

# Introducing gobbli

Deep learning with text doesn't have to be scary

Jason Nance  
Data Scientist  
RTI International



# Using a State of the Art Model for Text Classification

Line Count: ~~0~~ ~~101~~ ~~108~~ ~~142~~ ~~167~~ 187

**BERT Evaluation**

Evaluate the performance of Google's BERT model.

```
In [21]: import json as jq
import sys
import numpy as np
import time as tm
from collections import defaultdict
from data_loader import DataLoader

def preprocess_text(text, max_length):
    return text[:max_length]
```

ID	Label	Score	Confidence
0	0	0.000000	0.000000
1	1	0.000000	0.000000
2	1	0.000000	0.000000
3	1	0.000000	0.000000
4	1	0.000000	0.000000

```
In [23]: def eval_threshold(threshold):
    preds = []
    for i, (text, label) in enumerate(data_loader.train_loader.iter_instances()):
        pred = model.predict(text)
        if pred == label:
            preds.append(1)
        else:
            preds.append(0)
    return sum(preds) / len(preds)
```

```
In [24]: for threshold in (0.85, 0.9, 0.95, 0.97, 0.98, 0.99, 1.0):
    eval_threshold(threshold)
```

**Data Export Script**

```
#!/bin/bash
set -e

# Parameters
DATA_PATH="/path/to/data"
EXPORT_PATH="/path/to/export"

# Export data
python export_data.py $DATA_PATH $EXPORT_PATH
```

**Prediction Shell Script**

```
#!/bin/bash
set -e

# Parameters
MODEL_PATH="/path/to/model"
TEXT_PATH="/path/to/text"

# Run prediction
python predict.py $MODEL_PATH $TEXT_PATH
```

**Training Shell Script**

```
#!/bin/bash
set -e

# Parameters
TRAIN_PATH="/path/to/train"
VALID_PATH="/path/to/valid"
TEST_PATH="/path/to/test"

# Train model
python train.py $TRAIN_PATH $VALID_PATH $TEST_PATH
```

# Deep Learning: State of the Art

## Best Scores on DBpedia Classification Benchmark\*

- 2015: Char-level CNN (<https://arxiv.org/abs/1509.01626v3>)
- 2016: fastText (<https://arxiv.org/abs/1607.01759v3>), CNN/LSTM (<https://arxiv.org/abs/1602.02373v2>)
- 2017: DPCNN (<https://www.aclweb.org/anthology/papers/P/P17/P17-1052/>), M-ACNN (<https://arxiv.org/abs/1709.08294v3>)
- 2018: ULMFiT (<https://arxiv.org/abs/1801.06146v5>)
- 2019: BERT (<https://arxiv.org/abs/1810.04805>), MT-DNN (<https://arxiv.org/abs/1901.11504>), XLNet (<https://arxiv.org/abs/1906.08237>)

