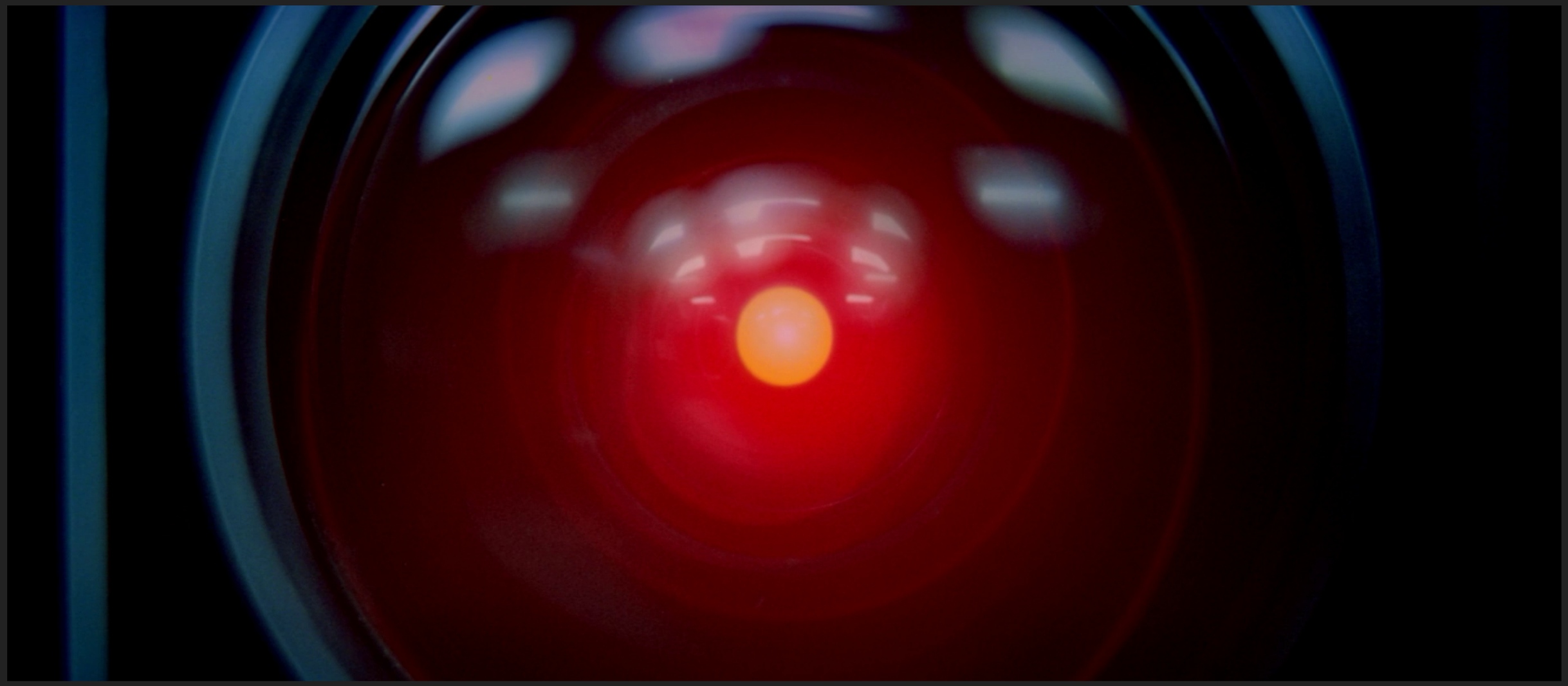


# NATURAL LANGUAGE PROCESSING

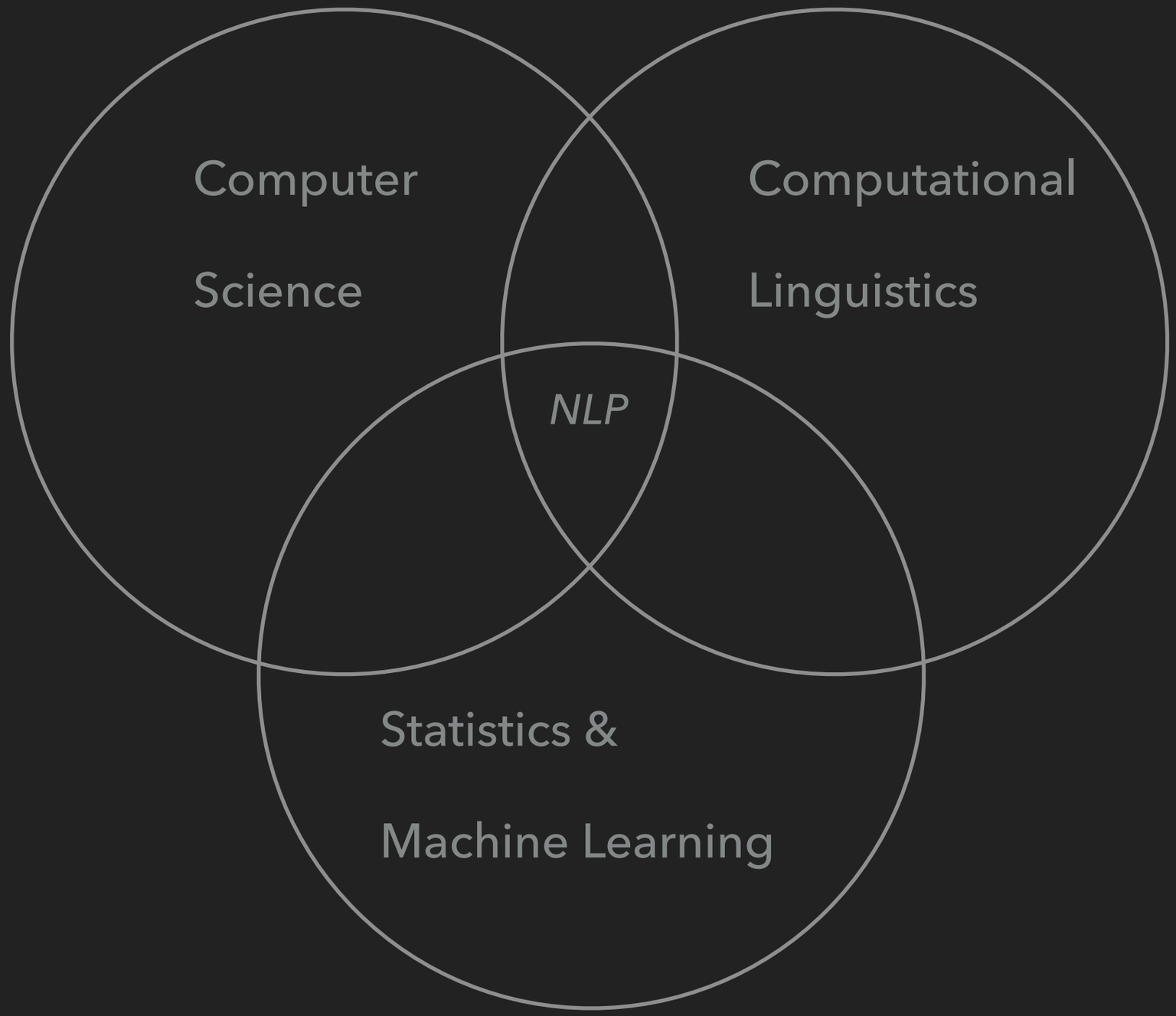
# OUTLINE

- Introduction
- History of natural language and NLP
- NLP and Artificial Intelligence
- Areas in NLP (major evaluations and tasks)

NLP IS A SUBFIELD OF COMPUTER SCIENCE, LINGUISTICS, AND ARTIFICIAL INTELLIGENCE CONCERNED WITH THE INTERACTIONS BETWEEN COMPUTERS AND HUMAN LANGUAGES

Siri

# SUBFIELD CIRCLES



# MAIN CHALLENGES

- Natural language understanding
- Natural language generation
- Speech recognition

