BREAK THROUGH TECH

Machine Learning Foundations

Lab 8

Week of July 21

BREAK
THROUGH
TECH

Icebreaker: Waterfall Word Association

Waterfall Word Association

Objective:

- See how words link together in real time, mirroring the way NLP systems process language.

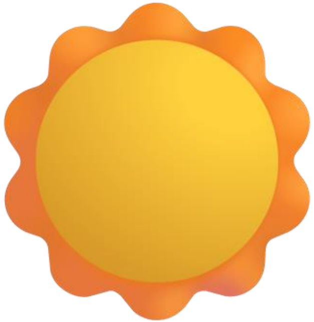
Rules:

- I'll display a slide with a word on it. You have 7 seconds to type out an associated word in chat **but don't hit enter until I say "GO!"**

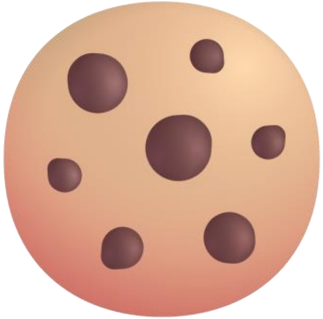
Why Are We Playing Waterfall Word Association?

- Words are interconnected and context-dependent, reflecting key principles of how natural language processing works
- The simultaneous flood of words in the chat visually demonstrates the diversity of human thought and language, similar to how NLP systems manage vast arrays of data.
- Think about language patterns and associations and how that sets the stage for us to explore NLP.

Word 1: Sunshine



Word 2: Cookie

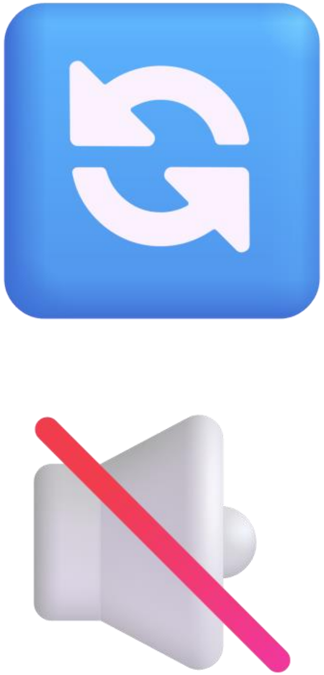


Word 3: Rainbow

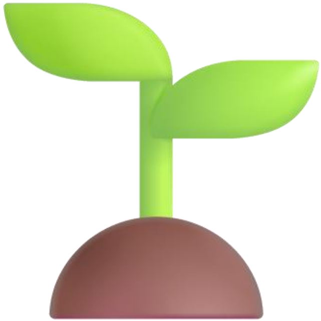
Word 4: Cat

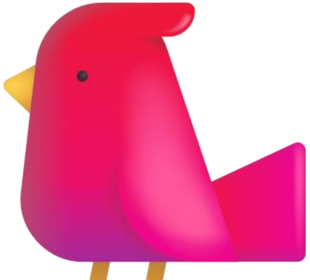
Word 5: Quasar

Word 6: Oxymoron

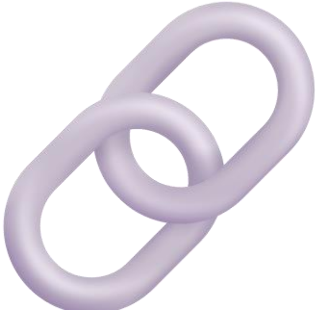


Word 7: Symbiosis





Word 8: Cryptocurrency



Wrap Up

- While the Waterfall Word Association game might seem straightforward to us, interpreting and generating language is a complex task. In the realm of NLP, machines must learn to not only recognize words but understand their contextual and associative meanings—much like linking a chain of words based on a single starting point.
- This game has shown us how humans can rapidly form associations and adapt meanings based on context. For computers, achieving this level of linguistic flexibility involves intricate algorithms and extensive learning datasets.
- Our activity was a playful way to demonstrate some of the core challenges in NLP. Like interpreting a cascade of words in our game, NLP systems must analyze text, understand sentiment, recognize patterns, and generate coherent responses.

