PHIL 7010: Formal Methods for AI, Data, and Algorithms

Week 6
Natural Language Processing

Boris Babic,
HKU 100 Associate Professor of Data Science, Law and Philosophy

Today

Learning goals

- Understand the definition and scope of Natural Language Processing (NLP)
- Understand essential NLP techniques
- Gain insights into word embeddings and the Word2Vec model
- Identify the main challenges in NLP

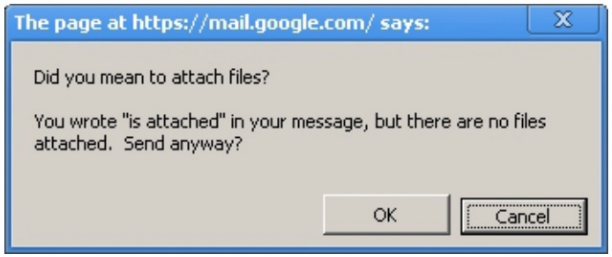
Natural Language Processing

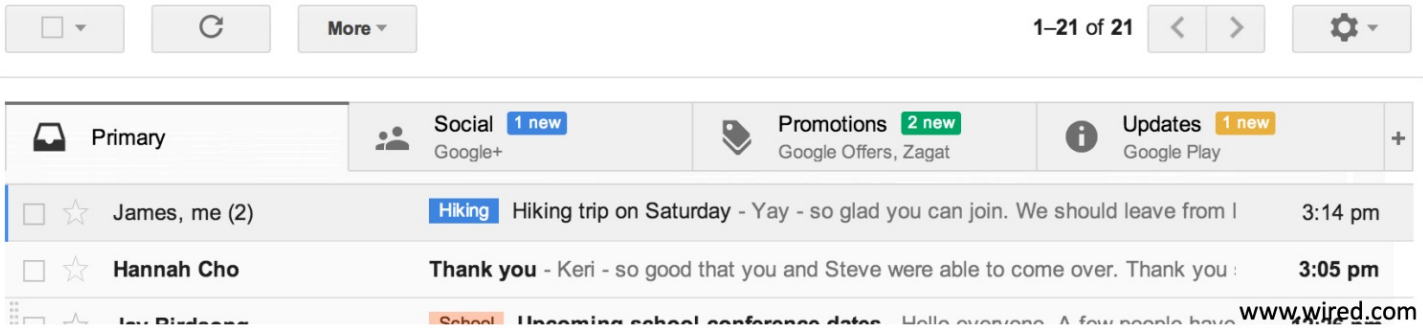
- **Wiki**: Natural language processing (NLP) is a field of computer science, artificial intelligence, and computational linguistics concerned with the interactions between computers and human (natural) languages.
- "Natural" languages
  - English, Mandarin, French, Spanish, Arabic
  - NOT python, R, Java, ...
- Ultimate goal: Natural human-to-computer communication
- Sub-field of Artificial Intelligence, but very interdisciplinary
  - Computer science, human-computer interaction (HCI), linguistics, cognitive psychology, speech signal processing (EE), ...

Go beyond the keyword matching

- Identify the structure and meaning of words, sentences, texts and conversations
- Deep understanding of broad language
- NLP is all around us

