Natural Language Processing

Introduction to NLP

What is NLP

- NLP enables machines to understand, interpret, and respond to human language
- A foundation for AI systems like ChatGPT, search engines, and voice assistants

Why is NLP Important?

- We live in a world filled with unstructured text: Emails, social media, news, customer reviews
- NLP helps analyze and extract meaning from text

Real-World Applications of NLP

- Social media analysis (sentiment detection, trend recognition)
- Target audience classification
- Predictions of user behavior
- Applications in healthcare, finance, and education

Evolution of NLP

- 1980s: Rule-based systems
- 1990s – 2000s: Statistical methods
- Today: Deep learning & generative AI

Fundamentals of Text Preprocessing

- Tokenization
- Stop-word removal
- Stemming & lemmatization
- Essential for preparing raw text for analysis

Evaluating NLP Models

Key metrics:

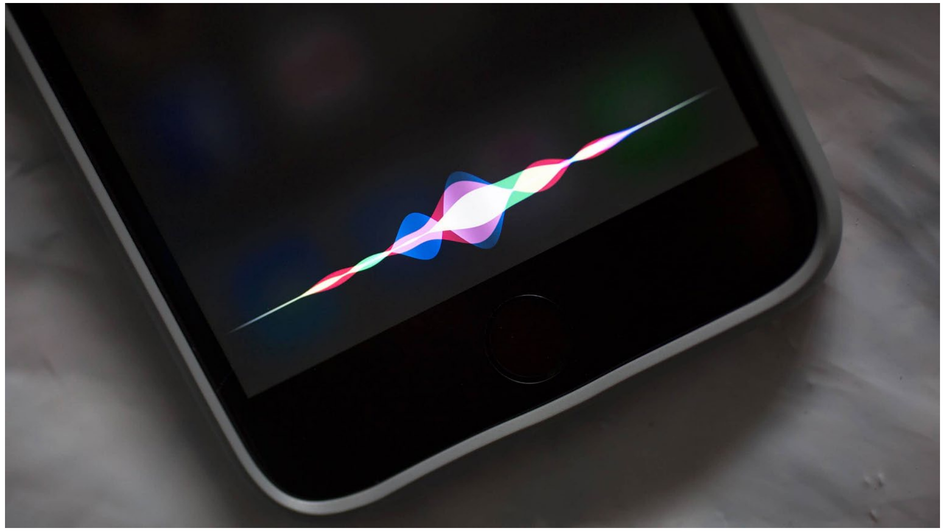
- Precision
- Recall
- F1-score

Leveraging Pre-trained Models

- Word2Vec, GloVe, FastText
- Saves time by using existing embeddings
- Focus on solving problems rather than reinventing the wheel

Natural Language Processing

What is NLP?



Google
Translate

So…what is NLP?

Natural Language Processing (NLP) is a field of artificial intelligence that enables computers to understand, interpret, and generate human language.

Key NLP Tasks

- Text Classification
- Sentiment Analysis
- Named Entity Recognition (NER)
- Machine Translation
- Text Generation

Example Text: *"I recently bought the Apple iPhone 15 in New York, and I absolutely love it! The battery life is fantastic, but the price was a bit steep."*

Text Classification

Task: Classify the review into predefined categories, such as “Positive,” “Negative,” or “Neutral.”

Output: “Positive”

Example Text: *"I recently bought the Apple iPhone 15 in New York, and I absolutely love it! The battery life is fantastic, but the price was a bit steep."*

Sentiment Analysis

Task: Determine the sentiment expressed in the review.

Output:

Overall Sentiment: **Positive**

Sentiment Breakdown:

“I absolutely love it!” → Positive

“The price was a bit steep.” → Negative

Example Text: *"I recently bought the Apple iPhone 15 in New York, and I absolutely love it! The battery life is fantastic, but the price was a bit steep."*

Named Entity Recognition (NER)

Task: Identify and extract named entities such as products, locations, and organizations.

Output:

Entities:

- Product: Apple iPhone 15
- Location: New York
- Organization: Apple

Example Text: *"I recently bought the Apple iPhone 15 in New York, and I absolutely love it! The battery life is fantastic, but the price was a bit steep."*

Machine Translation

Task: Translate the review into another language, such as Spanish.