\* <https://paperswithcode.com/sota/text-classification-on-dbpedia>

# The Problem: Hard-Coding for Benchmark Problems

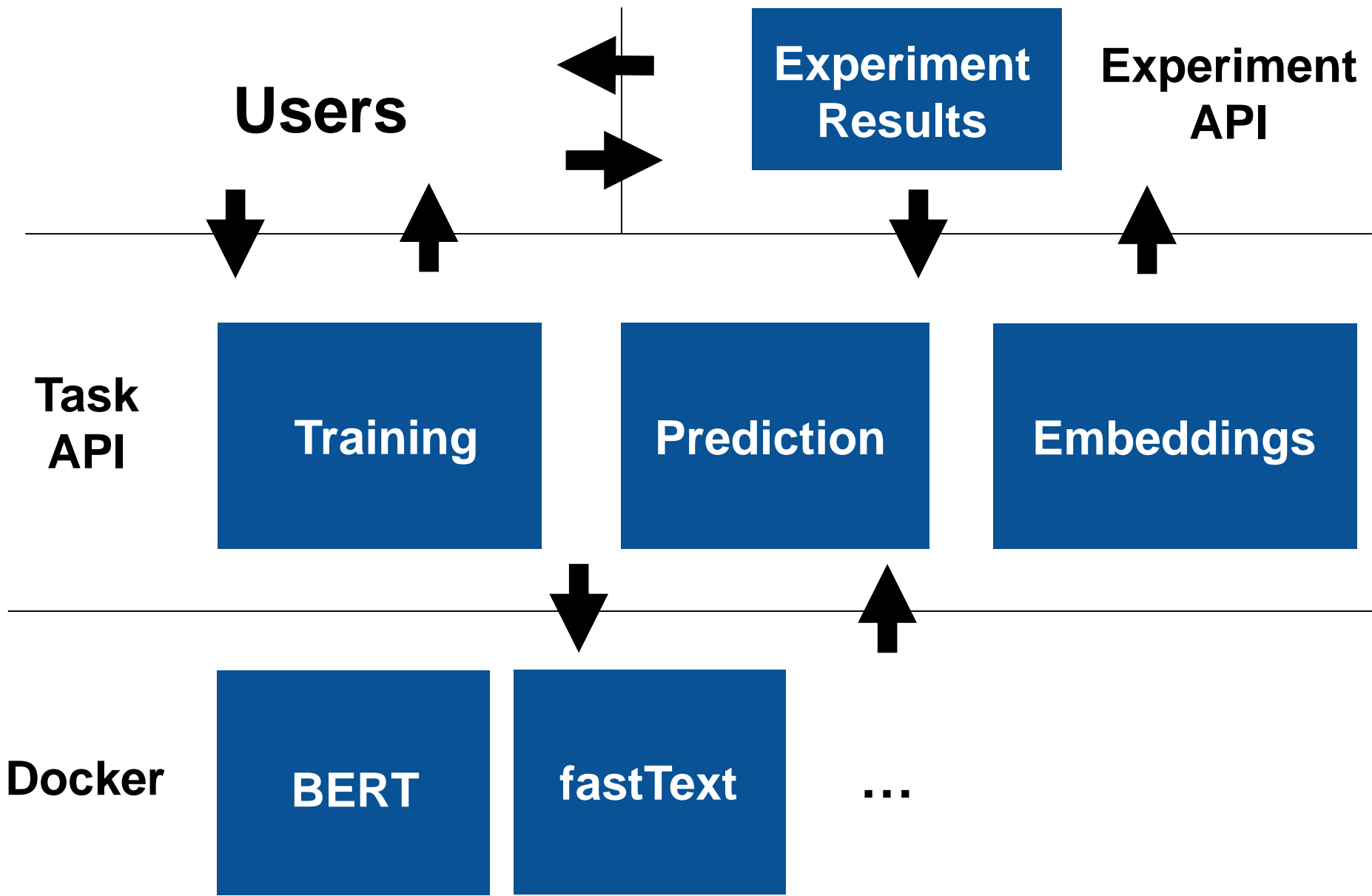
Rank	Name	Model	URL	Score	CoLA	SST-2	MRPC	STS-B	QQP	MNLI-m	MNLI-mm	QNLI	RTE	WNLI	
1	Microsoft D365 AI & UMD	Adv-RoBERTa (ensemble)		88.8	68.0	96.8	93.1/90.8	92.4/92.2	74.8/90.3	91.1	90.7	98.8	88.7	89.0	
2	Facebook AI	RoBERTa		88.5	67.8	96.7	92.3/89.8	92.2/91.9	74.3/90.2	90.8	90.2	98.9	88.2	89.0	
3	XLNet Team	XLNet-Large (ensemble)		88.4	67.8	96.8	93.0/90.7	91.6/91.1	74.2/90.3	90.2	89.8	98.6	86.3	90.4	
+	4	Microsoft D365 AI & MSR AI MT-DNN-ensemble		87.6	68.4	96.5	92.7/90.3	91.1/90.7	73.7/89.9	87.9	87.4	96.0	86.3	89.0	
5	GLUE Human Baselines	GLUE Human Baselines		87.1	66.4	97.8	86.3/80.8	92.7/92.6	59.5/80.4	92.0	92.8	91.2	93.6	95.0	
+	6	王玮	ALICE large ensemble (Alibaba DAMO NLP)		87.0	69.2	95.2	92.6/90.2	91.1/90.6	74.4/90.7	88.2	87.9	95.7	83.5	87.0
