
Grace Hopper
1906-1992
Annie Easley
1933-2011
together
we
build
/ ANITA
B.ORG
20th
GRACE HOPPER CELEBRATION
VIRTUAL

ANITA B.ORG 20th

GRACE HOPPER CELEBRATION

VIRTUAL

Towards Transparent AI:

Understanding Text-Based Model Predictions

Janhavi Mahajan and Ehi Nosakhare

together we build artificial intelligence

ANITA B.ORG 20th
GRACE HOPPER CELEBRATION
VIRTUAL

Janhavi Mahajan

Software Engineer

Microsoft

Ehi Nosakhare, PhD

Data Science Manager

Microsoft

Tutorial Goal

Learn how to interpret sentiment analysis results

After this tutorial you will know:

1. How to use the interpret-text package to understand state-of-art model in sentiment analysis
2. Get familiar with different techniques in interpret-text python package
3. Resources for building transparent AI applications

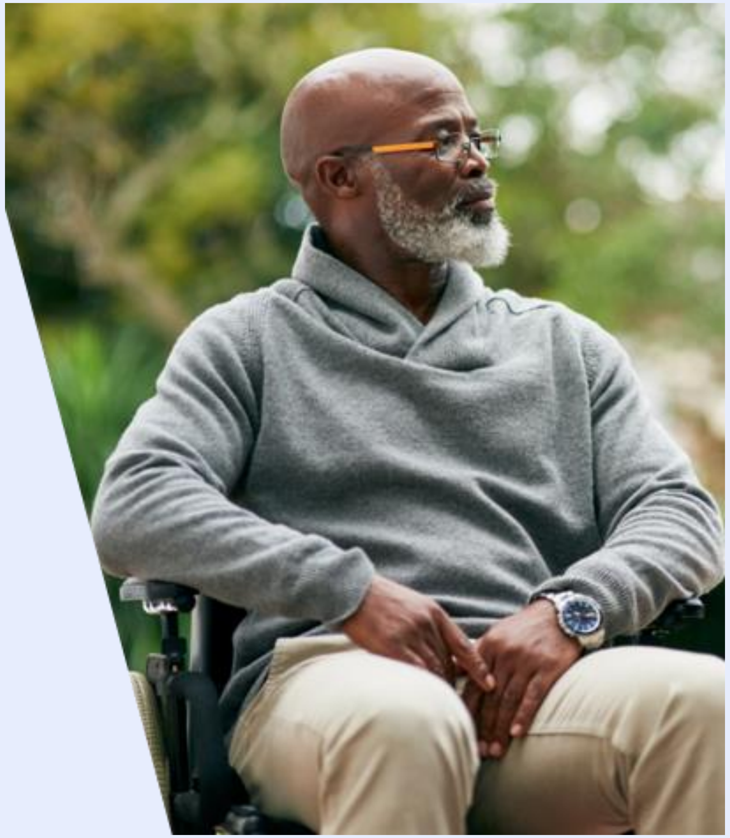
Setup Instructions

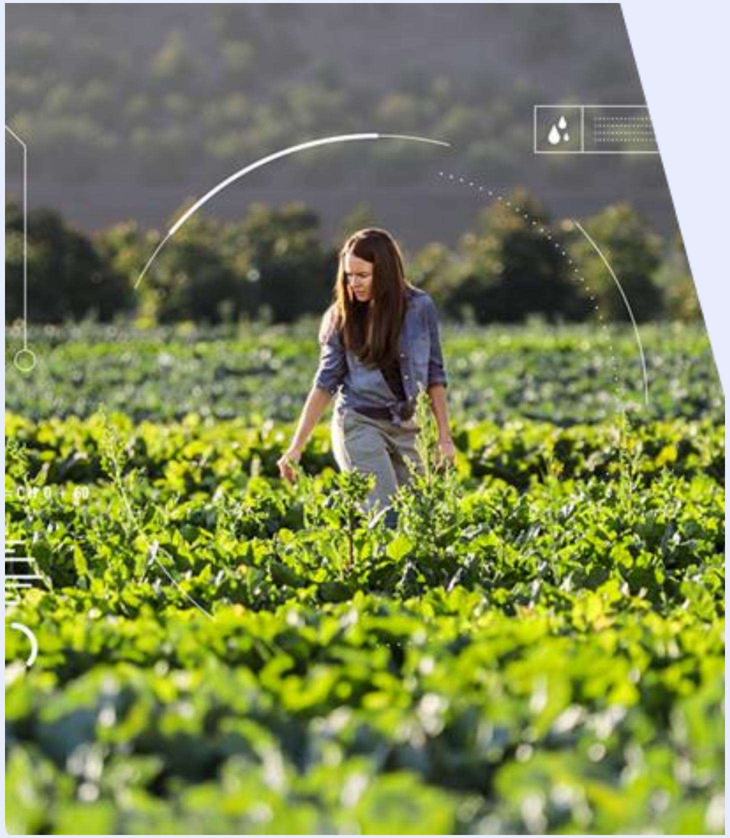
Go to: https://github.com/janhavi13/TowardsTransaparentAI_vGHC2020