BREAK THROUGH TECH

Week 8 Concept Overview + Q&A

Concept Overview

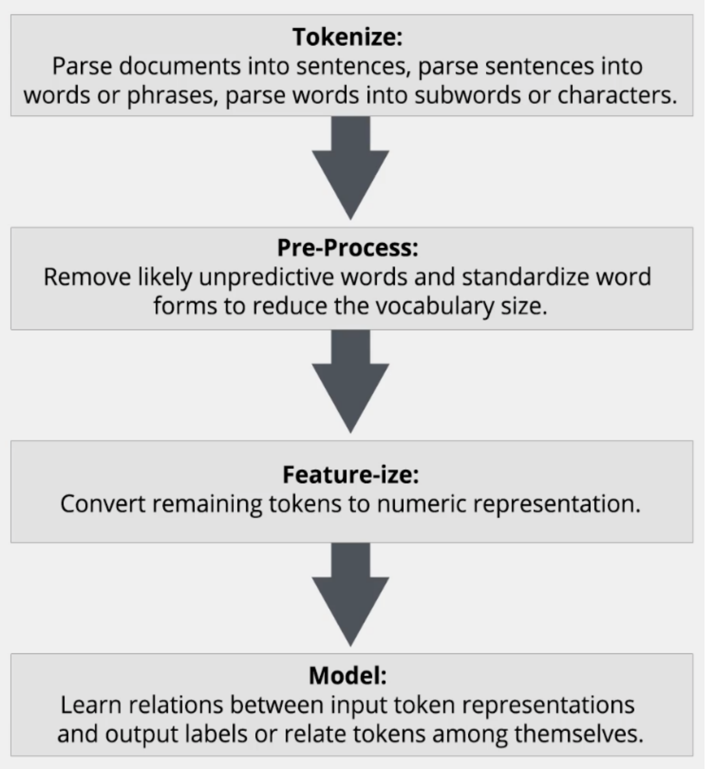
This week covered a number of topics. To refresh your memory, here is what you've completed:

- Explore the NLP pipeline
- Use various NLP preprocessing techniques to convert text to data suitable for machine learning
- Understand how vectorizers are used to convert text into numerical features and how word
- Explore how word embeddings are used to convert text into numerical features without losing the underlying semantic meaning
- Discover how deep neural networks are used in the NLP field
- Implement a feedforward neural network for sentiment analysis
- Create and implement a project plan to solve a machine learning problem

Applications of NLP

Application	**Description**
Sentiment Analysis	Determines the tone of the text - this text negative or positive?
Topic Modeling	Finds patterns in the text data that help with the understanding of the underlying themes and topics within documents.
Machine Translation	Translates text in language A to language B.
Text Summarization	Generates a short sequence of text to summarize a longer sequence of text.
Text Extraction	Automatically extracts relevant information, such as core words and phrases from documents.
Text Classification	Understands, processes, and categorizes documents.
Text Generation	Produces human-like written language.
Speech Recognition	The area of NLP which focuses on the interaction with spoken language or audio files.

NLP Steps



Tokenize:
Parse documents into sentences, parse sentences into words or phrases, parse words into subwords or characters.

Pre-Process:
Remove likely unpredictive words and standardize word forms to reduce the vocabulary size.

Feature-ize:
Convert remaining tokens to numeric representation.

Model:
Learn relations between input token representations and output labels or relate tokens among themselves.