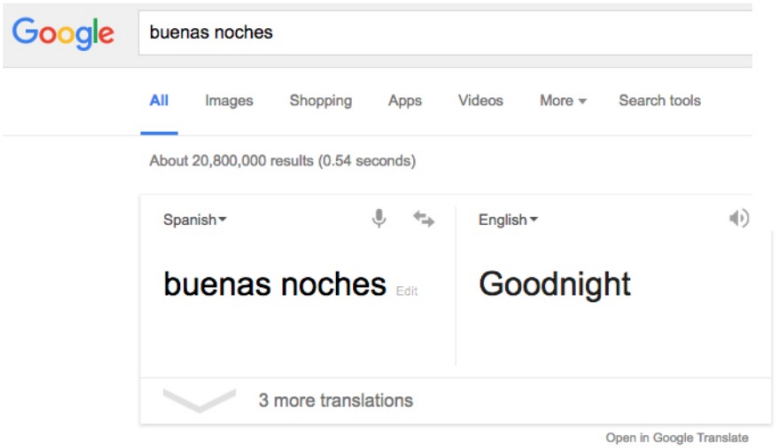
Applications: Text Classification



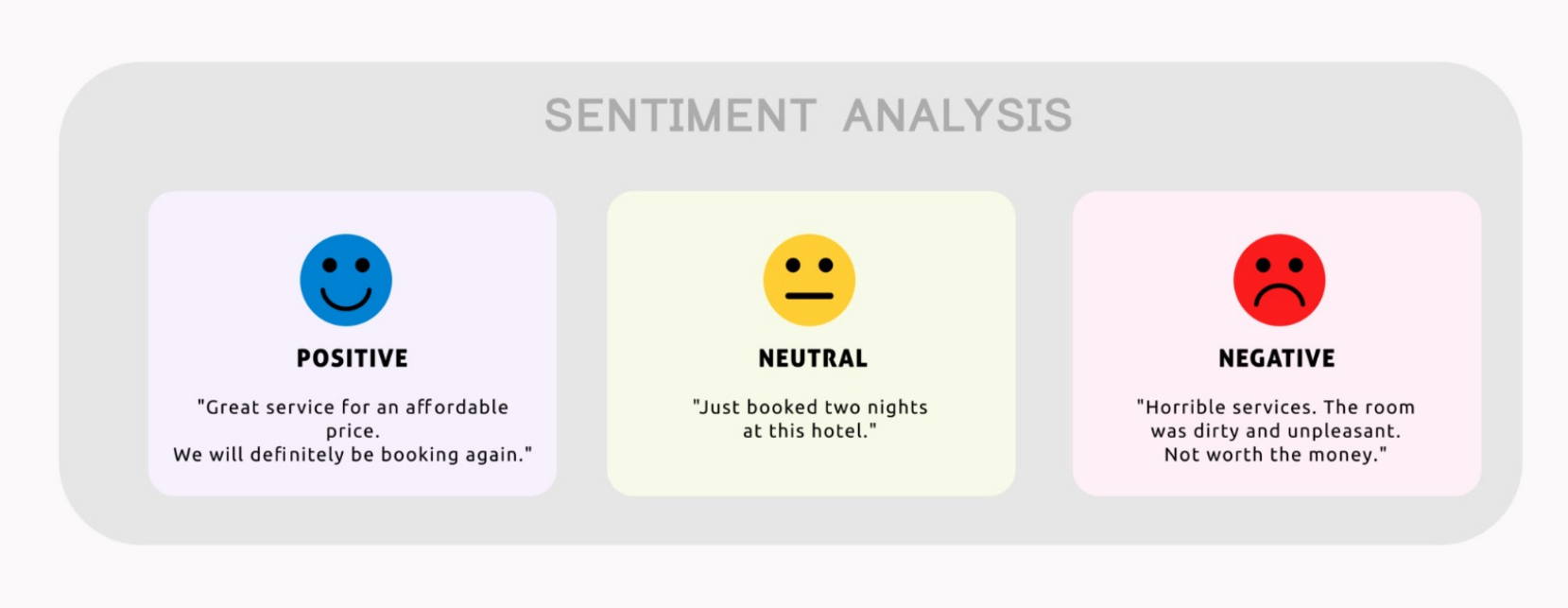


www.wired.com

Applications: Machine Translation

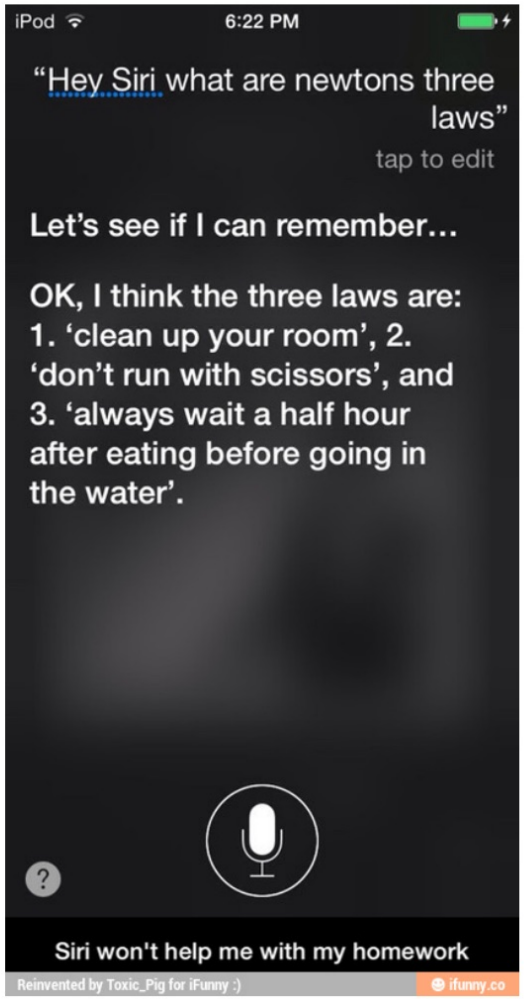


Applications: Sentiment Analysis



Applications: Question answering

'Watson' computer wins at 'Jeopardy'