Output: "Recientemente compré el Apple iPhone 15 en Nueva York, ¡y me encanta! La duración de la batería es fantástica, pero el precio fue un poco alto."

Example Text: *"I recently bought the Apple iPhone 15 in New York, and I absolutely love it! The battery life is fantastic, but the price was a bit steep."*

Text Classification

Task: Generate a summary or rephrase the review.

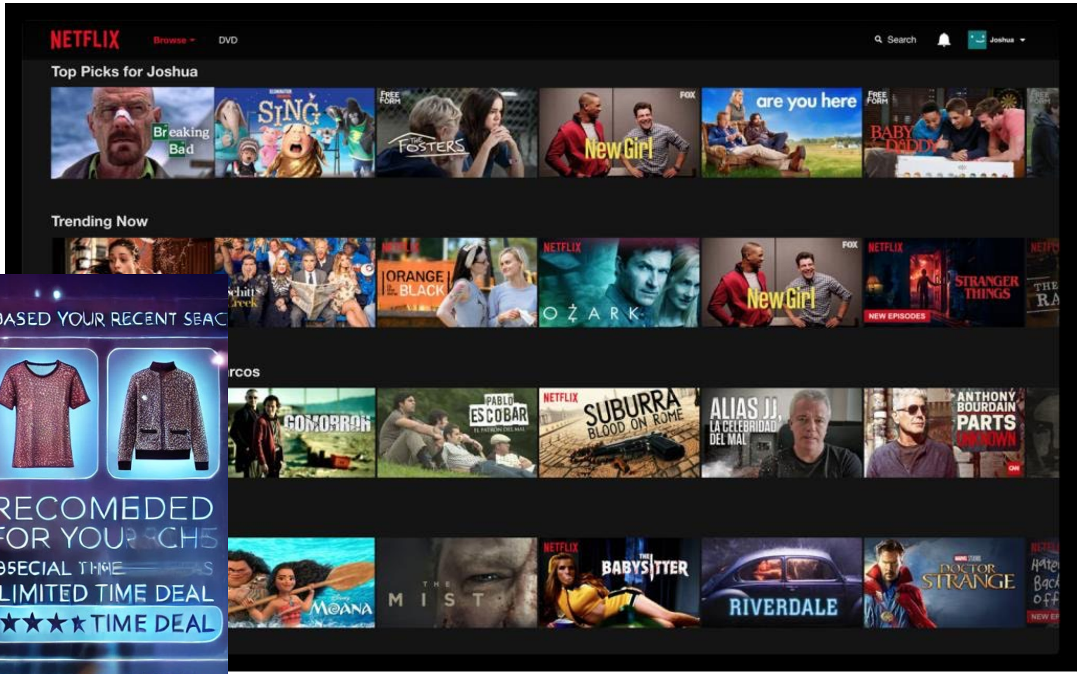
Output: "Loved the Apple iPhone 15 for its fantastic battery life, though the price was high. Purchased in New York."

Example Text: *"I recently bought the Apple iPhone 15 in New York, and I absolutely love it! The battery life is fantastic, but the price was a bit steep."*

Applications of NLP

- Virtual Assistants and Chatbots
- Search Engines
- Sentiment Analysis for Business Insights
- Text Summarization
- Healthcare Applications

Real-World Examples of NLP



NLP is not just about convenience—it's a cornerstone of modern AI applications.

Summary

- NLP is the bridge between human communication and computational systems
- Key NLP tasks
- Generative AI integrates NLP for chatbots, creative writing, and more.
- What are some examples of NLP you've seen in your daily life?


GILMAN HALL

Natural Language Processing

History of NLP

1980s – 2000s Symbolic AI

Example Rules

1. **Noun Identification Rules:**
   - If a word ends with **-ion**, **-ment**, **-ness**, or **-ity**, classify it as a noun.
   - If the word is preceded by an article (e.g., *a*, *an*, *the*), classify it as a noun.
   - Words appearing in a predefined **noun dictionary** (e.g., *dog*, *computer*, *happiness*) are classified as nouns.
2. **Verb Identification Rules:**
   - If a word ends with **-ing**, **-ed**, or **-s**, classify it as a verb.
   - If a word follows a pronoun (e.g., *I*, *he*, *she*, *we*), classify it as a verb.
   - Words appearing in a predefined **verb dictionary** (e.g., *run*, *play*, *learn*) are classified as verbs.

