In-process Diagnostic methods for Entity Representation Learning on Sequential Data at Scale

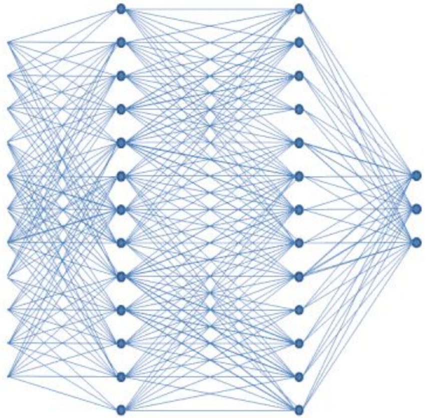
TEXAS
The University of Texas at Austin

PhD Proposal Presentation: Diego Garcia-Olano
Advisor: Dr. Joydeep Ghosh

June 15, 2020

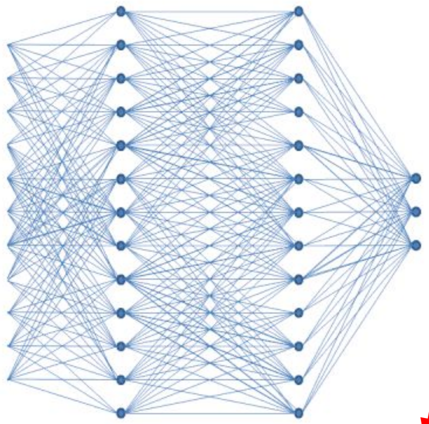
Explainable AI for Sequential Data

For image, text and time series data tasks, deep learning neural nets have become the default modeling choice.



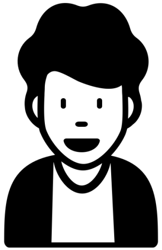
Explainable AI for Sequential Data

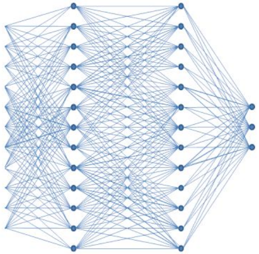
For image, text and time series data tasks, deep learning neural nets have become the default modeling choice.



Their ubiquity necessitates **transparency** into how such models arrive at the predictions they make in order that they be deemed **trustworthy** for use in critical domains.

- **Who** are we explaining to:
  End user? Expert/Researcher?
  Model developers? Other Models?

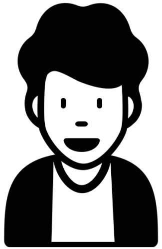


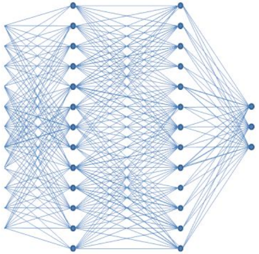


- **Who** are we explaining to:

  End user? Expert/Researcher?

  Model developers? Other Models?

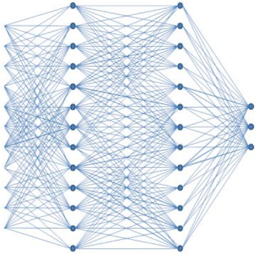




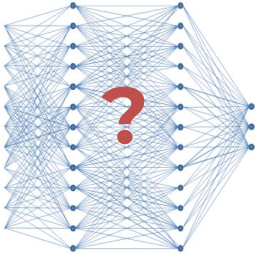
- **White Box** vs **Black Box**:

  Do we have access to the model internals?

  The data it was trained on?



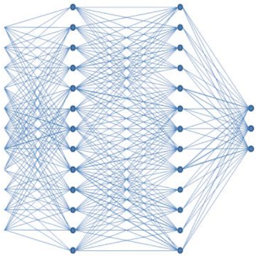
vs

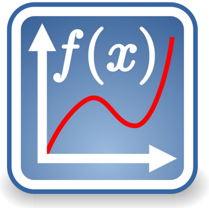


- **Explaining from what point in model process**:
  Pre-model, In-Process or Post Hoc

Secondary g(f(x))

data ->



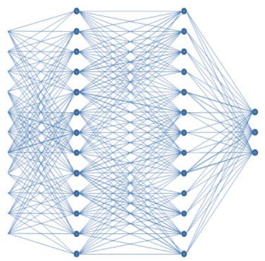


Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead (Rudin, et al, 2019 Nature)