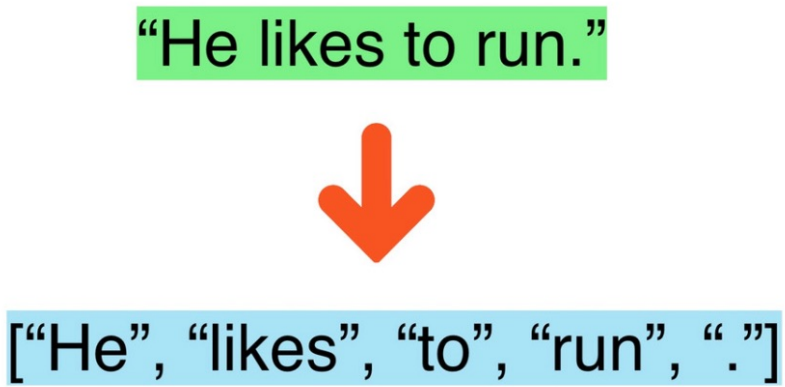


How Does NLP Work?

- NLP models work by finding relationships between the constituent parts of language — for example, the letters, words, and sentences found in a text dataset.
- Before a model processes text for a specific task, **the text often needs to be preprocessed** to improve model performance or to turn words and characters into a format the model can understand
- Various techniques may be used in this data preprocessing:
  - Tokenization
  - Stop word removal
  - Stemming and lemmatization

Data Preprocessing: Tokenization

- We usually start an NLP project with a large body of text, called a **corpus**.
- This could be a collection of tweets, website reviews or transcriptions of films, for example. We need to pre-process our corpus to give it enough structure to be used in a machine learning model and **tokenization is the most common first step**.
- Tokenization is the process of breaking down a corpus into tokens. The procedure might look like segmenting a piece of text into sentences and then further segmenting these sentences into individual words, numbers and punctuation, which would be tokens.
- Each token should be chosen to be as small as possible while still carrying meaning on its own.



Data Preprocessing: Stop word removal

- Stop word removal aims to remove the most commonly occurring words that don't add much information to the text.
- For example, "the," "a," "an," and so on.

```
Output:

Tokenized text with stop words :

['Oh', 'man', ',', 'this', 'is', 'pretty', 'cool', '.', 'We', 'will', 'do', 'more', 'such', 'things', '.']

Tokenized text with out stop words :

['Oh', 'man', ',', 'pretty', 'cool', '.', 'We', 'things', '.']
```

Data Preprocessing: Stemming and lemmatization

Stemming and lemmatization:

- Stemming is an informal process of converting words to their base forms using heuristic rules.
  - For example, "university," "universities," and "university's" might all be mapped to the base "univers."
  - One limitation in this approach is that "universe" may also be mapped to univers, even though universe and university don't have a close semantic relationship.
- Lemmatization is a more formal way to find roots by analyzing a word's morphology using vocabulary from a dictionary.
  - Lemmatization is generally more accurate and preserves the semantic meaning of the word, but it's also computationally more complex.

Word	Stemming	Lemmatization
information	inform	information
informative	inform	informative
computers	comput	computer
feet	feet	foot

Bag-of-Words

- After tokenization and pre-processing, we are left with variable-length sequences of text, but the problem is machine learning algorithms require fixed-length vectors of numbers.
- The simplest approach to overcome this is by using a **bag-of-words**, which simply counts how many times each word appears in a document.
  - It's called a bag because the order of the words is ignored - we only care about whether a word appeared or not.
- The linguistic reasoning behind this approach is that similar documents share similar vocabularies.
  - For example, football articles will often use words like score, pass, team whereas weather reports will use a completely different set of words like rain, sun, umbrella.

	the	red	dog	cat	eats	food
1. the red dog →	1	1	1	0	0	0
2. cat eats dog →	0	0	1	1	1	0
3. dog eats food →	0	0	1	0	1	1
4. red cat eats →	0	1	0	1	1	0