Lemmatization
Am, are, is => be
Cats, cats', cats => cats
Playing, player, played => play

Make n-grams
= "This product is terrible, definitely not great"

Bi-grams: "this product", "product is", "is terrible", "terrible definitely", "definitely not", "not great"

Tri-grams: "this product is", "product is terrible", "is terrible definitely", "terrible definitely not", "definitely not great"

Remove stop words
How to identify a "stop world"

1. Word belongs to a pre-specified language-specific set
2. Word has document frequency > K or document frequency < J

Text Vectorization: Common Vectorizers

Binary:

Use binary presence of the token in the document

Count:

Use the count of the token in the document

Term Frequency Inverse Document Frequency (TF-IDF):

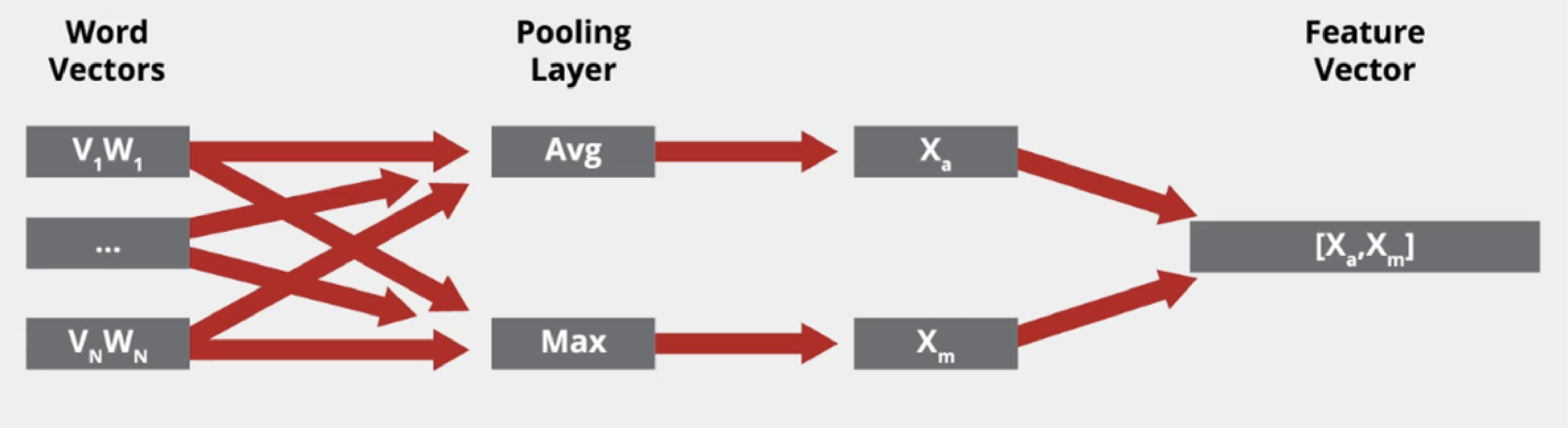
Use the term frequency within a document divided by the document frequency

Text Vectorization: Using Word Embeddings

Use word embeddings to accomplish the following:

- Seek to capture the meanings of words within a body of text
- Massively reduce feature count
- Massively reduce data sparsity
- Pool similar words based on similar semantic meaning

Text Vectorization: Using Word Embeddings - Pooling Layers



Sequence-to-Sequence Models

Special types of neural networks that deal with text data. There are different types of sequence-to-sequence models, but they all typically consist of two components:

Encoder:

A neural network that takes in a sequence of words and outputs a vector or a code that can be viewed as a summary of the input sequence.

Decoder:

A neural network that takes in the vector output of an encoder and turns it into a scalar or sequence of outputs. These can be words represented by word embeddings or other things, depending on the application.

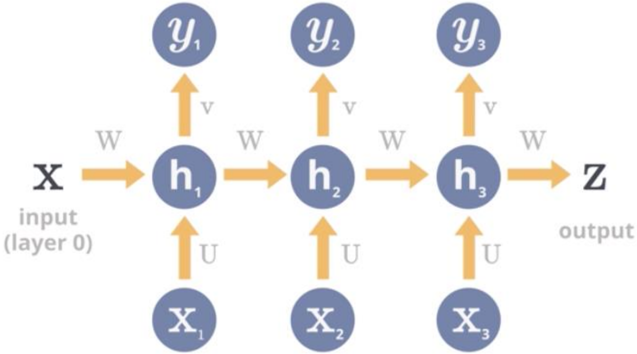
Sequence-to-Sequence Models

Deep Averaging Network:

A deep averaging network consists of two components: a traditional neural network and a word embedding.