## GOAL

to have computers *understand* natural language in order to perform useful tasks

## HOW

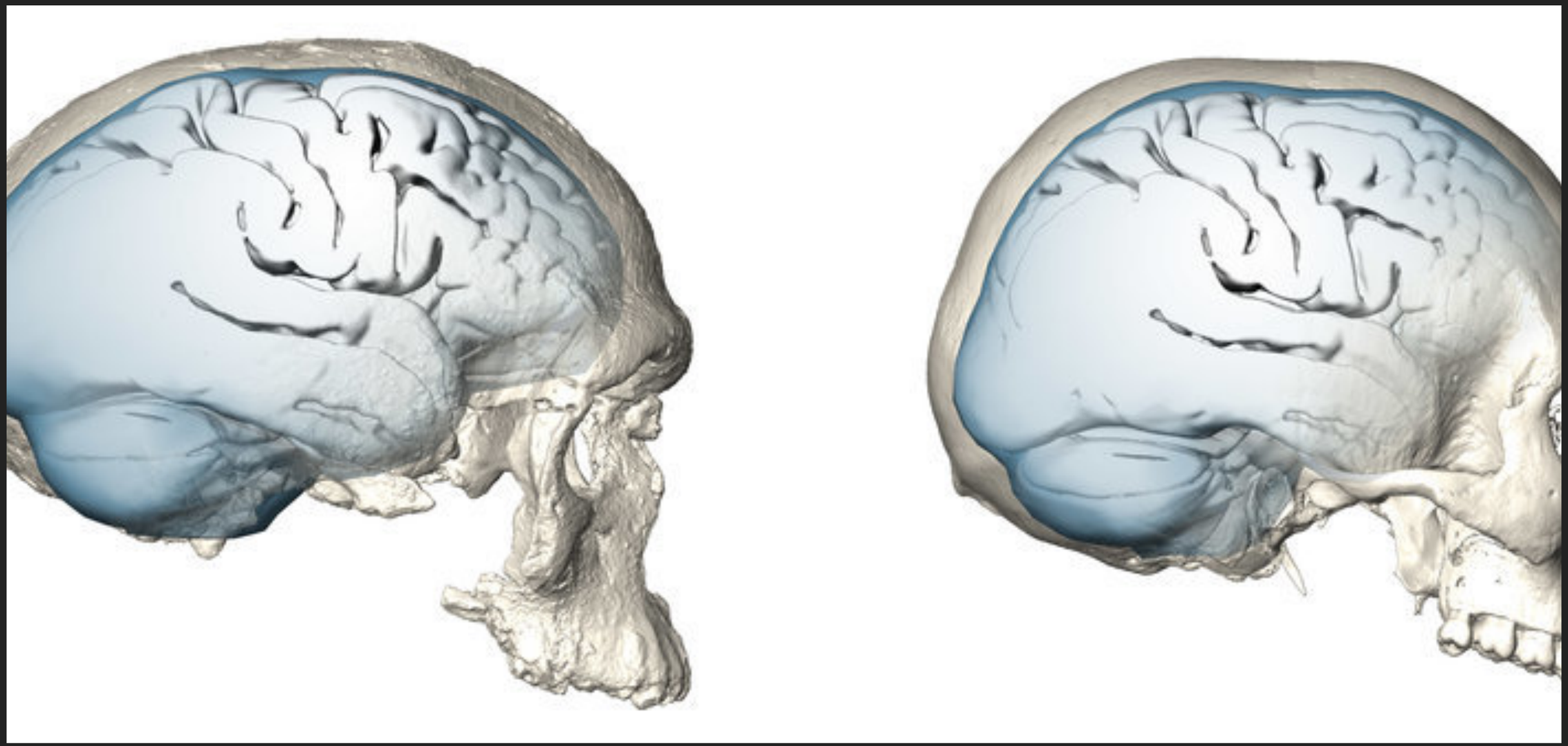
*transforming* free-form text into structured data and back

