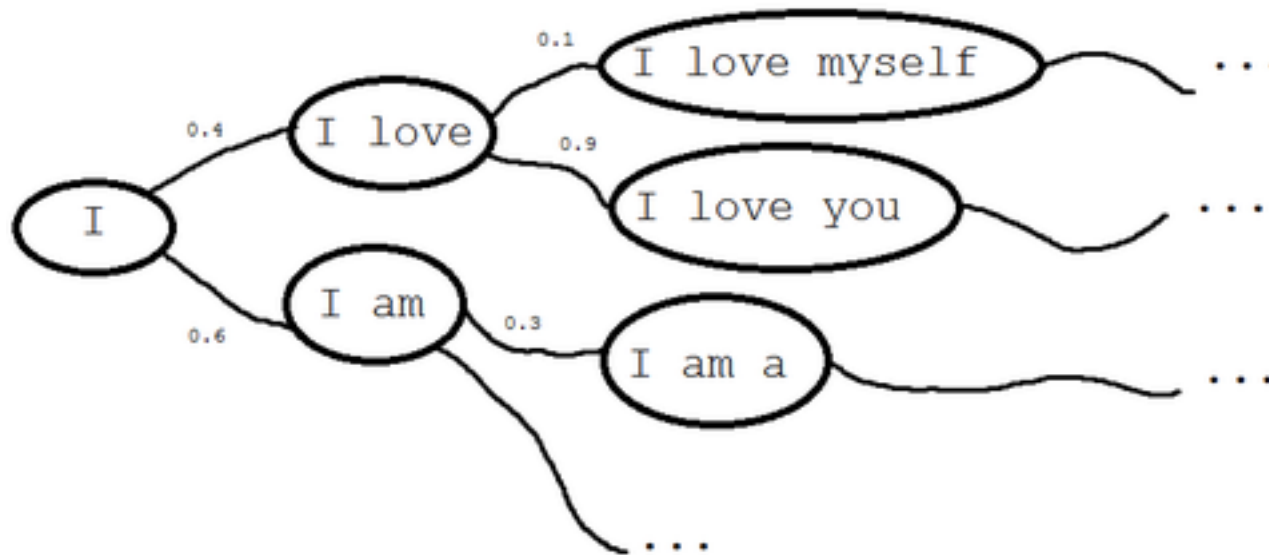
Microsoft's Twitter Bot: beware the use of AI



2016 - now we're getting *serious!*

Question: (how) did bots influence the most recent US presidential election?