"The dog is running happily in the park."

1980s – 2000s Symbolic AI

Processing Steps

- **Tokenize the Sentence:**
  - ["The", "dog", "is", "running", "happily", "in", "the", "park"]
- **Apply Noun Identification Rules:**
  - "The" → Preceding article, next word "dog" → **Noun**
  - "Park" → Preceding article, next word "the" → **Noun**
- **Apply Verb Identification Rules:**
  - "Is" → Verb dictionary → **Verb**
  - "Running" → Ends with **-ing** → **Verb**
- **Output:**

  Nouns: ["dog", "park"] **Verbs:** ["is", "running"]

"The dog is running happily in the park."

Advantages & Limitations

Advantages of Rule-Based Systems:

- Easy to understand and implement.
- Highly accurate for simple, well-defined linguistic tasks.

Limitations:

- Struggle with exceptions to rules
  (e.g., "building" could be a noun or a verb depending on context).
- Require significant manual effort to create and maintain rules.
- Perform poorly with ambiguous or complex sentences.

"He is building a house,"

2000s Statistical Models

Bag of Words (BoW) Example:

- Step 1: Tokenize the Sentence
  Tokens: ["the", "dog", "is", "running", "happily", "in", "the", "park"]
- Step 2: Create a Vocabulary
  Vocabulary: ["the", "dog", "is", "running", "happily", "in", "park"]
- Step 3: Count Word Occurrences
  Word count: *the*=2, *dog*=1, *is*=1, ..., *park*=1
- Step 4: Represent as a Numeric Vector
  Vector Representation: [2, 1, 1, 1, 1, 1, 1]

"The dog is running happily in the park."

JOHNS HOPKINS WHITING SCHOOL of ENGINEERING

Great Learning

Advantages & Limitations

- BoW model captures the frequency of words
- Does **not** preserve word order or semantics

"The dog is running happily in the park."	=	*"Happily running in the park is the dog."*

Advantages of Bag of Words:

- Simple and easy to implement.
- Works well for basic text classification tasks, like spam detection.

Limitations:

- Loss of context
- High dimensionality
- Semantic limitations

Essential concept in NLP

representation of text in numerical format

2010s Sequential Processing with NN

Recurrent Neural Network Example:

- Step 1: Input Representation

  How are embeddings determined?

  - Word2Vec – learns embedding based on context *Continuous BoW* or *Skip-Gram*
  - GloVe – global co-occurrence in corpus
  - BERT or GPT – based on meaning in sentence
  - Pre-Trained Embeddings vs. Learned During Training

"The dog is running happily in the park."

Example Word Embeddings:

- "The" → [0.1, 0.2, 0.3]
- "dog" → [0.4, 0.5, 0.6]
- "is" → [0.7, 0.8, 0.9]
- "running" → [0.2, 0.3, 0.4]
- "happily" → [0.5, 0.6, 0.7]
- "in" → [0.8, 0.9, 0.1]
- "the" → [0.1, 0.2, 0.3]
- "park" → [0.3, 0.4, 0.5]

2010s Sequential Processing with NN

Recurrent Neural Network Example:

- Step 1: Input Representation
  - Initialization
  - Update During Training
  - Final Representation
- Step 2: Process Each Word Sequentially

1. **Input Word:** "The"
   - Compute new hidden state: $h_1 = f(W \cdot x_{the} + U \cdot h_0)$
   - $h_1 = [0.15, 0.25, 0.35]$
2. **Input Word:** "dog"
   - Compute new hidden state: $h_2 = f(W \cdot x_{dog} + U \cdot h_1)$
   - $h_2 = [0.4, 0.6, 0.7]$
3. **Input Word:** "is"
   - Compute new hidden state: $h_3 = [0.6, 0.7, 0.8]$

Example Word Embeddings:
- "The" → [0.1, 0.2, 0.3]
- "dog" → [0.4, 0.5, 0.6]
- "is" → [0.7, 0.8, 0.9]
- "running" → [0.2, 0.3, 0.4]
- "happily" → [0.5, 0.6, 0.7]
- "in" → [0.8, 0.9, 0.1]
- "the" → [0.1, 0.2, 0.3]
- "park" → [0.3, 0.4, 0.5]

"The dog is running happily in the park."