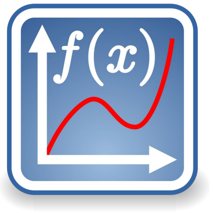
- **Explaining from what point in model process**:

    Pre-model, In-Process or Post Hoc

data ->



Secondary g(f(x))



Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead (Rudin, et al, 2019 Nature)

- **Global** model vs **Individual** instance based explanations

Types of Explainable AI

Post Hoc

Feature Attribution: which features contributed most to a model's output

- Path Integrated Gradients (IG)
- Shapley Additive Explanations (SHAP)
- Interpretability with Differential Masking

Influential examples: which training data most influenced a model's output

- Influence Functions
- Representer Point Selection for Explaining Deep Neural Networks

Counterfactuals: minimal change that would have led to a different output

BERT probing: assess how well a LM encodes semantic/syntatic properties of language by evaluating ("probing") on downstream tasks

Issues with Post Hoc secondary model explainers

Feature importance/saliency methods
- Need Baselines (Shap / IG)
- Are local/linear approximations of the actual model faithful explanations?
- Can we interpret Attention weights as explanations?

Influence functions:
- Expensive to compute
- Correlation to true influence for deep architectures

Counterfactuals:
- Semantic distance and meaning with text?

BERT probing:
- Don’t generalize past probing tasks and don’t “explain” model decisions

In-Process

Prototypes: learn “prototypical” representations
- Deep Learning for Case-Based Reasoning through Prototypes

Deep k-NN models: utilize layer representations as additional “clustering” features
- Deep k-Nearest Neighbors: Towards Confident, Interpretable and Robust DL

Concept Bottlenecks: layer specific additional task loss
- Concept bottleneck models
- On completeness-aware concept-based explanations in deep neural networks

Retrieval as Explanation: for tasks involving entity retrieval as an intermediate step
- REALM: retrieval-augmented language model pre-training
- Entities as experts: Sparse memory access with entity supervision

Feature Importance as an auxiliary loss during training:
- Incorporating Priors with Feature Attribution on Text Classification

Require access and modifications to the underlying model ….

In-Process

Prototypes: learn “prototypical” representations
- Deep Learning for Case-Based Reasoning through Prototypes

Deep k-NN models: utilize layer representations as additional “clustering” features
- Deep k-Nearest Neighbors: Towards Confident, Interpretable and Robust DL

Concept Bottlenecks: layer specific additional task loss
- Concept bottleneck models
- On completeness-aware concept-based explanations in deep neural networks

Retrieval as Explanation: for tasks involving entity retrieval as an intermediate step
- REALM: retrieval-augmented language model pre-training
- Entities as experts: Sparse memory access with entity supervision

Feature Importance as an auxiliary loss during training:
- Incorporating Priors with Feature Attribution on Text Classification

Require access and modifications to the underlying model ….
which is fine for critical applications!

Objectives

In-process explainable models for Sequential Data

- **are an Useful & Under-explored area for sequential data modeling**
- **provide Interpretable and Faithful explanations of model decisions**
- **allow for model “diagnosis” and intervention at inference time.**

Objectives

In-process explainable models for Sequential Data

- **are an Useful & Under-explored area for sequential data modeling**
- **provide Interpretable and Faithful explanations of model decisions**
- **allow for model "diagnosis" and intervention at inference time.**

Entity Representation learning allows for an additional interesting and underexplored explainability aspect that grounds models.

Scalability is vital to the adoption of models in practice
Both play a central role in this work.

Completed Work

- Learning Dense Representations for Entity Retrieval. (CoNLL 2019)
- Deep Classification of Time-Series Data with Learned Prototype Explanations. (ICML time series workshop 2019 *joint work with Alan Gee*)
- Biomedical Interpretable Entity Representations. (ACL-IJCNLP 2021)

Completed Work

Learning Dense Representations for Entity Retrieval. (CoNLL 2019)	Constructed a **dual mention-entity encoder** that learns dense representations for efficient neural **Entity Retrieval** with an **in-process, iterative hard negatives procedure** for **model learning and inference time inspection**.
Deep Classification of Time-Series Data with Learned Prototype Explanations. (ICML 19)	Adapted a **prototypical autoencoder** classifier to be compatible with **time series data** and allow for **tunable prototype diversity** leading to improved accuracy and **global and instance level explanations**.
Biomedical Interpretable Entity Representations. (ACL-IJCNLP 2021)	Learned a distantly supervised entity type system and data set for use in training a **Biomedical Interpretable Entity model** whose representations exist in a **semantically meaningful vector space** & whose **predictions may be interpreted and diagnosed** with an oracle method.