Recurrent Neural Network (RNN):

RNNs are made up of neural networks that are applied over and over again at each element of a sequence but keeps track of a hidden state vector and uses it to encode and decode sequential information - it keeps track of previous words in a sequence. Since an RNN has a “memory,” it is able to consider the output it has learned from previous inputs in order to guide its decisions.



Large Language Models (LLMs)

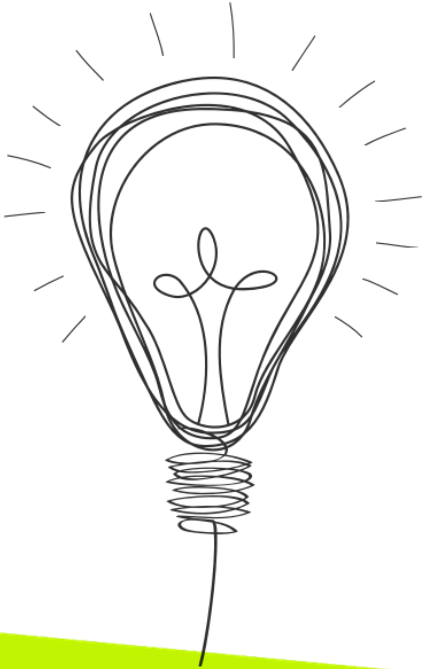
Large language models (LLMs) are ML models that comprehend and generate human-like text.

Tasks LLMs can complete:

- Summarization
- Language translation
- Writing
- Answer questions
- Text generation

Questions & Answers

What questions do you have about the online content this week?



BREAK
THROUGH
TECH

Breakout Groups: Big Picture Questions

Big Picture Questions

You have 15 minutes to discuss the following questions within your breakout groups:

- How would you decide whether to use lemmatization in a specific NLP task? What are the advantages and disadvantages of this approach, and what factors would influence your decision?
- How would you determine the appropriate tokenization strategy for a given NLP task? What are some common tokenization techniques, and how might the choice of tokenization impact downstream tasks like sentiment analysis?
- Explain the practical considerations and decision-making process when choosing between vectorizers and word embeddings for representing text data in NLP tasks? Discuss the benefits, limitations, and specific use cases of each approach, emphasizing how, why, and when you would opt for one over the other.
- Explain what sequence-to-sequence models are and why they are better suited for NLP tasks over traditional neural networks.
- Describe what a large language model (LLM) is and why it is needed.

BREAK
THROUGH
TECH

Class Discussion

Big Picture Responses **#1 How would you decide whether to use lemmatization in a specific NLP task ? What are the advantages and disadvantages and what factors would influence your decision ?**

Lemmatization is a powerful technique in NLP for reducing words to their base forms, enhancing the accuracy and depth of text analysis by preserving the semantic meaning and context of words.

Words in their **base or canonical form** is called the **lemma**.

Lemmatization often requires POS tagging (Part-of-Speech tagging) to correctly identify the grammatical category of a word (like noun, verb, adjective, etc.) and apply appropriate normalization rules.

Advantages of Lemmatization:

- **Improved Accuracy**: Lemmatization reduces words to their canonical forms, which can enhance accuracy in tasks that require understanding the meaning of words.
- **Better Coverage**: It covers a wider range of variations compared to simple stemming, potentially reducing the risk of vocabulary fragmentation.
- **Context Preservation**: Unlike stemming, which simply chops off suffixes, lemmatization takes into account the morphological analysis of words within their context, preserving their intended meaning.

Disadvantages of Lemmatization:

- **Complexity**: Lemmatization algorithms are often more complex than stemming algorithms, **requiring more computational resources and linguistic knowledge.**
- **Ambiguity Handling**: Some words may have multiple possible lemma forms depending on their usage, which introduces ambiguity.
- **Language-dependent**: The effectiveness of lemmatization can vary across languages, especially for highly inflected languages where the lemma may not always be straightforward to determine.