2010s Sequential Processing with NN

Recurrent Neural Network Example:

- Step 3: Prediction

  At each step, RNN uses hidden state to predict the **next word** in the sequence.

  After “The dog is running…” predicts “happily”

  RNN predicts next word after each step, generating the sentence sequentially

"The dog is running happily in the park."

- Example Word Embeddings:
  - "The" → [0.1, 0.2, 0.3]
  - "dog" → [0.4, 0.5, 0.6]
  - "is" → [0.7, 0.8, 0.9]
  - "running" → [0.2, 0.3, 0.4]
  - "happily" → [0.5, 0.6, 0.7]
  - "in" → [0.8, 0.9, 0.1]
  - "the" → [0.1, 0.2, 0.3]
  - "park" → [0.3, 0.4, 0.5]

Advantages & Limitations

Advantages of RNNs:

- Sequential Context
- Handles Variable-Length Input

Limitations:

- Vanishing Gradient Problem
- Sequential Bottleneck

Limitations led to the development of the Transformer!

2017 Transformers

- Google introduced the **transformer** in 2017
- Transformer models weigh the importance (attention) of different words
- Attention models have advantages:
  - Parallelization
  - Memory efficient
  - Adaptive focus/weighting
  - Scalable to input/output length

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com
Noam Shazeer* Google Brain noam@google.com
Niki Parmar* Google Research nikip@google.com
Jakob Uszkoreit* Google Research usz@google.com
Llion Jones* Google Research llion@google.com
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu
Łukasz Kaiser* Google Brain lukaszkaiser@google.com
Illia Polosukhin* ‡ illia.polosukhin@gmail.com

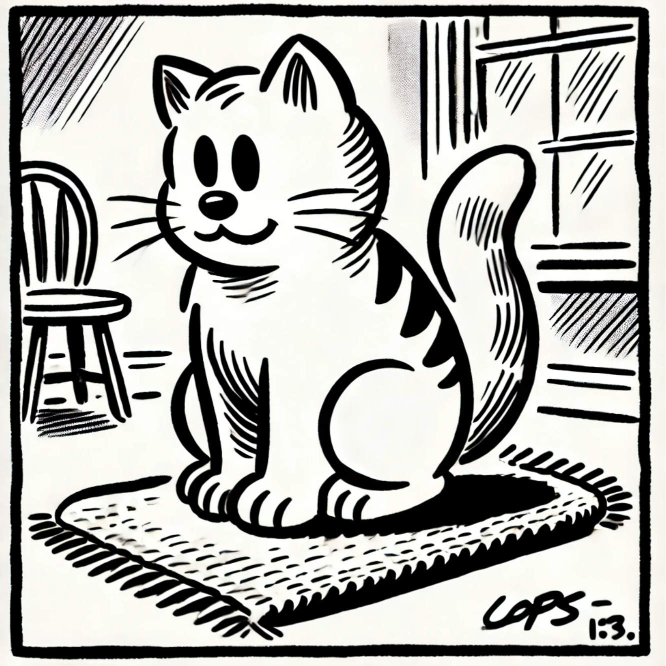
Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.0 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention Is All You Need. In *Advances in Neural Information Processing Systems* (pp. 5998-6008).

Attention is all you need

The	cat	sat	on	the	mat.



- **Contextual Understanding**: considers the context of each word by neighboring words
- **Parallel Processing**: can process words all at once instead of sequentially
- **Dynamic Weighting**: Weigh words based on relevance.