# WHY NLP IS DIFFICULT

> THE POPE'S BABY STEPS ON GAYS
>
> The Times

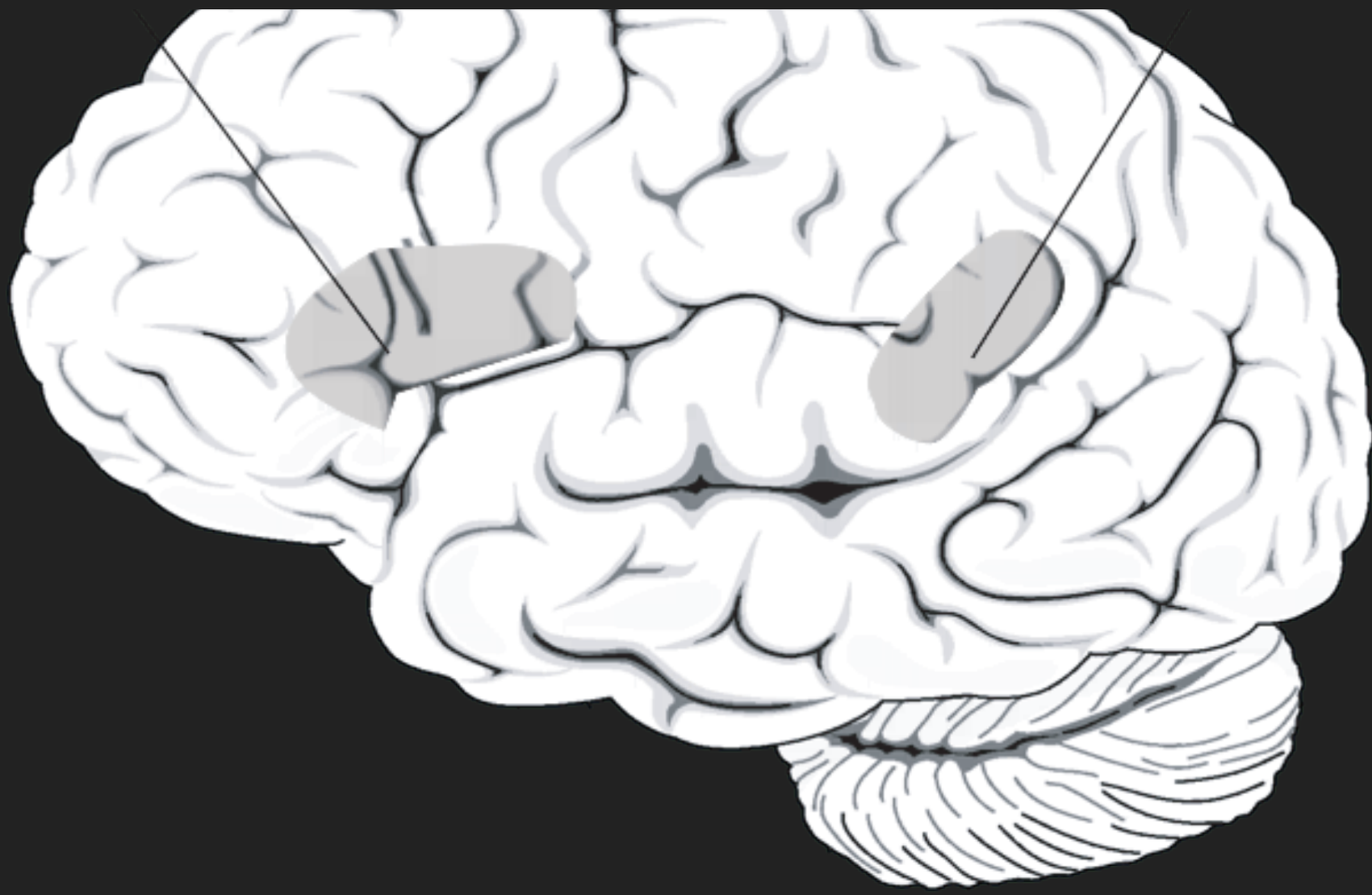
# THE EMERGENCE OF HOMO (2 – 1.7 MLN YEARS)