7	Stanford Hazy Research	Snorkel MeTaL		83.2	63.8	96.2	91.5/88.5	90.1/89.7	73.1/89.9	87.6	87.2	93.9	80.9	65.0	
8	XLM Systems	XLM (English only)		83.1	62.9	95.6	90.7/87.1	88.8/88.2	73.2/89.8	89.1	88.5	94.0	76.0	71.0	
9	Zhuosheng Zhang	SemBERT		82.9	62.3	94.6	91.2/88.3	87.8/86.7	72.8/89.8	87.6	86.3	94.6	84.5	65.0	
10	Danqi Chen	SpanBERT (single-task training)		82.8	64.3	94.8	90.9/87.9	89.9/89.1	71.9/89.5	88.1	87.7	94.3	79.0	65.0	
11	Kevin Clark	BERT + BAM		82.3	61.5	95.2	91.3/88.3	88.6/87.9	72.5/89.7	86.6	85.8	93.1	80.4	65.0	
12	Nitish Shirish Keskar	Span-Extractive BERT on STILTs		82.3	63.2	94.5	90.6/87.6	89.4/89.2	72.2/89.4	86.5	85.8	92.5	79.8	65.0	
13	Jason Phang	BERT on STILTs		82.0	62.1	94.3	90.2/86.6	88.7/88.3	71.9/89.4	86.4	85.6	92.7	80.1	65.0	
14	廖亿	RGLM-base (Huawei Noah's Ark Lab)		81.0	55.1	94.2	90.7/87.7	89.5/88.7	72.2/89.4	85.6	85.1	92.1	78.5	65.0	
+	15	Jacob Devlin	BERT: 24-layers, 16-heads, 1024-hidden		80.5	60.5	94.9	89.3/85.4	87.6/86.5	72.1/89.3	86.7	85.9	92.7	70.1	65.0