Self-Attention and Transformers (1)

- Different contextual words are *attended to* when encoding different meanings:

a spherical object

The	dog	chased	after	the	ball	as	it	rolled	down	the
Cinderella	went	to	the	ball	in	a	fancy	glass	coach	

a formal dance party

The	dog	chased	after	the	ball	as	it	rolled	down	the
Cinderella	went	to	the	ball	in	a	fancy	glass	coach	

Advantages & Limitations

Advantages of Transformers:

- Parallel processing
- Attention mechanism

Limitations:

- Large in size
- High compute / energy consumption

"He is building a house,"

2018 Generative AI

- **OpenAI** introduced the first **Generative Pre-trained Transformer (GPT)** model in 2018.
- Pre-training and Fine-tuning
- Contextual Understanding
- Each iteration of GPT improved on the last
  - GPT 2 (2019)
  - GPT 3 (2020)
  - GPT 4 (2023)
  - GPT 4o3 and 4o3-mini (2025)

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention Is All You Need. In *Advances in Neural Information Processing Systems* (pp. 5998-6008).

Summary

- Early NLP began with symbolic AI and transitioned to statistical AI.
- RNNs allowed sequential data processing.
- Transformers overcame challenges with RNNs.
- GPT models harness the power of transformers to redefine NLP.

JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Great Learning

Natural Language Processing

Applications in NLP

Applications of NLP

Twitter/X Example

"tweet": [

"I love the new iPhone! It's fantastic. #Apple",

"The service at this restaurant was terrible. Never going back. #Disappointed",

"Tesla's new model is groundbreaking! #Innovation",

"I had an average experience with the product. It's okay. #Neutral",

]

Text Classification

Text classification involves categorizing text into predefined categories. It's a cornerstone for spam detection, news categorization, and even support ticket triaging.

Examples:

- Email spam detection
- Customer support

Sentiment Analysis

Sentiment analysis determines the sentiment expressed in a text, whether positive, negative, or neutral.

Examples:

- Brand monitoring
- Customer feedback

Named Entity Recognition (NER)

NER identifies entities such as names, locations, dates, and organizations in text.

Examples:

- Healthcare
- News aggregation

Parts-of-Speech (PoS) Tagging

POS tagging identifies grammatical parts of speech (e.g., nouns, verbs) in text.

Examples:

- Grammar checkers
- Chatbots

Machine Language Translation

Machine translation automatically translates text from one language to another.

Examples:

- Global communication
- E-commerce

Twitter/X Example

Twitter Example

```
"tweet": [
    "I love the new iPhone! It's fantastic. #Apple",
    "The service at this restaurant was terrible. Never going back. #Disappointed",
    "Tesla's new model is groundbreaking! #Innovation",
    "I had an average experience with the product. It's okay.        #Neutral",
]
```

Summary

- Classification
- Sentiment Analysis
- Named Entity Recognition (NER)
- Parts-of-Speech (PoS) Tagging
- Machine Language Translation


GILMAN HALL

Natural Language Processing

Text Pre-Processing

Vocabulary

What is Vocabulary in NLP?

- Set of unique words (or tokens) in the text dataset
- Foundation for converting text to numeric representation

Why is Vocabulary Important?

- Richness of understanding
- Efficient representation
- Focus on meaningful data

Step 1: Tokenization

- Words: e.g., "The dog is running" → ["The", "dog", "is", "running"]
- Subwords: e.g., "running" → ["run", "ning"]
- Characters: e.g., "run" → ["r", "u", "n"]

What Tokenization Matters?

- Building blocks for subsequent steps
- Effective by isolating most meaningful components

Step 2: Remove Stop Words

What are Stop Words?

- Frequent words with little semantic meaning: e.g. “the”, “is”, “and”
- Reduces noise allowing model to focus on meaningful terms

Why Remove Stop Words?

- Improved efficiency
- Focus on important words

Step 3: Standardize (Stemming/Lemmatization)

Stemming

- Reduces words to their root form by removing suffixes and prefixes.
- Example: “running,” “runner,” “runs” → “run”
- Simple but sometimes produces non-standard forms (e.g., "studies" → "studi").

Lemmatization

- Converts words to their base or dictionary form using linguistic rules.
- Example: “better” → “good”; “running” → “run”
- More accurate than stemming but computationally heavier.

Why Standardize?

- Reduces vocabulary size to improve performance

Step 4: Embeddings

- Serve as the bridge between raw text and machine-readable formats.

- Techniques like Word2Vec, GloVe, or contextual embeddings like BERT are often applied after the earlier pre-processing steps.

Summary

- Tokenization
- Stop Word Removal
- Lemmatization
- Embeddings


GILMAN HALL

Natural Language Processing

Word Embeddings

Word embeddings are dense vector representations of words in a continuous vector space.