Bag-of-Words

- We've now converted the text into a bag-of-words matrix where each row corresponds to a single document (or say, sentence) in the corpus. Each column corresponds to a token in the vocabulary.
- We can think of it as a collection of points in a multi-dimensional vector space
- Importantly, the dimension of this space if fixed, i.e. each vector has the same length.
- It allows us to measure the distances between these points among other things. Points (documents) that are close together will correspond to documents being similar in their vocabularies.
- In the NLP domain it is much more common to use Cosine Similarity. This measures the cosine of the angle between any two points (more precisely their vectors starting from the origin).
  - e.g. the closer the score 1, the smaller the angle between the vectors and the more similar the documents are

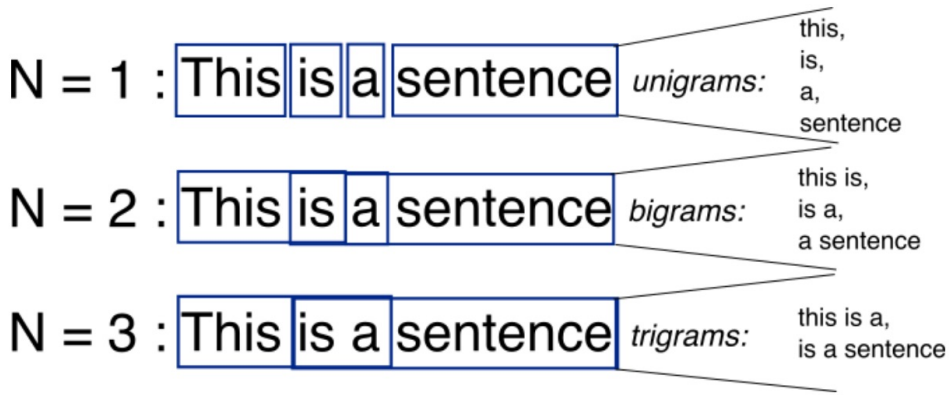
Challenges of Bag-of-Words

Whilst using a bag-of-words is a great tool for simple NLP applications, it does have a number of drawbacks that we need to be aware of.

- There is no way to handle Out-of-Vocabulary (OOV) words. If a new word appears in a later document, it will just be dropped.
- It isn't able to capture similarity between synonyms.
- Word order is lost so words have no relationship to each other. For example, "man eats bread" is very different to "bread eats man" but they would have the same representations.

n-gram

- One way to get around the problem of losing word order information is to use n-grams. This is when we group chunks of n tokens together to behave as if they were a single token
- A 2-gram (aka bigram) would have 2 tokens per chunk, a 3-gram (aka trigram) would have 3 tokens per chunk, etc.
- This helps us capture some context that using single tokens wouldn't. The vocabulary then becomes the collection of n-grams produced.
- Measuring similarity is exactly the same as before. However, using n-grams can significantly increase the size of the vocabulary making computations slower.



Word Embedding

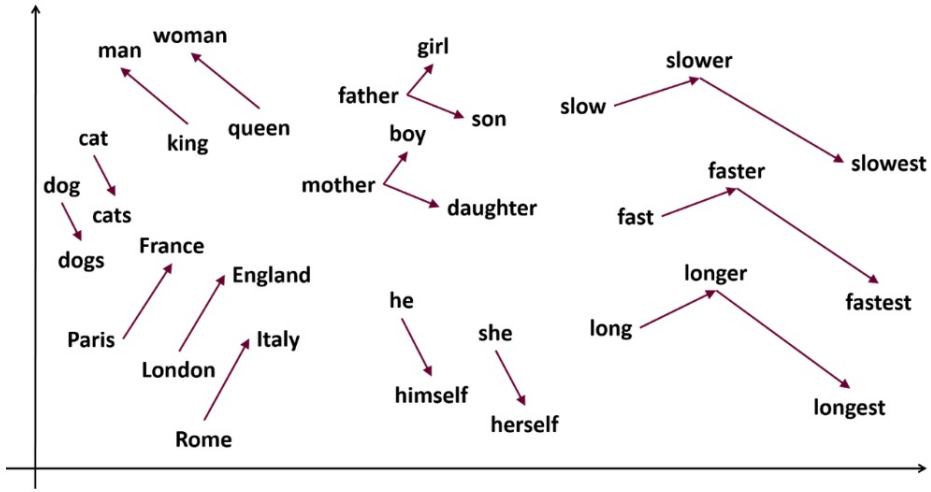
- We've seen how to vectorize words using one-hot encoding, but this is memory inefficient and fails to capture relationships between words, even when we use n-grams, it becomes computationally inefficient when the vocabulary size increases.
- A better way then would be to represent words as **shorter and denser vectors** that capture some meaning between words.
- An **embedding** is simply a representation of an object (e.g. a word, movie, graph, etc) as a vector of real numbers. It embeds an object into a high-dimensional vector space.
  - For example, let's say we have a collection of video games. We can represent each game by measuring a number of its attributes like $[< fantasy >, < strategy >, < multiplayer >, < action >, < adventure >]$.
  - So a game like 'Minecraft' could be represented by [0.1,0.6,0.4,0.5,0.9]
- Notice how there are many different ways to embed the same object, the features we hand-selected may not be the best ones to represent these objects.
- How do we find their best representations?