Agenda

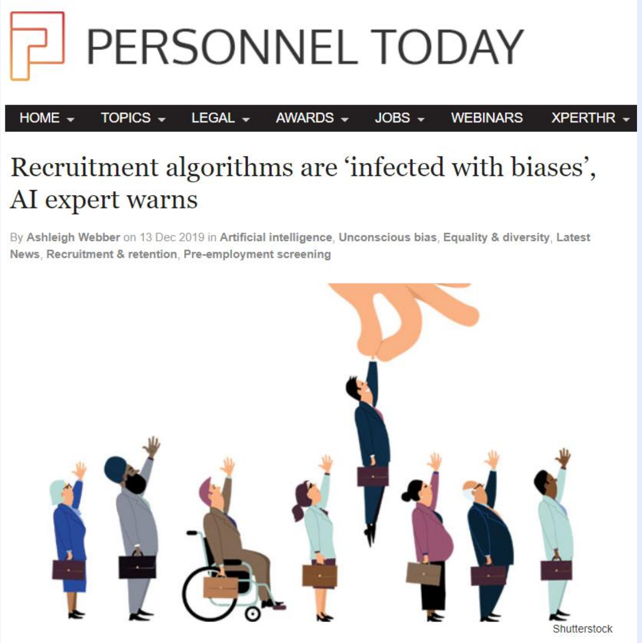
- Why Responsible AI?
- Introduce Natural Language Processing
- Interpretability for Text
- Getting Started with Interpret-Text
- The Three explainers
- Hands on tutorial for interpretability in a sentiment analysis task