Learning Dense Representations for Entity Retrieval

Gillick, D., Kulkarni, S., Lansing, L., Presta, A., Baldridge, J., Ie, Eugene., Garcia-Olano, D. "Learning Dense Representations for Entity Retrieval". Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL), Hong Kong, China, 2019.

Motivation

Entity Resolution:
Predict the most probable "entity" in a knowledge graph (Wikipedia) that a "mention" links to given its surrounding "context."

Motivation

Entity Resolution:
Predict the most probable "entity"
in a knowledge graph (Wikipedia)
that a "mention" links to
given its surrounding "context."

Example Query: What is George Harrison's favorite Nintendo game?

Mention: George Harrison
Context: What is __ favorite Nintendo Game ?
Entity: ????

5.7 million entities to choose from in Wikipedia (considering only english)

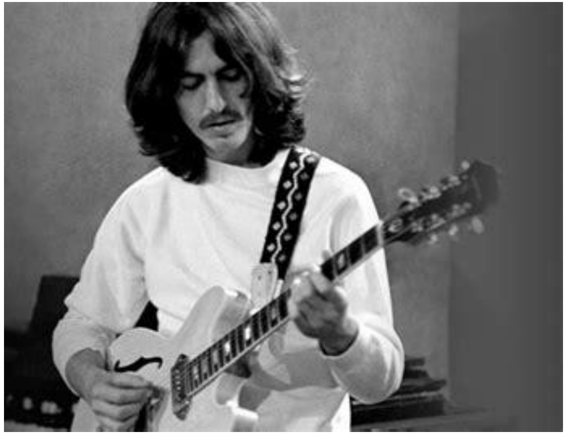
Finding the real entity this mention resolves to
allows us to **learn representations grounded in the real world**.

and to **leverage structured data** from the knowledge graph.

Motivation

Example Query: What is George Harrison’s favorite Nintendo game?

George Harrison



Q2643

George Harrison

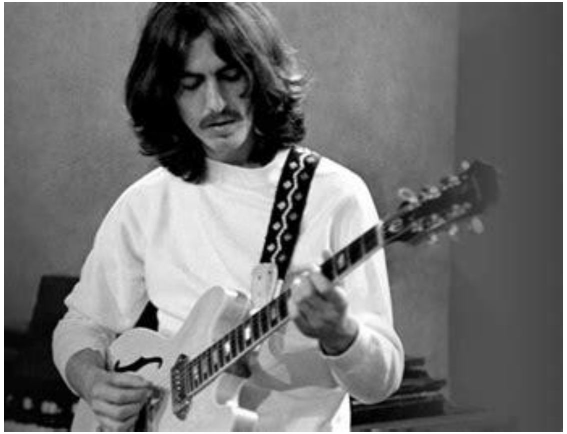
Q5540278

Motivation

Example Query: What is George Harrison's favorite Nintendo game?

Beatles Guitarist

Highest Popular Prior



Q2643

Former Senior VP of Marketing

at Nintendo of America.

Q5540278

Motivation

Prior State of the Art for Entity Resolution:

- Train on (Mention, Context, Entity) Triples.

2 Stages

(1) Retrieve Candidates

- Construct a Mention to Entities Lookup **"Alias" Table**.

  9.8 Million unique mention strings

  5.7 Million unique entities

(2) Re-Rank them

- **Limitations**

1) Low Recall
2) Context not considered. Can't predict unseen entities

Contributions

Define a **novel dual encoder architecture** for
learning entity and mention **encodings** suitable **for retrieval**

Describe a fully **unsupervised, iterative hard-negative mining** algorithm
that greatly improves retrieval performance and
can be used to track and **explain model learning.**

Approximate nearest neighbor search yields quality candidate entities efficiently.

Outperform discrete retrieval baselines (alias table, BM25) and
gives results competitive with the best reported accuracy on TACKBP-2010.