# BRAIN EVOLUTION – VALOIS RUBICON (2.5 MLN YEARS)

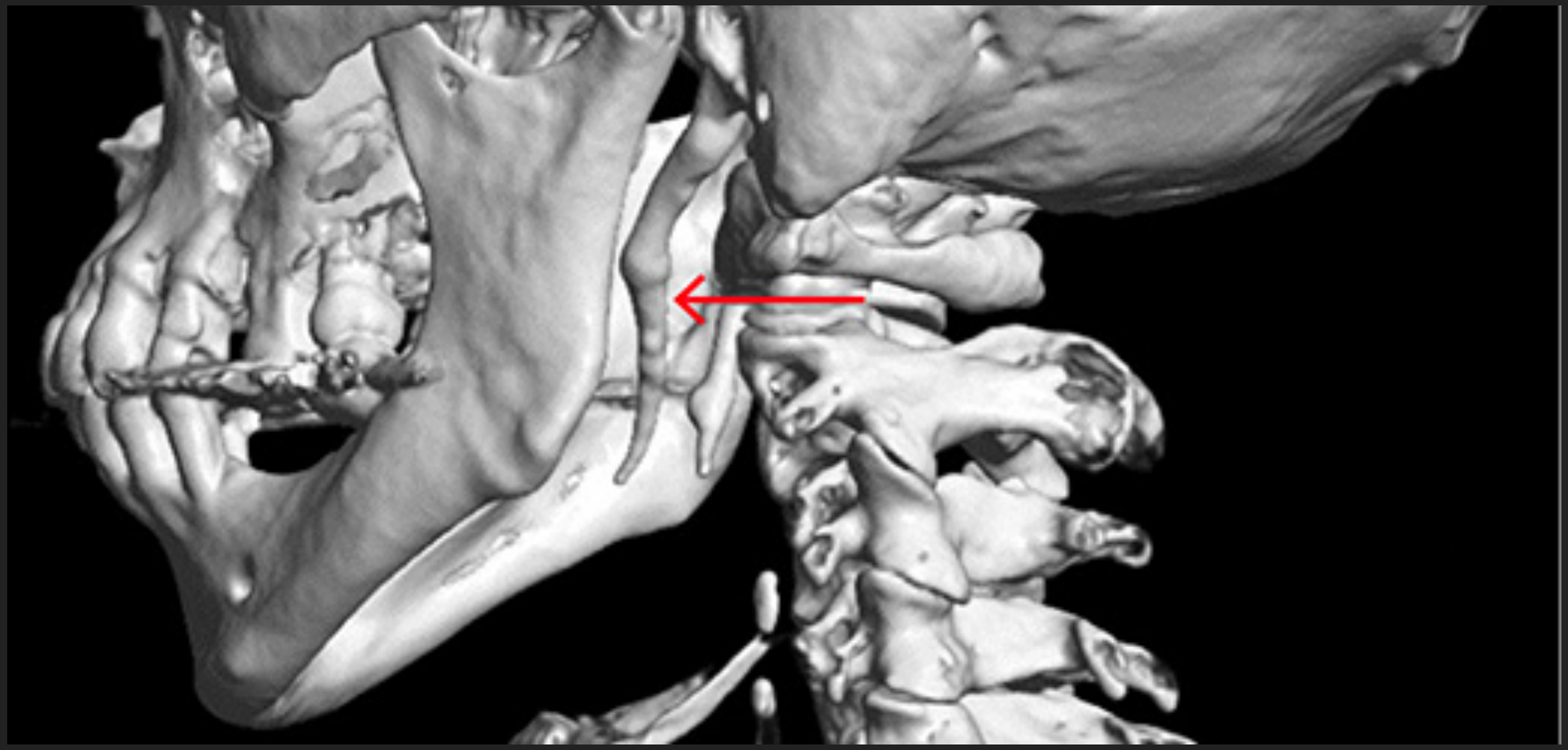


# BRAIN EVOLUTION – HUNTING AND FIRE

# HOMO'S EVOLUTION – BROCA'S AND WERNICKE'S AREAS



# HOMO'S EVOLUTION – STYLOID PROCESS OF TEMPORAL BONE



# HOMO'S EVOLUTION – THE EMERGENCE OF SPEECH FUNCTION

# THE EVOLUTION OF WRITING SYSTEMS (5000 YEARS)

- Pictograph system
- Hieroglyph system (logography)
- Syllabary
- Phonetic alphabet