AI will have a considerable impact on business and society as a whole





AI impact raises a host of complex and challenging questions

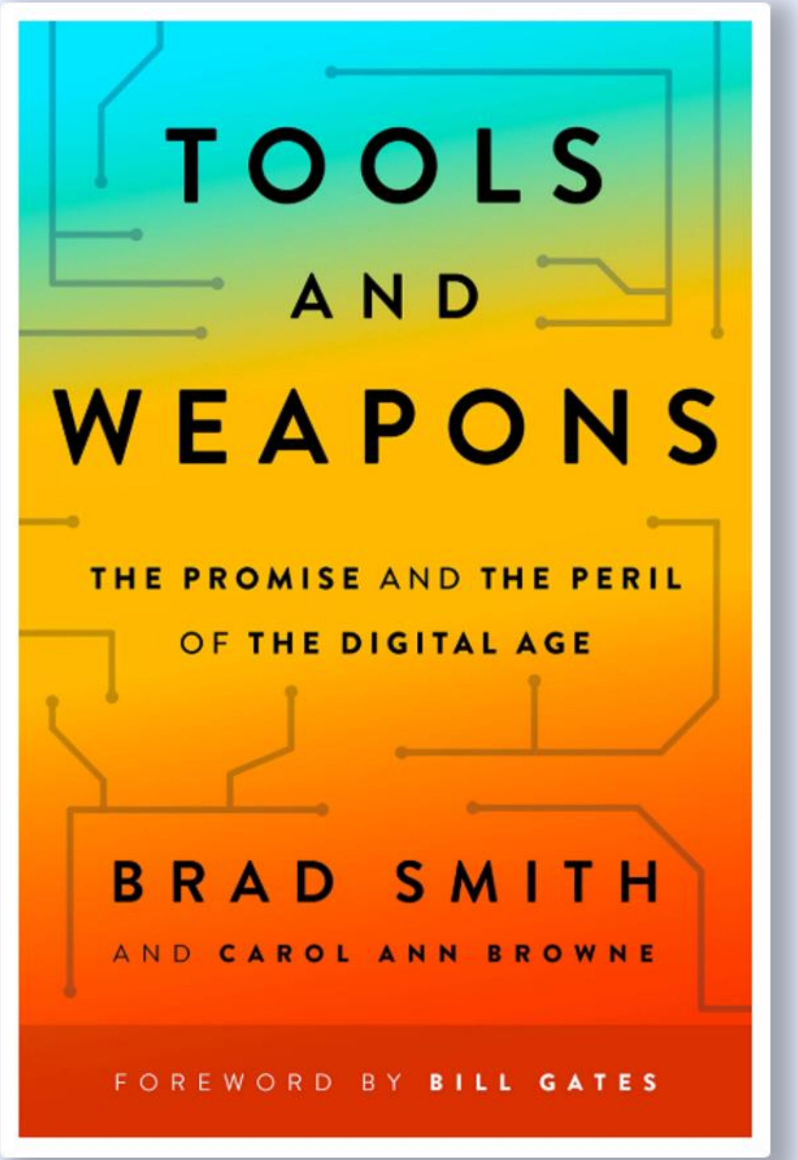


Automated Recruiting

Criminal Justice System

Financial Industry

"When your technology changes the world, you bear a responsibility to help address the world you have helped create."



Responsible AI Principles

- Fairness
- Reliability & Safety
- Privacy & Security
- Inclusiveness
- Transparency
- Accountability

Source: https://www.microsoft.com/en-us/ai/responsible-ai

Responsible AI Principles

Fairness

Reliability & Safety

Privacy & Security

Inclusiveness

Transparency

Accountability

Transparency

AI systems should have algorithmic interpretability

Interpretable Machine Learning refers to methods and models that make the behavior and predictions of machine learning systems understandable to humans.

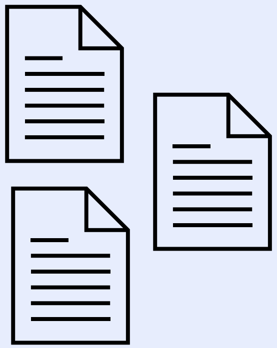
Natural Language Processing

Field of computer science, artificial intelligence, and computational linguistics concerned with the interactions between computers and human (natural) languages[1]

Growing in popularity with applications in topic classification, sentiment analysis, and entailment in a wide variety of business scenarios.

[1]https://github.com/microsoft/nlp-recipes

Classical NLP Pipeline



Pre-processing

- Tokenization
- Normalization
- Noise removal
- .....

Modeling

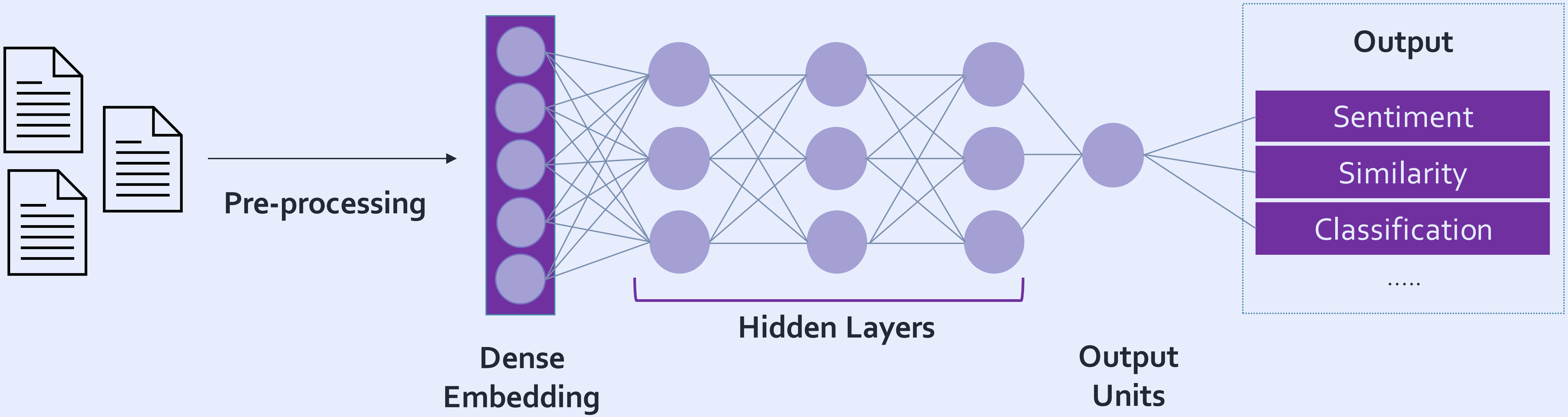
- Logistic Regression
- Random Forest
- LightGBM
- .....

Output

- Sentiment
- Similarity
- Classification

.....

