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 conversion alone — between the program and a real human.
Hey Burt, ask if it has a favourite colour.

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

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!


“...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-)**


---

<table>
<thead>
<tr>
<th>I am the psychotherapist. Please, describe your problems.</th>
<th>Each time you are finished talking, type BET twice.</th>
</tr>
</thead>
<tbody>
<tr>
<td>I have a terrible headache.</td>
<td></td>
</tr>
<tr>
<td>Maybe you should consult a doctor of medicine, I am a psychiatrist.</td>
<td></td>
</tr>
<tr>
<td>Okay, I'll do that then.</td>
<td></td>
</tr>
<tr>
<td>Can you elaborate on that?</td>
<td></td>
</tr>
<tr>
<td>What are you talking about? No I cannot elaborate. See you later.</td>
<td></td>
</tr>
<tr>
<td>Why do you say that?</td>
<td></td>
</tr>
<tr>
<td>Now you're just talking nonsense!</td>
<td></td>
</tr>
<tr>
<td>What makes you believe now I am just talking nonsense?</td>
<td></td>
</tr>
<tr>
<td>You are not making any sense.</td>
<td></td>
</tr>
</tbody>
</table>

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-)


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)


“I think you misunderstood me when I said I wanted our factory to go all green.”
NLP For Software Engineers - HCI Design


15 years later
Formal Languages for NLP

Google Translate

What have we learned in the last 30 years?

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

Watson

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

Personal Assistants

Siri

Google Assistant

Amazon Alexa

Cortana
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


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.

```
Cat
Car
Dog
Cats
Car
```

```
- Cat
  - C
    - a
      - t
        - r
          - s
  - a
  - D
    - o
      - g

4
1
1
1
4
3
1
1
1
1
1
1
4
1
```
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