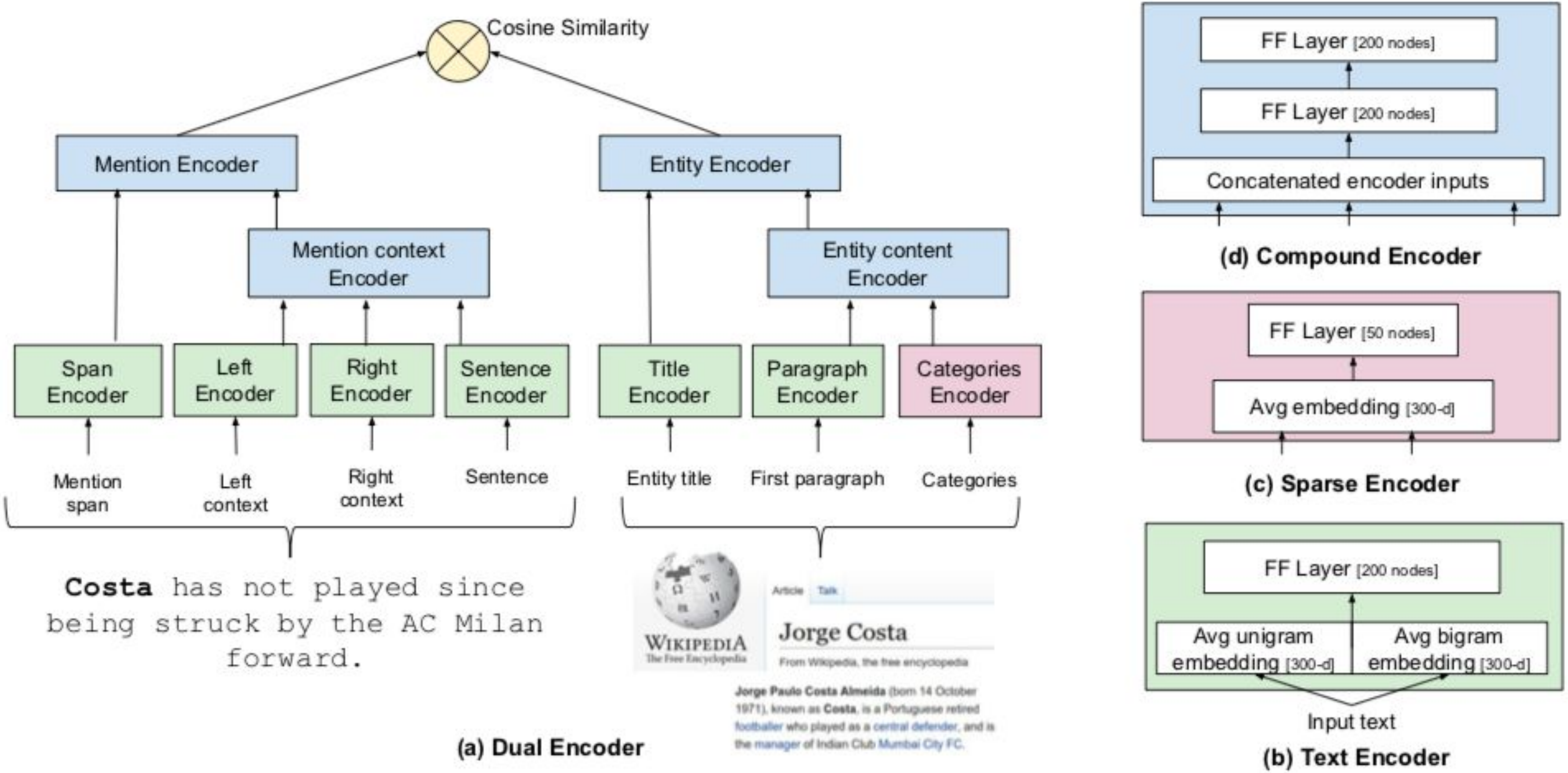


Figure 1: Architecture of the dual encoder model for retrieval (a). Common component architectures are shown for (b) text input, (c) sparse ID input, and (d) compound input joining multiple encoder outputs. Note that all text encoders share a common set of embeddings.

Training with Sampled Softmax

The dual encoder learns a mention encoder φ and an entity encoder ψ,

where the **score** of a mention-entity pair (m, e) is:

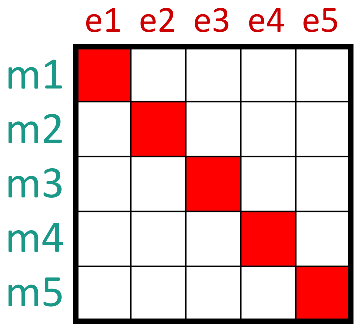
$$\mathbf{s}(m, e) = \cos(\ \varphi(m),\ \ \psi(e)\ )$$

Training with Sampled Softmax

The dual encoder learns a mention encoder φ and an entity encoder ψ,

where the **score** of a mention-entity pair (m, e) is:

$$\mathbf{s}(m, e) = \cos(\ \varphi(m),\ \ \psi(e)\ )$$



These pairs constitute only positive examples,
so we use **in-batch random negatives** (Henderson et al., 2017;):

We build the all-pairs similarity matrix for all mentions & entities in a batch.
& **optimize a softmax loss** on each row of the matrix.