Word2Vec

Word2Vec, introduced by Google in 2013

1. **Skip-gram Model:**
   - Predicts the context words given a target word.
   - Example: Given the word "dog," the model predicts words like "barks," "walks," or "puppy."
   - Useful for capturing rare words because it focuses on word pairs that co-occur.
2. **Continuous Bag of Words (CBOW):**
   - Predicts a target word given the context words.
   - Example: Given the context "The ___ is barking," predicts "dog."
   - Faster to train and works well for frequent words.

GloVe

GloVe was introduced by Stanford in 2014.

- It combines **global statistics** (how often words appear together in the corpus) with **local context** (how words are used in sentences).
- For example, words like "ice" and "cold" might often co-occur, whereas "ice" and "fire" are less likely, leading to embeddings that reflect these relationships.

FastText

FastText, developed by Facebook

1. Instead of treating words as atomic units, FastText breaks them into character n-grams (subwords).
   Example: The word "apple" is represented as the n-grams "app," "ppl," and "ple."

2. This approach handles **Out-of-Vocabulary (OOV)** words effectively. For instance, even if the word "university" wasn't seen during training, the model can infer its embedding from the subwords "uni," "vers," and "ity."

Comparison

Technique	Strengths	Limitations
Word2Vec	Captures word analogies and is computationally efficient.	Struggles with OOV words and ignores subword info.
GloVe	Combines global and local context for better global semantic representation.	Pre-training required; less flexible for new data.
FastText	Handles OOV words and morphologically rich languages well.	Larger models due to subword representations.

Summary

- Embeddings like Word2Vec, GloVe, FastText represent words, capturing semantic meaning.
- Each technique has strengths suited to specific use-cases.
- Demonstrated a use-case in Python.


GILMAN HALL

Natural Language Processing

Evaluation Metrics for NLP

Evaluation Metrics for NLP Tasks

- N-grams for text-to-text tasks.
- AI/ML classification metrics like accuracy, precision, recall, F1, etc.
- Precision and F1 for imbalanced datasets.

N-Grams

What are N-Grams? N-grams are contiguous sequences of n words in a text.
For example:

- Unigram: "The" (single word)
- Bigram: "The cat"
- Trigram: "The cat sat"

Why N-Grams Matter? N-grams are used to measure the overlap between the generated text and a reference text. The more overlap, the better the model is at mimicking human-like or reference outputs.

N-Grams

Common Metrics Using N-Grams

- **BLEU (Bilingual Evaluation Understudy)**: A metric that compares n-grams in the generated and reference text, rewarding matches and penalizing length mismatches.

- **ROUGE (Recall-Oriented Understudy for Gisting Evaluation)**: Focuses on recall by comparing the overlap between reference and generated text.