Deep learning NLP Pipeline



Interpretability Tools

Interpret-Text

- Can explain ML models locally for each document
- Incorporates innovative text interpretability techniques
- Provides an interactive visualization dashboard
- Community can further expand its offerings

Facebook's Captum

- Supports a wide range of explanation methods and scenarios
- Visualization offerings for text methods
- Requires neural networks in Pytorch

AllenNLPInterpret

- Comprehensive coverage of scenarios.
- Gradient based saliency methods
- Requires neural networks in Pytorch
- Supports two Adversarial attacks and attention maps

Others:
- INNvestigate – Similar to captum
- Tf-explain, IBM-AIX360 – Does not support text use cases
- Layerwise relevance propagation toolbox (LRP)
- VisBERT, BertVIS– Specifically visualize inner workings of BERT

Interpret-text

- We have implemented **one classical** and **two state-of-the-art explainers** for text classification scenario to cover both NLP approaches.

  - Classical Text Explainer (glass-box)

  - Unified Information Explainer (post-hoc and model agnostic)

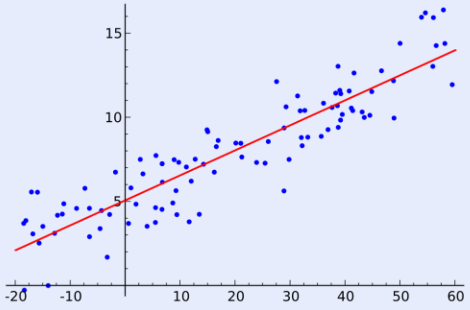
  - Introspective Rationale Explainer (plug-in during training, model agnostic)

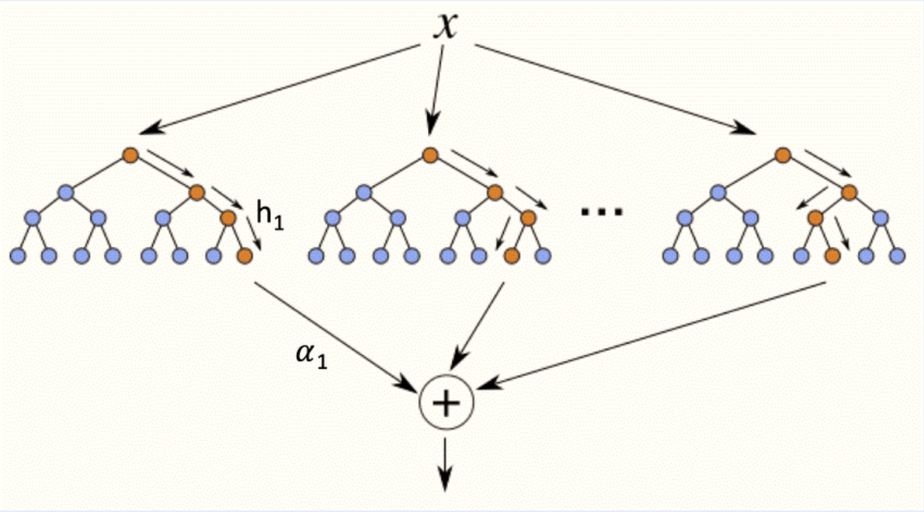
https://github.com/interpretml/interpret-text

Classical Text Explainer

- Modular API
- Handles text preprocessing, encoding, training, and hyperparameter tuning
- Inherently interpretable models
- Compatible with scikit-learn's linear and tree-based ensemble models

Default Configuration: 1-gram bag-of-words + scikit-learn count vectorizer + Logistic regression