We do this **sampled softmax** (Jozefowicz et al, 2016)
in place of a full softmax
because the normalization term
is *intractable* to compute over all 5.7M entities.

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

Training with Sampled Softmax

For each training pair (m_i, e_i) in a batch of ***B*** pairs, the loss is computed as:

$$L(m_i, e_i) = -f(m_i, e_i) + \log \sum_{j=1}^{B} \exp(f(m_i, e_j))$$

$$\text{where} \quad f(m_i, e_j) = a \cdot s(m_i, e_j)$$

We track **in-batch recall@1** accuracy on val set
and stop training after the metric flattens out (about 40M steps).

Recall@1 means for each instance,
the models gets a score of 1
if the correct entity is ranked above all in-batch random negatives,
0 otherwise.

<u>Hyperparams</u>: **batch size of 100**, fixed learning rate 0.01
SGD with Momentum of 0.9,

Iterative Hard Negative Sampling

Random negatives are not enough to train an accurate entity resolution model

So after learning an initial model using random negatives,
we identify more challenging **"hard negatives"** via the following:

1. Encode all mentions and entities found in training pairs using current model.
2. For each mention, retrieve the most similar 10 entities (i.e., its nearest neighbors).
3. Select all entities ranked above the correct one as negative examples.

Iterative Hard Negative Sampling

Random negatives are not enough to train an accurate entity resolution model

So after learning an initial model using random negatives,
we identify more challenging **"hard negatives"** via the following:

1. Encode all mentions and entities found in training pairs using current model.
2. For each mention, retrieve the most similar 10 entities (i.e., its nearest neighbors).
3. Select all entities ranked above the correct one as negative examples.

We merge these new hard negative mention/entity pairs
with the original positive pairs to construct an additional task
& resume training the dual encoder using logistic loss on them.

For a pair (m, e) with label y ∈ {0, 1}, the **hard negative loss** is defined as:

$$L_h(m, e; y) = -y \cdot \log f(m, e) - (1 - y) \cdot \log(1 - f(m, e))$$

$$\text{where} \quad f(m, e) = g(a_h \cdot s(m, e) + b_h)$$

Multi-task loss & Task Results

The hard negative task is mixed with the original random negatives task

$$L_{multi} = L_{orig} + L_{hard}$$

System	R@1	Entities
AT-Prior	71.9	5.7M
AT-Ext	73.3	5.7M
Chisholm and Hachey (2015)	80.7	800K
He et al. (2013)	81.0	1.5M
Sun et al. (2015)	83.9	818K
Yamada et al. (2016)	85.2	5.0M
Nie et al. (2018)	86.4	5.0M
Barrena et al. (2018)	87.3	523K
DEER (this work)	87.0	5.7M

Table 1: Comparison of relevant TACKBP-2010 results using Recall@1 (accuracy). While we cannot control the candidate entity set sizes, we attempt to approximate them here.

Multi-task loss & Task Results

The hard negative task is mixed with the original random negatives task

$$L_{multi} = L_{orig} + L_{hard}$$

System	R@1	Entities
AT-Prior	71.9	5.7M
AT-Ext	73.3	5.7M
Chisholm and Hachey (2015)	80.7	800K
He et al. (2013)	81.0	1.5M
Sun et al. (2015)	83.9	818K
Yamada et al. (2016)	85.2	5.0M
Nie et al. (2018)	86.4	5.0M
Barrena et al. (2018)	87.3	523K
DEER (this work)	87.0	5.7M

Table 1: Comparison of relevant TACKBP-2010 results using Recall@1 (accuracy). While we cannot control the candidate entity set sizes, we attempt to approximate them here.