# HISTORY OF NLP

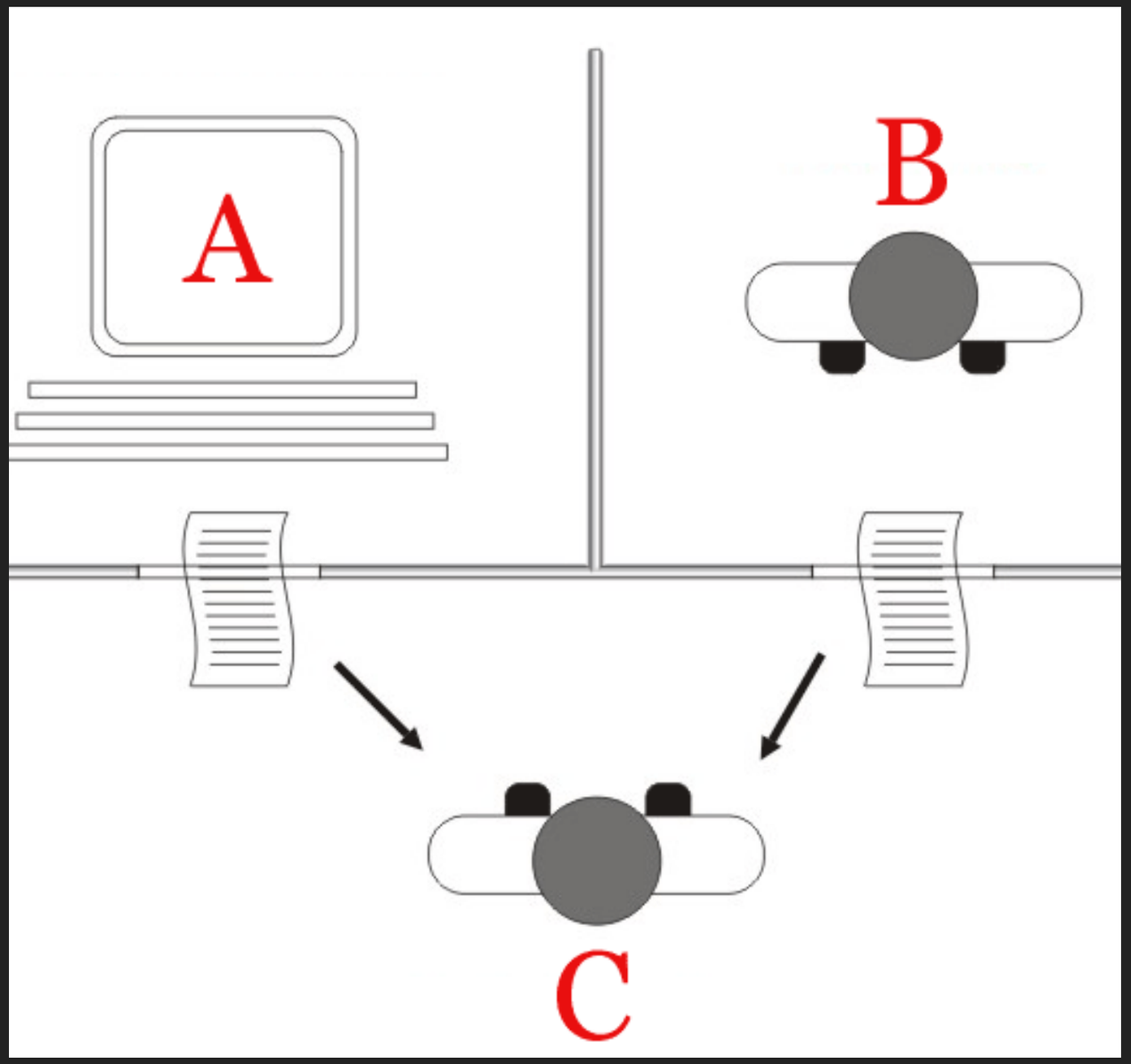
- The first patents for "translating machines" (mid-1930s)
- "Computing Machinery and Intelligence" by Alan Turing (1950)
- The Georgetown experiment in automatic translation (1954)
- ELIZA (1960)
- Hand-written rules, corpus linguistics, statistical models, part-of-speech tagging, machine learning (1980 - 2000)
- Representation learning, deep neural-network-style machine learning (2010 - )

# LINGUISTIC RELATIVITY

- The *strong* version says that language *determines* thought and that linguistic categories limit and determine cognitive categories
- The *weak* version says that linguistic categories and usage only *influence* thought and decisions

# TURING TEST

The "standard interpretation" of the Turing test, in which player C, the interrogator, is given the task of trying to determine which player – A or B – is a computer and which is a human. The interrogator is limited to using the responses to written questions to make the determination.



# LINGUISTICS' STRUCTURES

- Phonetics
- Morphology
- Syntax
- Semantics
- Pragmatics
- Discourse
- Semiotics

# MAJOR EVALUATIONS AND TASKS

- Information search and extraction
- Information classification
- Author identification (gender/age/social group)
- Named entity recognition
- Sentiment analysis
- Question answering
- Automatic summarization
- Machine translation
- Conversational agents
- Natural language generation
- Speech recognition

# MAJOR METHODS

- Linguistics structures and rules
- Linear algebra
- Mathematical statistics
- Probability theory
- Machine Learning and Deep Learning

# NLTK LIBRARY (PYTHON)

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum.