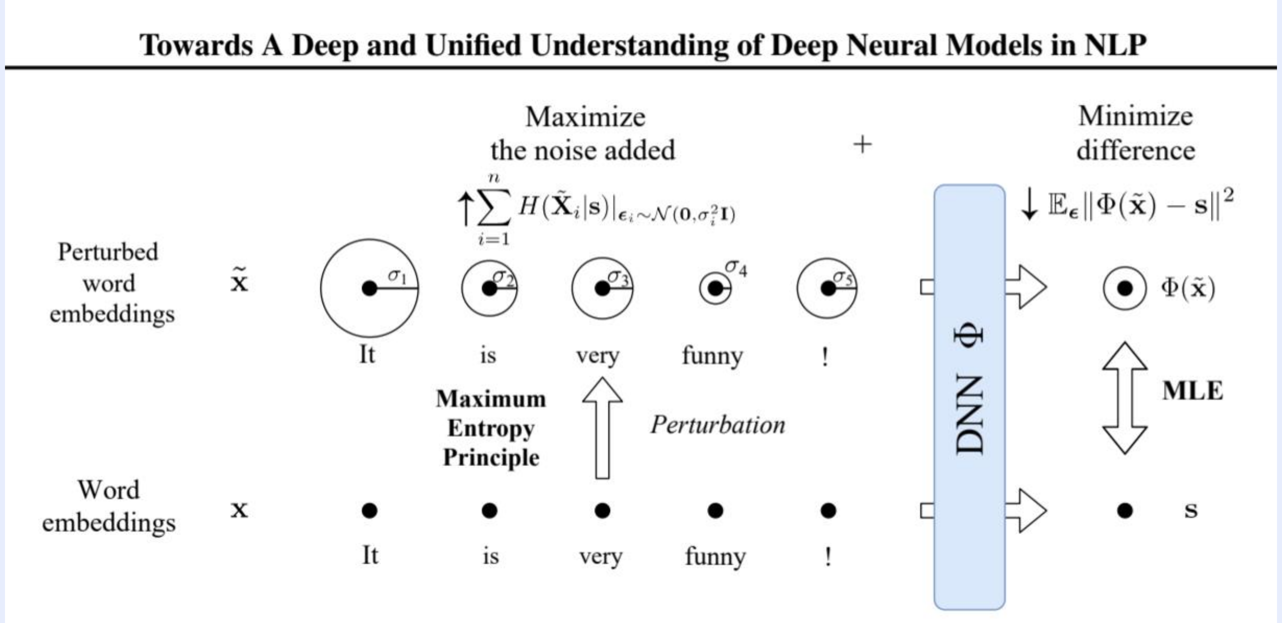




Unified Information Explainer

- Mutual Information based post-hoc interpretability model
- Provides unified and coherent explanations on intermediate layers of a variety of DNNs
- BERT is currently implemented
- Future work will extend API to LSTM and RNN

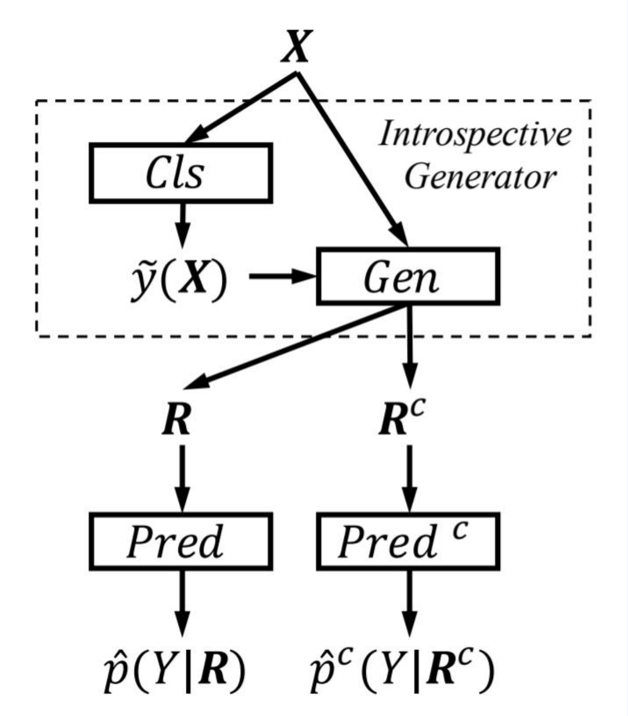
[Towards A Deep and Unified Understanding of Deep Neural Models in NLP, Guan et al. [ICML 2019]]



Introspective Rationale Explainer

- End-to-end training routine
- Incorporates an introspective generator as a pre-processing step
- Divides input text into rationales and anti-rationales
- Training routine maximizes the model's accuracy only using the rationales
- Since the model only sees the rationale, strong guarantees are provided on what's important

[Rethinking Cooperative Rationalization: Introspective Extraction and Complement Control, Yu et al. [EMNLP 2019]]