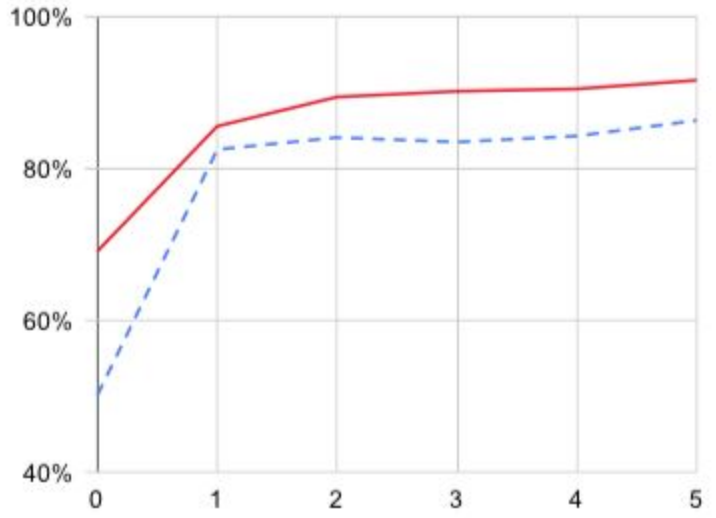
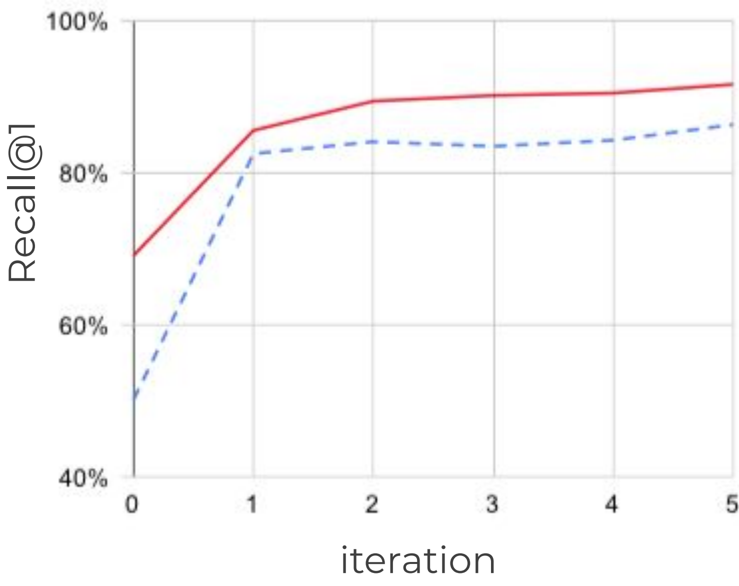


Figure 2: Recall@1 improvement for successive iterations of hard negative mining for Wikinews (solid) and TACKBP-2010 (dashed).

Hard Negative In-Process Explanations

During each iteration of learning, we identify entities which our model assigns a higher ranking than the true entity associated with a given mention and context.



These **hard negative triples** (m, e’, 0) **can be inspected over time** during training or inference to assess the mention/contexts and entities that are added which are difficult for the model to learn (esp. later iterations)

This **interpretable in-process information about the learning process** could be used to:

- improve **error analysis**,
- identify cases where **additional supervision** could be useful
- gauge **confidence** in inference time predictions

Hard Negative In-Process Explanations

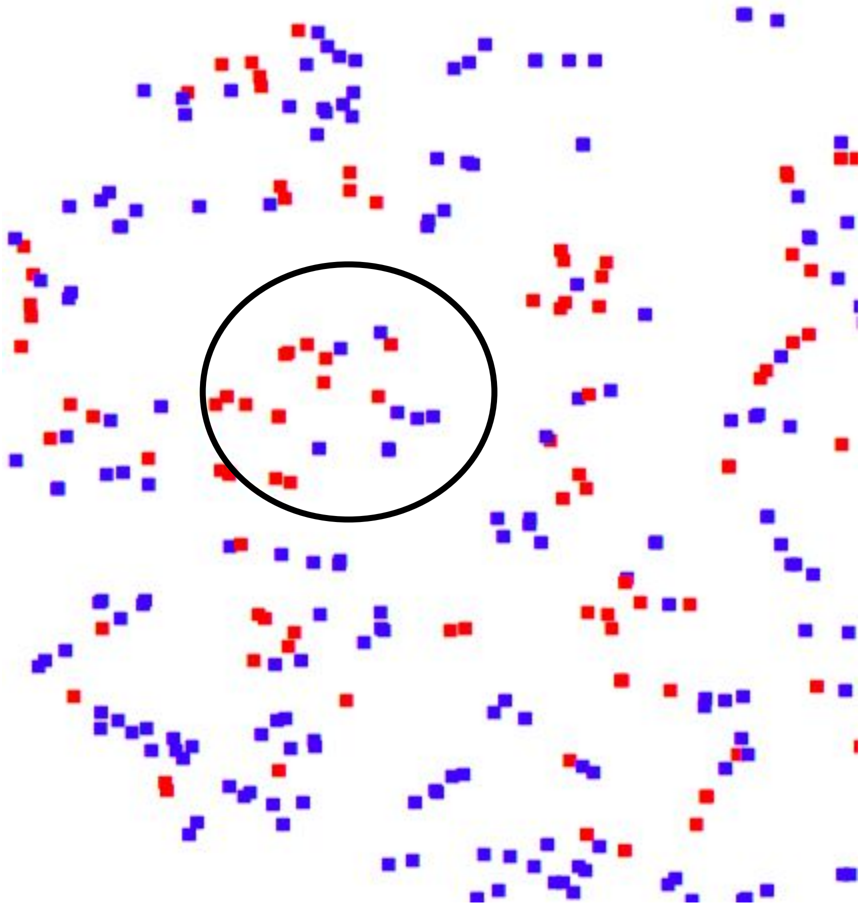
At inference time,
given a test mention/context,