<https://gluebenchmark.com/leaderboard/>

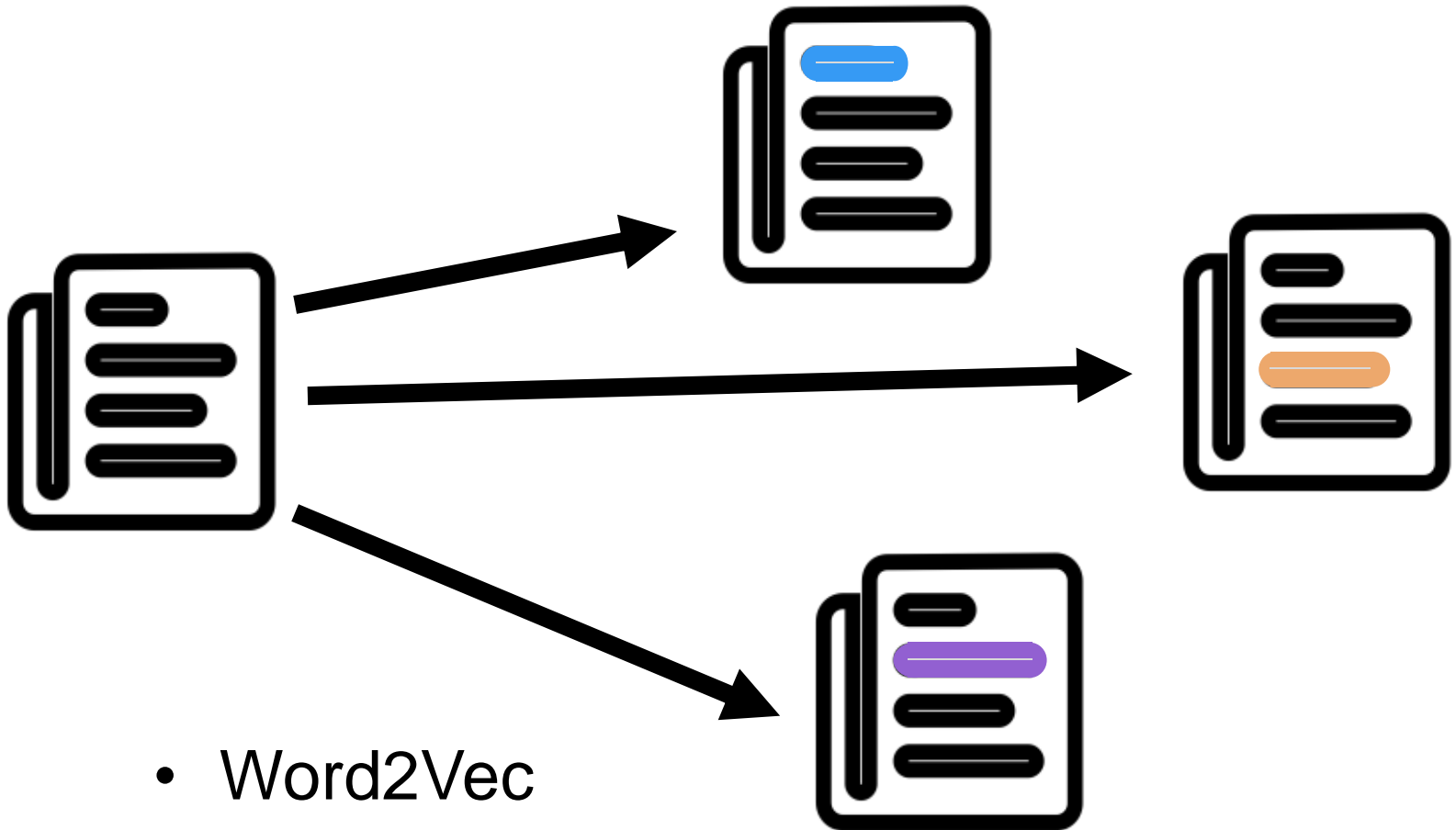
# gobbli: A Uniform Interface for Text Deep Learning Models