Big Picture Responses **#2 How would you determine the appropriate tokenization strategy for a given NLP task? What are some common tokenization techniques, and how might the choice of tokenization impact downstream tasks like sentiment analysis?**

Tokenization involves understanding the characteristics of the text data, the requirements of the task, and the downstream applications such as sentiment analysis.
Tokenization Strategy: Understand the Task Requirements, Analyze the Text Data, Consider Language-specific Factors, Explore Tokenization Techniques[Word Tokenization, SubWord Tokenization, Character Tokenization, Customized Tokenization]

Impact of Tokenization on Sentiment Analysis:

Word Tokenization: Simple and straightforward. May struggle with out-of-vocabulary words or compound words where sentiment can be influenced by the whole phrase. May lose context from compound words or phrases. However, straightforward for most sentiment analysis tasks.

Subword Tokenization: Can handle unseen words better by breaking them down into subword units. Captures more nuanced meanings and can potentially improve sentiment analysis accuracy.

Character Tokenization: Preserves fine-grained information but requires models capable of understanding word-level semantics from character sequences.

Customized Tokenization: Designing tokenization rules specific to your task or domain.

Conclusion: The choice of tokenization strategy in NLP significantly impacts the performance of downstream tasks like sentiment analysis. It’s crucial to select a strategy that balances simplicity, coverage of language nuances, and effectiveness for the specific task at hand. Experimentation and evaluation are key to determining the optimal tokenization approach for your particular NLP application..

Big Picture Responses **#3 Explain the practical considerations and decision-making process when choosing between vectorizers and word embeddings for representing text data in NLP tasks? Discuss the benefits, limitations, and specific use cases of each approach, emphasizing how, why, and when you would opt for one over the other.**

Word Embedding: word representation in Natural Language Processing (NLP) that allows words to be represented as dense vectors of real numbers, where each word is mapped to a continuous vector space. These vectors capture semantic relationships and contextual meanings of words based on their usage in a large corpus of text.
One benefit of word embedding is Semantic Understanding: They capture semantic relationships between words, allowing models to understand similarities and differences in meaning.
another benefit is Generalization: Word embeddings can generalize to unseen words based on their embeddings, which is useful for handling out-of-vocabulary words in NLP tasks.
One limitation is Domain Specificity: Word embeddings trained on general corpora may not capture domain-specific nuances effectively.

Vectorizer: a technique or method used to convert textual data into numerical vectors that machine learning models can understand and process. The goal of vectorization in NLP is to transform text data from its raw form (sequences of words or characters) into a structured format that can be used as input to machine learning algorithms
One benefit of a vectorizer (count or tfid) is Interpretability: Depending on the vectorizer used, the resulting vectors can be interpretable (e.g., counts with CountVectorizer) which can aid in understanding feature importance.
One limitation of a vectorizer is Loss of Context: They typically do not capture semantic relationships between words or understand the meaning of words beyond their frequencies.

Big Picture Responses **#4 Explain what sequence-to-sequence models are and why they are better suited for NLP tasks over traditional neural networks.**

Sequence-to-sequence (Seq2Seq) models are a type of neural network architecture designed for tasks where the input and output are both sequences of data, such as machine translation, text summarization, and question answering. These models are particularly powerful in Natural Language Processing (NLP) because they can handle variable-length input and output sequences efficiently.

Seq2Seq models are well-suited for NLP tasks because they can handle variable-length sequences, capture contextual dependencies effectively, and perform end-to-end learning of complex mappings from input to output sequences. These capabilities make them superior to traditional neural networks for tasks where understanding and generating sequences of text data are central, such as machine translation, summarization, and dialogue systems.

Architecture: Encoder-Decoder Structure - Seq2Seq models consist of two main components: an encoder and a decoder.