Word2Vec

- To train a word embedding, we first need to ask ourselves what makes two words semantically similar?
- One popular answer to this is the distributional hypothesis, which says that "words which appear in similar contexts (i.e. share similar surrounding words) have similar meanings".
  - For example, consider the sentence "My family enjoys eating bacalhau at Christmas".
  - You probably have no idea what 'bacalhau' is but just from the context we can infer that it is some kind of food. (In fact, the word refers to Portuguese salted cod.)
- There are many algorithms that use this idea to train word embeddings. We are going to focus on the main one, namely **Word2vec**.

Word2Vec

- Word2vec is a way of learning word embeddings by training shallow neural networks
- Two amazing things come out of Word2vec:
  - similar words are close together
  - we can perform addition and subtraction on these word vectors.
- For example, “king” - “man” + “woman” = ”queen”. That is, if we take the vectors for king, man, woman and add/subtract them in this way, we will end up with a vector close to the one corresponding to queen.
- This means that these vectors can capture very precisely abstract concepts (like gender and royalty) without any input from us.



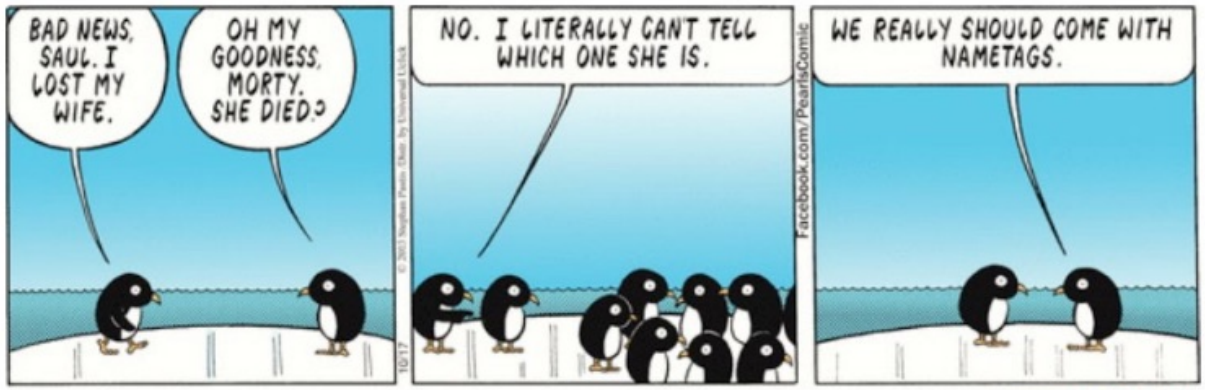
Modeling

After data is preprocessed, it is fed into an NLP architecture that models the data to accomplish a variety of tasks.

- Traditional Machine learning NLP techniques: take output from the vectorizer as its input to perform simple tasks like classification
  - Logistic regression: sentiment analysis, spam detection, and toxicity classification
  - Naive Bayes: spam detection, classifications
    - $P(label|text) = \frac{P(label)*P(text|label)}{P(text)}$
- Deep learning methods: take as input a word embedding and, at each time state, return the probability distribution of the next word as the probability for every word in the dictionary.
  - Convolutional Neural Network (CNN)
  - Recurrent Neural Network (RNN)
  - Transformers