A 'Simple' Problem: Predictive Text

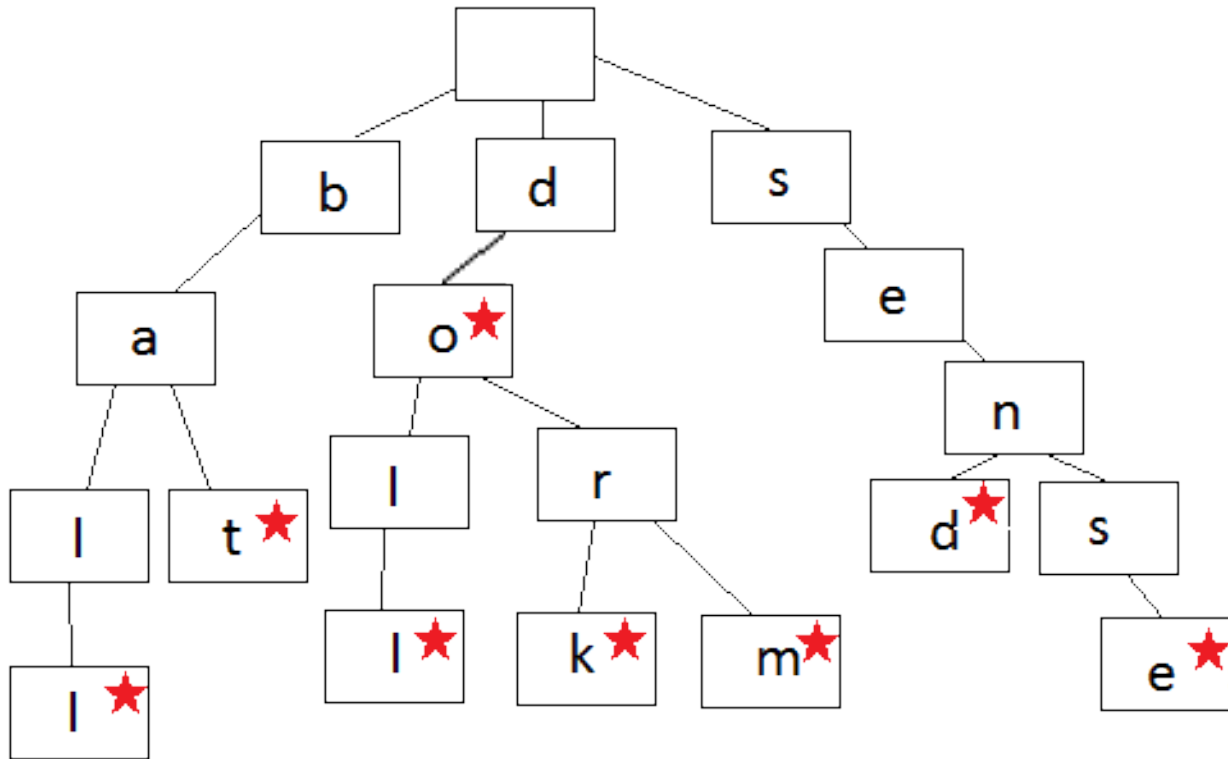


Most machine learning algorithms in commercial products (like SwiftKey on Android) use Markov chains. The example, above, illustrates a chain of words and probabilities based on previous texting.

For individual words we can use Markov chains of letters.

retrieve data structure

<http://tech.bragboy.com/2010/04/trie-in-java.html>



Idea: weight the branches like with Markov changes to

Problem: Text predictor system

Analyse the advantages and disadvantages of such an approach

Design, implement and test a prototype system that demonstrates the *feasibility*

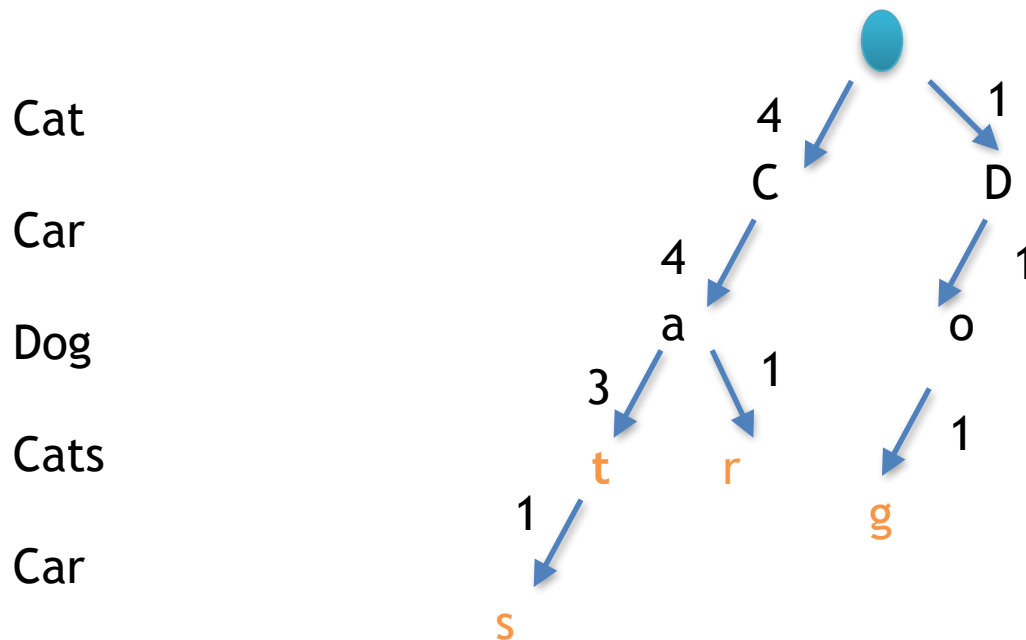


Hint: you may wish to use the *trie* data structure you have already developed

Problem: Text predictor system

1) Read in words from input file (ignore case and non alphabetic characters)

Build weighted trie.



Problem: Text predictor system

2) After trie is built, input a text string (possibly empty) and suggest most likely words (up to a max of 5, eg) based on the weighted trie

If word wanted is in the list of suggestions, update the trie with the new count

Otherwise, ask user to input new word, and also update trie

3) Save the weighted trie to an output file, and permit it to be read into predictive text system

4) Add function to merge 2 already saved weighted tries