# NLTK LIBRARY (PYTHON) - - TOKENIZE AND TAG SOME TEXT

```
>>> import nltk
```

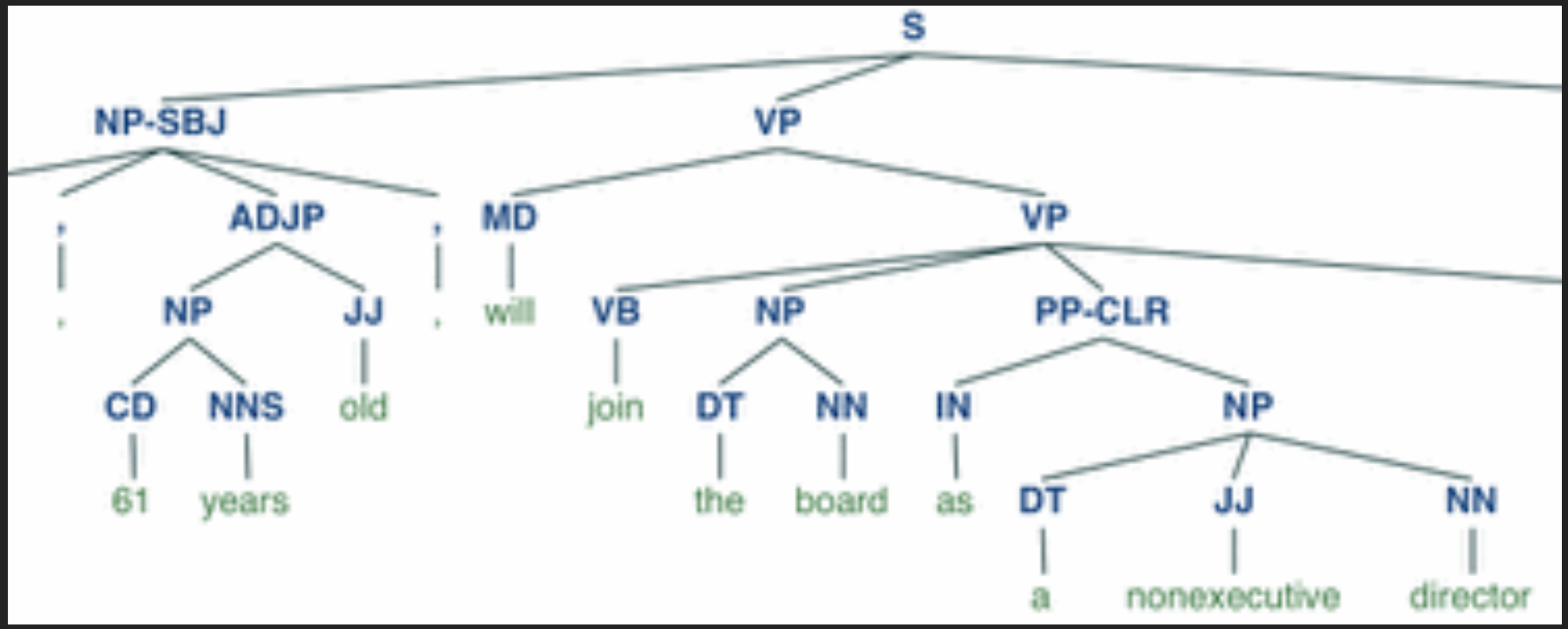
```
>>> sentence = """At eight o'clock on Thursday morning
... Arthur didn't feel very good."""
>>> tokens = nltk.word_tokenize(sentence)
>>> tokens
['At', 'eight', "o'clock", 'on', 'Thursday', 'morning',
'Arthur', 'did', "n't", 'feel', 'very', 'good', '.']
>>> tagged = nltk.pos_tag(tokens)
>>> tagged[0:6]
[('At', 'IN'), ('eight', 'CD'), ("o'clock", 'JJ'), ('on', 'IN'),
('Thursday', 'NNP'), ('morning', 'NN')]
```