Challenges in NLP

- NLP is hard
  - Human languages are messy, ambiguous, and ever-changing
- What challenges get in the way of understanding and responding to natural language?
  - Ambiguity
  - Implicit references
  - Language is dynamic
  - Scale
  - many more .....
- Ambiguity as an example:
  - Word sense: bank (finance or river?)
  - Part of speech: chair (noun or verb?)
- Scale:
  - Wikipedia: 2.9 billion words (English)
  - Web: several billions of words



credit: A. Zwicky

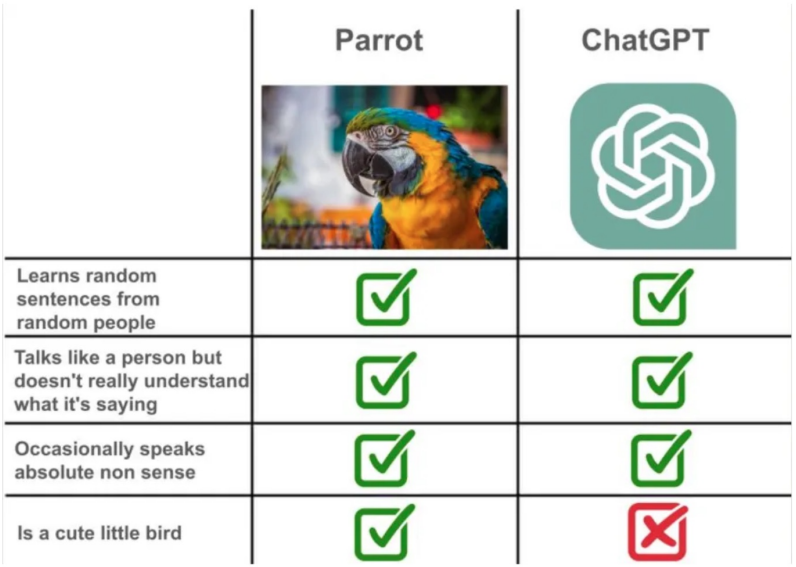
Controversies Surrounding NLP

NLP has been at the center of a number of controversies. Some are centered directly on the models and their outputs, others on second-order concerns, such as who has access to these systems, and how training them impacts the natural world.

- Stochastic parrots
- Black box
- Coherence versus sentience

Stochastic parrots

- A 2021 paper titled "On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?" by Emily Bender, Timnit Gebru, Angelina McMillan-Major, and Margaret Mitchell examines how language models may repeat and amplify biases found in their training data.
- The authors point out that **huge, uncurated datasets scraped from the web are bound to include social biases and other undesirable information**, and models that are trained on them will absorb these flaws.
- Solutions?
  - Greater care in curating and documenting datasets
  - Evaluating a model's potential impact prior to development
  - Encouraging research in directions other than designing ever-larger architectures to ingest ever-larger datasets



Black box

- When a deep learning model renders an output, it's **difficult or impossible to know why it generated that particular result**.
- While traditional models like logistic regression enable engineers to examine the impact on the output of individual features, neural network methods in natural language processing are essentially black boxes.
- Such systems are said to be "not explainable," since we can't explain how they arrived at their output.
- An effective approach to achieve explainability is important, as regulators want to confirm that a natural language processing system doesn't discriminate against some groups of people, and law enforcement, where models trained on historical data may perpetuate historical biases against certain groups.
- Solutions?
  - Use simpler (explainable) models
  - Post-hoc explanation techniques (e.g. LIME, Transformer Interpret, etc.)

Coherence versus sentience

- One concern that individuals have had about the AI industry for years is machine learning programs' ability to seemingly think for themselves and express feelings.
- NLPs are often the version of AI that concerns individuals in this regard due to the computer's ability to mimic and present written text in a way that expresses the same emotions and thought patterns as humans.
- Recently, a Google engineer tasked with evaluating the LaMDA language model was so impressed by the quality of its chat output that he believed it to be sentient.

The Effective NLP Cycle

- Pick a problem (usually some disambiguation)
- Get a lot of data (usually a labeled corpus)
- Preprocessing the data (tokenizer, Embedding, check for bias...)
- Build the simplest thing that could possibly work
- Repeat:
  - Examine the most common errors are
  - Figure out what information a human might use to avoid them
  - Modify the system to exploit that information
    - Feature engineering
    - Representation redesign
    - Different machine learning methods

Today

Learning goals

- Understand the definition and scope of Natural Language Processing (NLP)
- Understand essential NLP techniques
- Gain insights into word embeddings and the Word2Vec model
- Identify the main challenges in NLP