1) Get K nearest mention/contexts from training set

2) Collectively assess how each of them performed over iterations (gather the hard negatives along with the true entities)

3) Get top entity prediction(s) for the test mention/context via cosine similarity

4) Utilize 2 and 3 results to calculate confidence measures for the final entity prediction



Inspecting Entity Encodings for Semantic Meaning

Figure 3.5: t-SNE visualization of our learned embeddings for select country Wikipedia page entities. More at diegoolano.com/deer/

Retrieval at Scale

Inference is done by computing cosine similarity between the test mention/context encoding and each of the cached entity encodings.

Approximate Search using quantization-based approaches (Guo et al. (2016)) can be used to speed up retrieval greatly!

Method	Mean search time (ms)	Wikinews R@100
Brute force	291.9	97.88
AH	22.6	97.22
AH+Tree	3.3	94.73

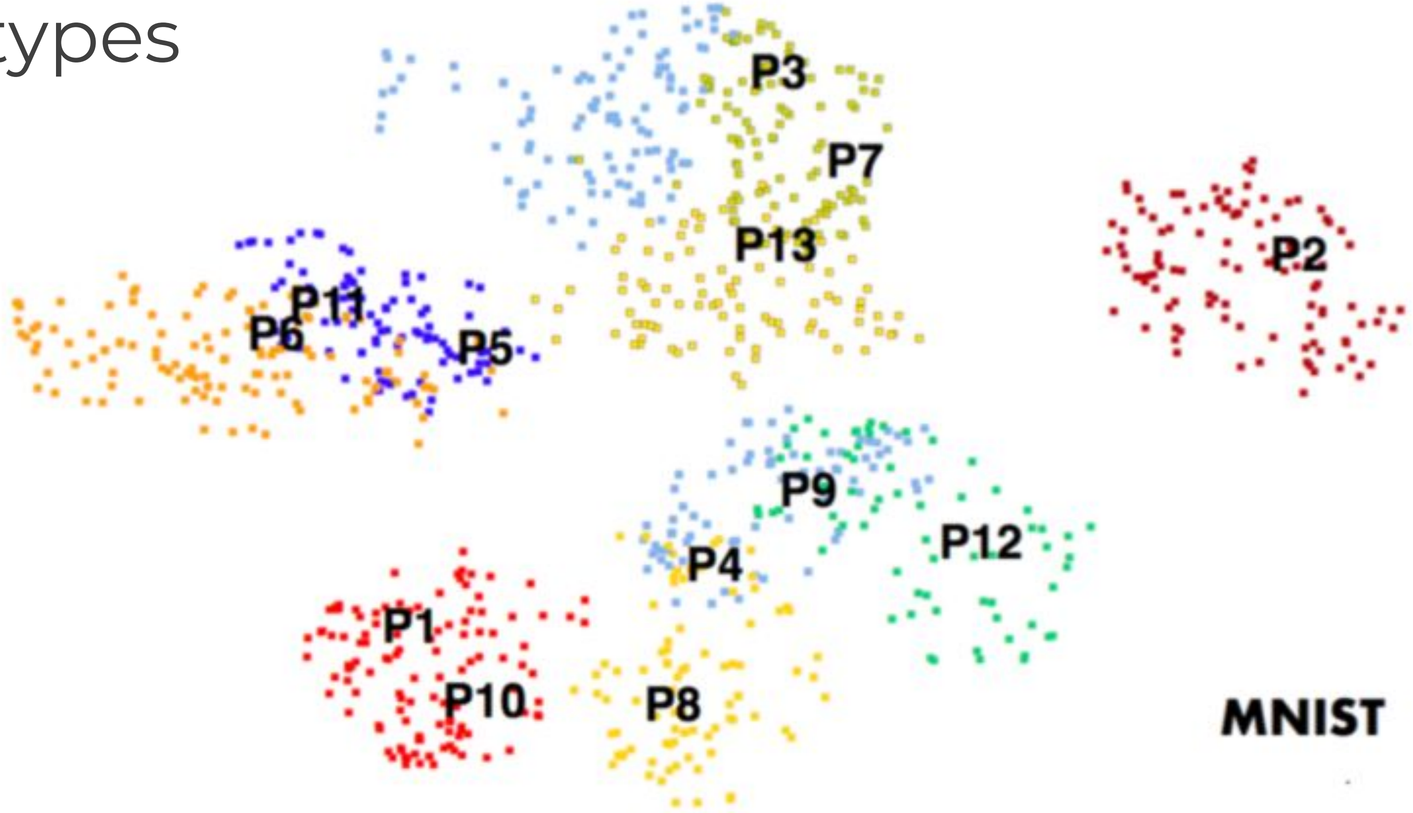
Table 3: Comparison of nearest-neighbor search methods using the DEER model. The benchmark was conducted on a single machine. AH indicates quantization-based asymmetric hashing; AH+Tree adds an initial tree search to further reduce the search space.