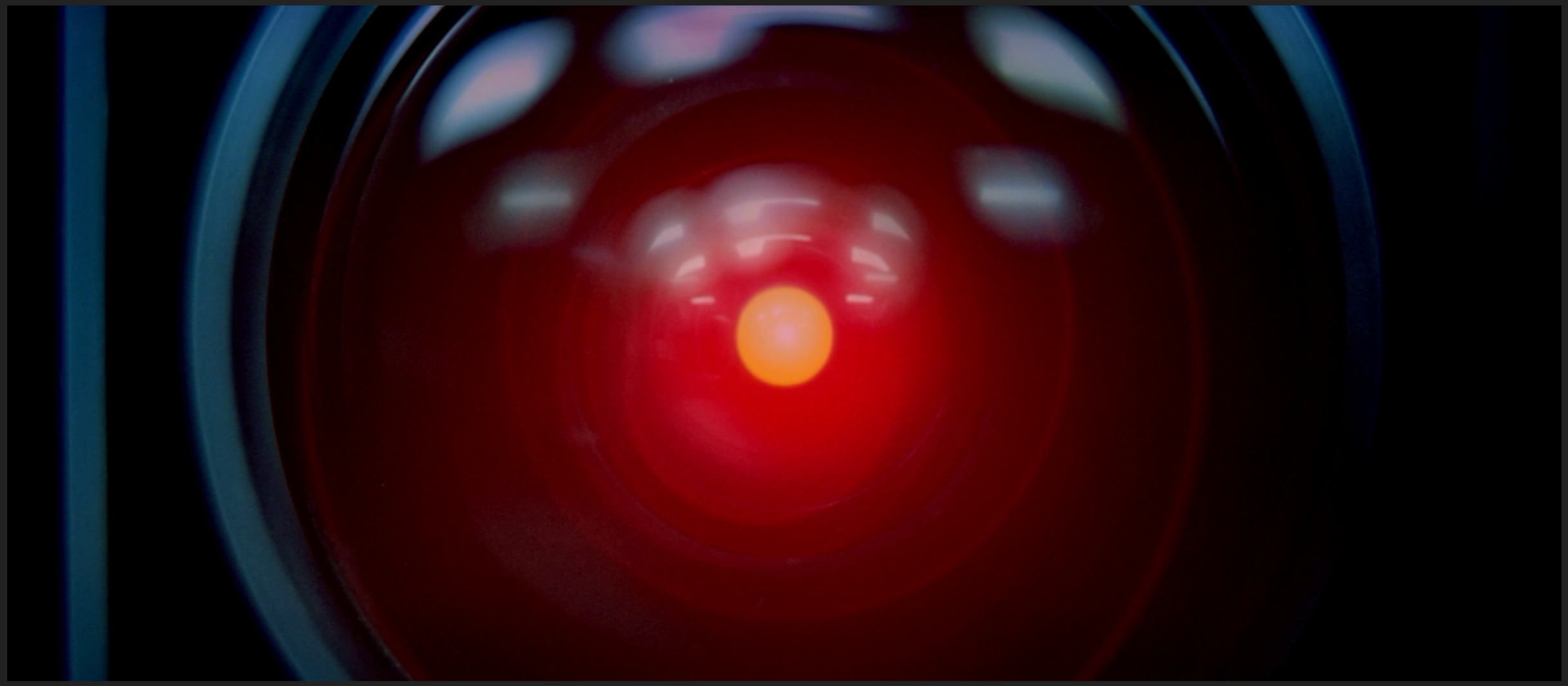
# NLTK LIBRARY (PYTHON) – IDENTIFY NAMED ENTITIES

```
>>> entities = nltk.chunk.ne_chunk(tagged)
>>> entities
Tree('S', [('At', 'IN'), ('eight', 'CD'), ("o'clock", 'JJ'),
           ('on', 'IN'), ('Thursday', 'NNP'), ('morning', 'NN'),
         Tree('PERSON', [('Arthur', 'NNP')]),
           ('did', 'VBD'), ("n't", 'RB'), ('feel', 'VB'),
           ('very', 'RB'), ('good', 'JJ'), ('.', '.')])
```

# NLTK LIBRARY (PYTHON) – DISPLAY A PARSE TREE

```
>>> from nltk.corpus import treebank
>>> t = treebank.parsed_sents('wsj_0001.mrg')[0]
>>> t.draw()
```





# NATURAL LANGUAGE PROCESSING