# gobbli: Library Design



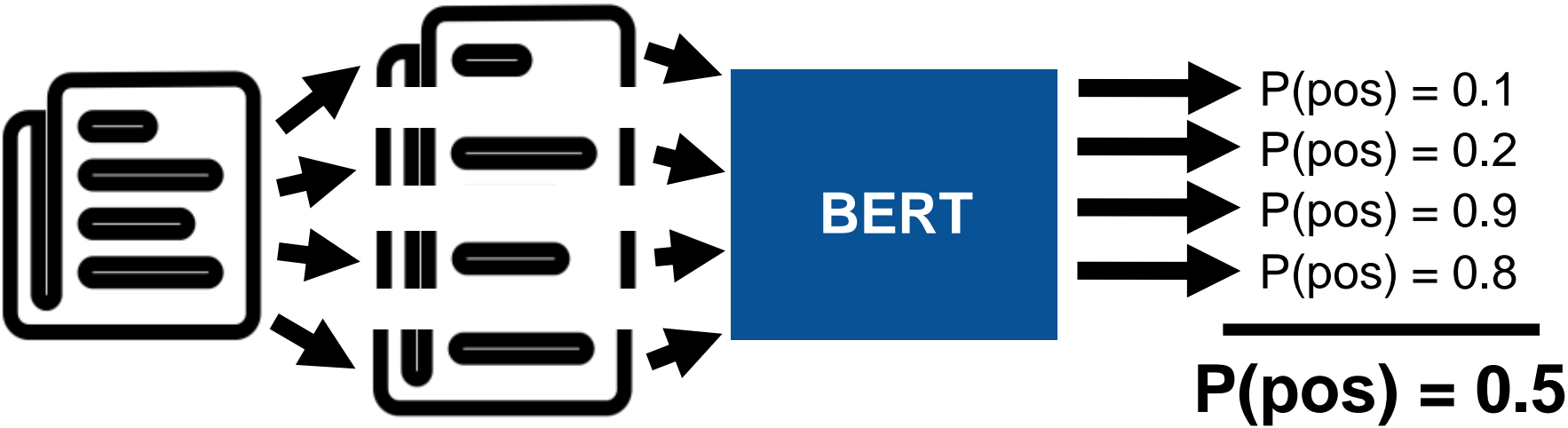
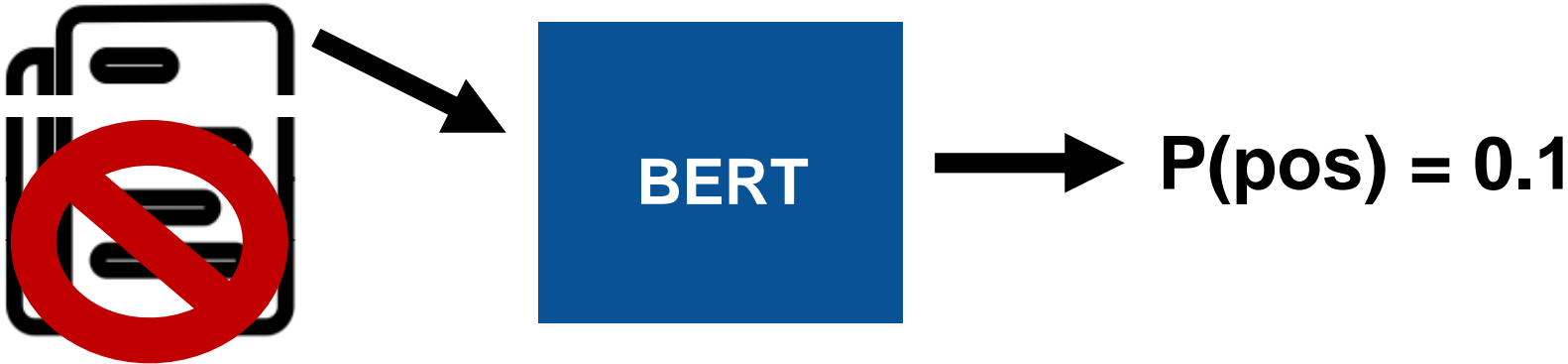
## Data Augmentation



- Word2Vec
- WordNet
- BERT Masked Language Model

# gobbli: Additional Features

## Document Windowing





# gobbl: Benefits and Drawbacks



- + Cross-platform
- + Abstracts dependency management
- Latency/overhead



- + Parallel/distributed training
- Experiment API only

# Example Experiment Results

# Example Experiment Results: Metrics

Metrics:

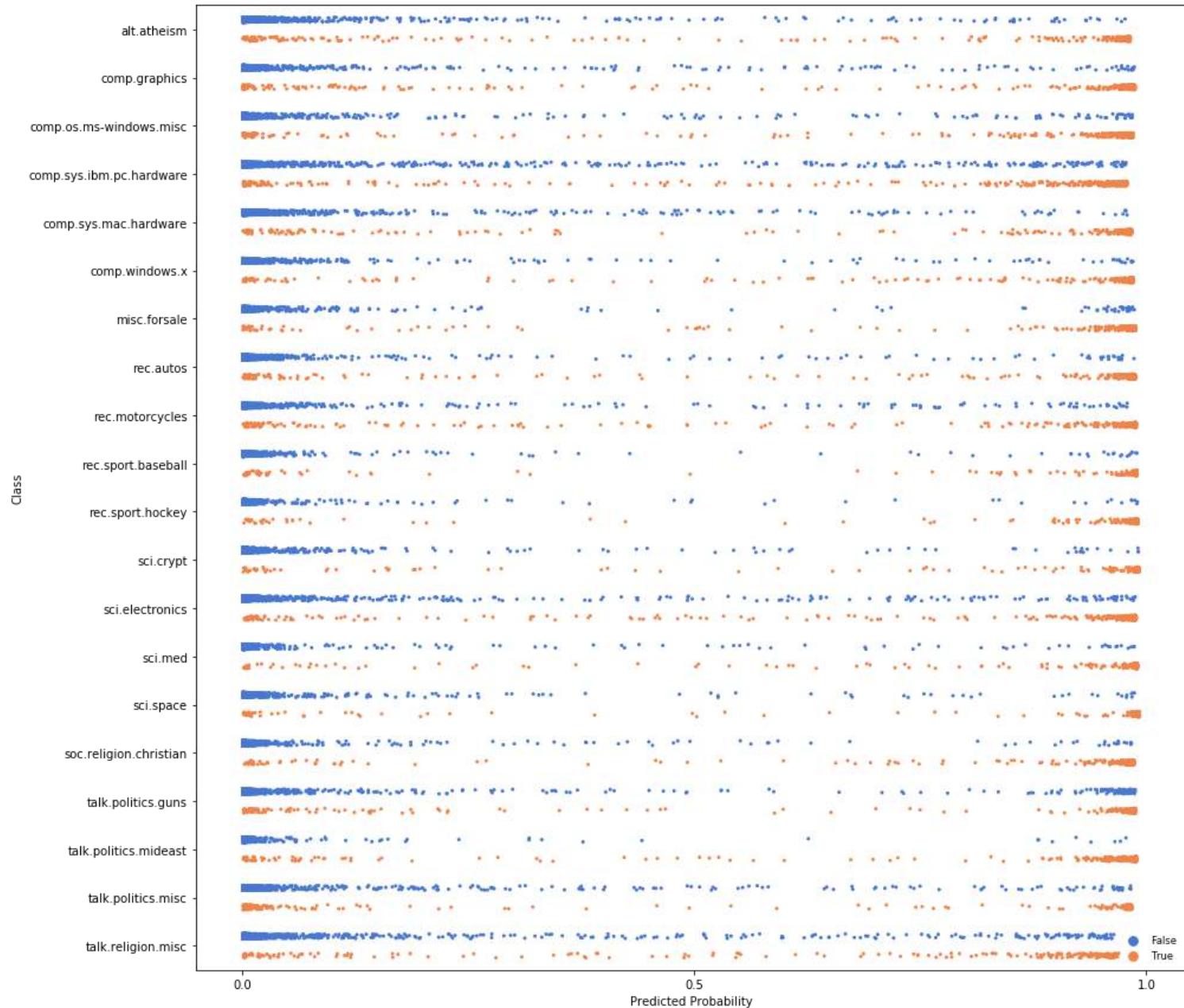
-----  
Weighted F1 Score: 0.8806791429898766  
Weighted Precision Score: 0.8806909370983464  
Weighted Recall Score: 0.88068  
Accuracy: 0.88068

Classification Report:

-----

	precision	recall	f1-score	support
neg	0.88	0.88	0.88	12500
pos	0.88	0.88	0.88	12500
accuracy			0.88	25000
macro avg	0.88	0.88	0.88	25000
weighted avg	0.88	0.88	0.88	25000

# Example Experiment Results: Plot



# Example Experiment Results: Errors Report

**True Class:** comp.os.ms-windows.misc

**Predicted Class:** sci.med (Probability: 0.97)

**Text:**

“My wife is a physiotherapist and she is looking for some cliparts of skeleton and male/female body. We're currently using Windows Draw which can import all kind of graphic formats. Therefore, anything will do. Please advise ...”

# gobbli: Status and Next Steps

- Initial open source release on GitHub
  - <https://github.com/RTIInternational/gobbli/>
  - Models implemented: BERT, MT-DNN, USE, fastText, pytorch\_transformers (XLNet, XLM, BERT, RoBERTa)
- Next steps:
  - Support multilabel classification
  - Helper module for downstream tasks using embeddings
  - Helper module for exploratory descriptives
  - Other bug fixes/enhancements requested by the community



**Jason Nance**  
**Data Scientist**  
**RTI International**  
jnance@rti.org