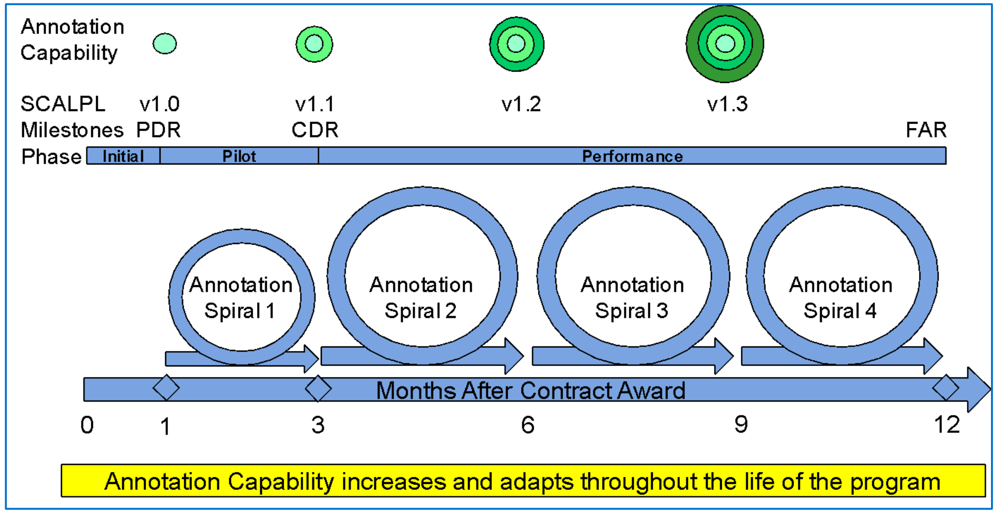
Classification Metrics

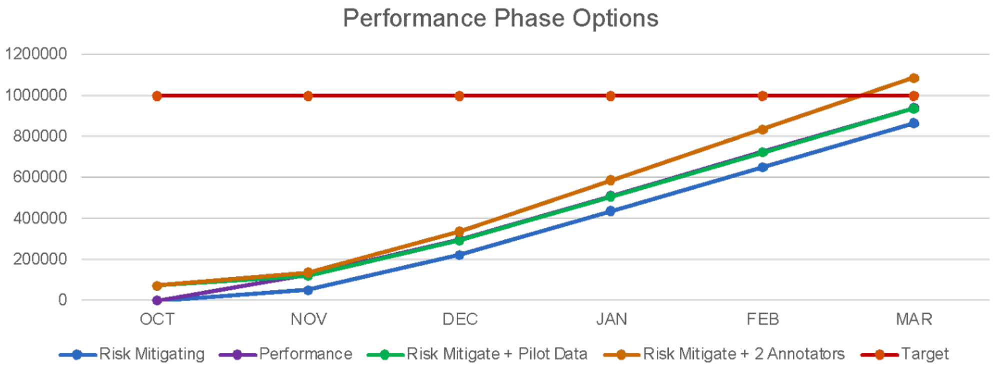
- Accuracy – Proportion of correctly matched samples.
- Precision – Proportion of correctly predicted instances. *I only flag true spam.*
- Recall – How many actual positive instances identified. *I found all of the spam.*
- F1 – Harmonic mean of precision and recall.
  Strikes a balance, especially for imbalanced datasets.

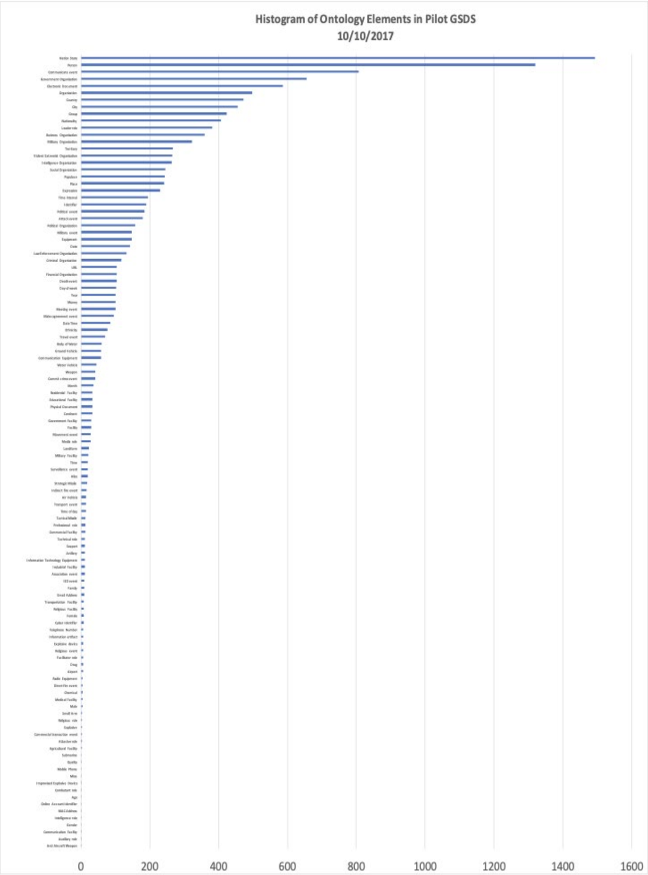
JOHNS HOPKINS
WHITING SCHOOL of ENGINEERING

Great Learning

Data Labeling Challenges







Summary

- N-Grams
- Classification Metrics
- Data Labeling Challenges and Imbalanced Data
- Precision and Recall Trade-offs


JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING
Great Learning
GILMAN HALL