Encoder: Takes the input sequence (e.g., a sentence in one language) and processes it into a fixed-size context vector or hidden state representation that captures the meaning of the input sequence.

Decoder: Takes the context vector from the encoder and generates the output sequence (e.g., a translated sentence in another language) one token at a time.

Example Applications:

Machine Translation: Translate sentences from one language to another.
Text Summarization: Generate a concise summary of a longer text.
Speech Recognition: Convert spoken language into text.
Dialogue Generation: Produce responses in a conversation.

Advantages Over Traditional Neural Networks:

Handling Variable-Length Input/Output: Traditional neural networks like feedforward or convolutional neural networks (CNNs) are designed for fixed-size input vectors. In NLP tasks, input sentences can vary in length, and output sequences may also vary in length (e.g., translations can have different word counts). Seq2Seq models naturally handle these variable-length sequences through their encoder-decoder structure.

Big Picture Responses **#5 Describe what a large language model (LLM) is and why it is needed.**

A type of artificial intelligence model designed to understand and generate human-like text based on vast amounts of data. These models are typically based on deep learning architectures, specifically transformer models, and are trained on massive datasets to learn the nuances of natural language.
Versatility and Adaptability:
LLMs can be applied to a wide range of NLP tasks without the need for task-specific architectures or extensive feature engineering. Their ability to generalize across tasks and domains makes them highly versatile and adaptable to various applications in natural language understanding and generation.

Improvement in NLP Benchmarks:
LLMs have significantly advanced the state-of-the-art in NLP benchmarks and tasks such as machine translation (e.g., Google Translate), text summarization (e.g., BERTSUM), and sentiment analysis (e.g., sentiment classifiers).

Applications in Real-World Scenarios:
In practical applications, LLMs can automate tasks that involve processing large volumes of text data, improving productivity and accuracy in areas such as **customer service**, **content generation**, and **information retrieval**.
They also enable the development of more advanced AI systems, including **virtual assistants and chatbots**, that can interact with users in natural language more effectively.

BREAK THROUGH TECH

Break

BREAK THROUGH TECH

Breakout Groups: Lab Assignment

Lab 8a

In this lab, you will implement the machine learning project plan of your choosing. You will be working in a Jupyter Notebook.

- Load your data set.
- Inspect and analyze the data.
- Prepare your data for your model.
- Fit your model to the training data and evaluate the model's performance.
- Improve the model's performance.
- Create a portfolio to showcase your project.

BREAK
THROUGH
TECH

Working Session 1 Debrief

Lab Debrief

So far,

- What did you enjoy about this lab?
- What did you find difficult about this lab?
- What questions do you still have about this lab?

BREAK
THROUGH
TECH

Working Session 2 Debrief

Lab Debrief

- What did you enjoy about this lab?
- What did you find hard about this lab?
- What questions do you still have about this lab?
- How did you approach problem-solving during the exercise?
- What would you do differently if you were to repeat the exercise?

BREAK
THROUGH
TECH

Concluding Remarks

Concluding Remarks

- Key takeaways
- Additional resources

Concluding Remarks

- Key takeaways
- Additional resources

Next week

In the following week, you will:

- Follow software development best practices
- Explore common ML failure modes
- Diagnose how data size contributes to execution bottlenecks
- Diagnose how feature issues contribute to degraded model performance
- Discuss societal failure mode
- Understand the sources of discriminatory bias and how to measure and mitigate them
- Improve the fairness and accountability of a model
- Continue implementing a project plan to solve a machine learning problem
- Create a portfolio for your project.

And in the lab, you will:

- Continue working on week 8 lab
- Create your portfolio

BREAK
THROUGH
TECH

Content + Lab Feedback Survey

Content + Lab Feedback Survey

To complete your lab, please answer the following questions about BOTH your online modules and your lab experience. Your input will help pay it forward to the Break Through Tech student community by enabling us to continuously improve the learning experience that we provide to our community.

Thank you for your thoughtful feedback!

https://forms.gle/eUQQZgS6BPRpqgZ7A