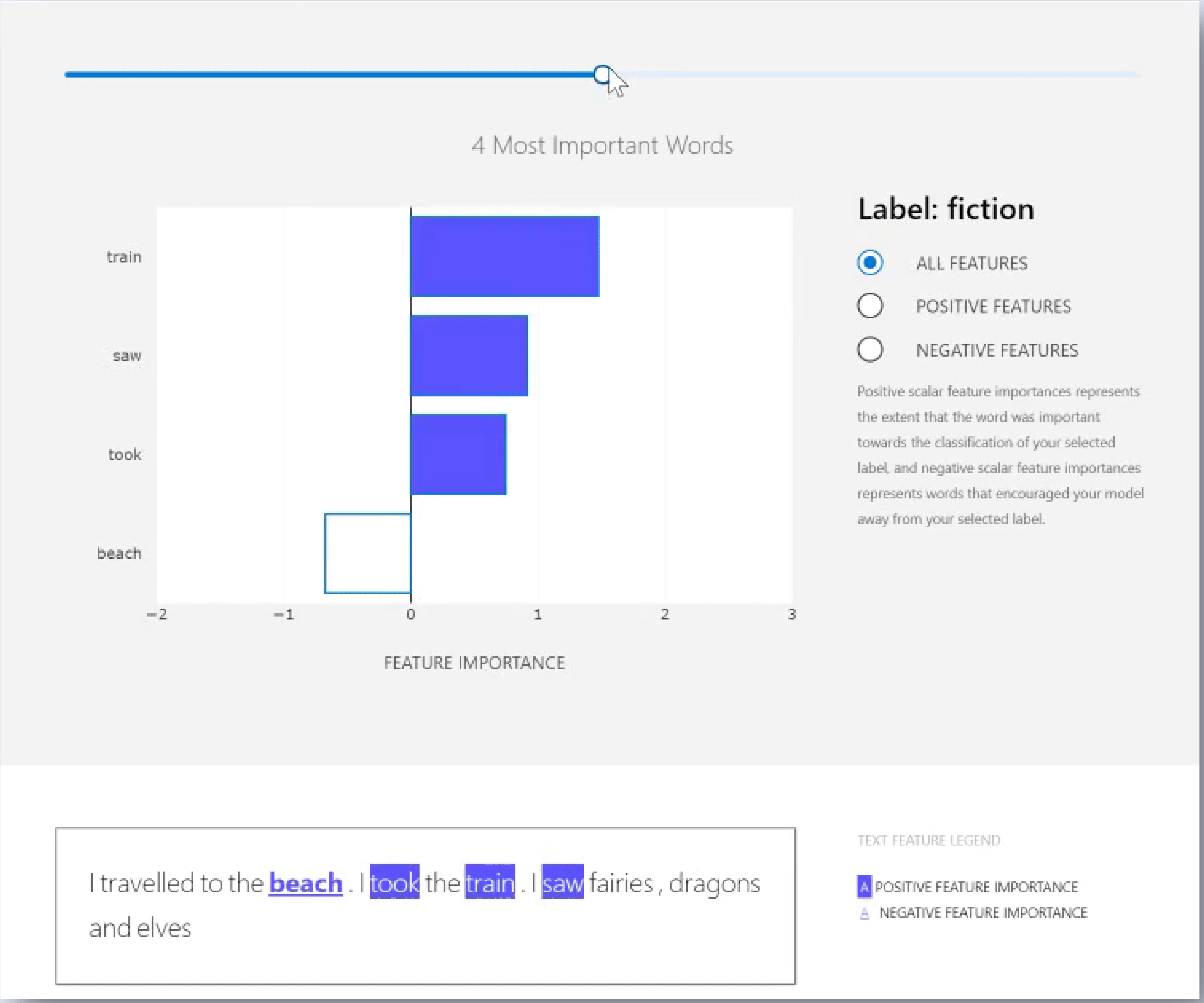
Comparing Explanation Methods

	Classical Text Explainer	Unified Information Explainer	Introspective Rationale Explainer
Input model support	Scikit-learn linear models and tree-based models	PyTorch	PyTorch
Explain BERT	No	Yes	Yes
Explain RNN	No	No	Yes
NLP Pipeline Support	Handles text pre-processing, encoding, training, hyperparameter tuning	Uses BERT tokenizer however user needs to supply trained/fine-tuned BERT model, and samples of trained data	Generator and predictor modules handle the required text pre-processing.

InterpretML

Interpretability for Text Data

https://github.com/interpretml/interpret-text



together we build artificial intelligence

InterpretML

We welcome contributions – if you have further questions and thoughts please reach out!

https://github.com/interpretml/interpret-text

Responsible ML Resources

Microsoft Responsible AI Resource Center
https://aka.ms/RAIresources

InterpretML
https://github.com/interpretml
https://aka.ms//InterpretMLWhitepaper
https://docs.microsoft.com/azure/machine-learning/how-to-machine-learning-interpretability

Thank You

@janhavi_m

/in/janhavimahajan

@_ehinosa

/in/ehinosa