Explaining Deep Classification of Time-Series Data with Learned Prototypes

Garcia-Olano, D.*, Gee, A.*, Ghosh, J., Paydarfar, D. “Deep Classification of Time-Series Data with Learned Prototype Explanations”. International Conference on Machine Learning (ICML 2019 time series workshop)

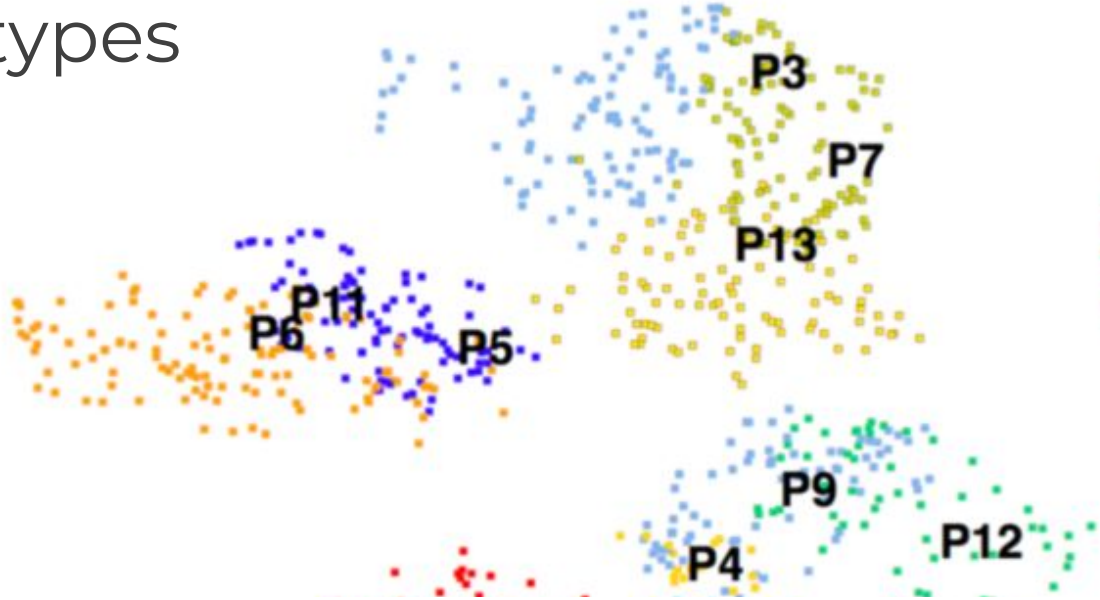
* *equal contribution*

Prototypes



*Li et al. Deep learning for case-based reasoning through prototypes. (2017)

Prototypes

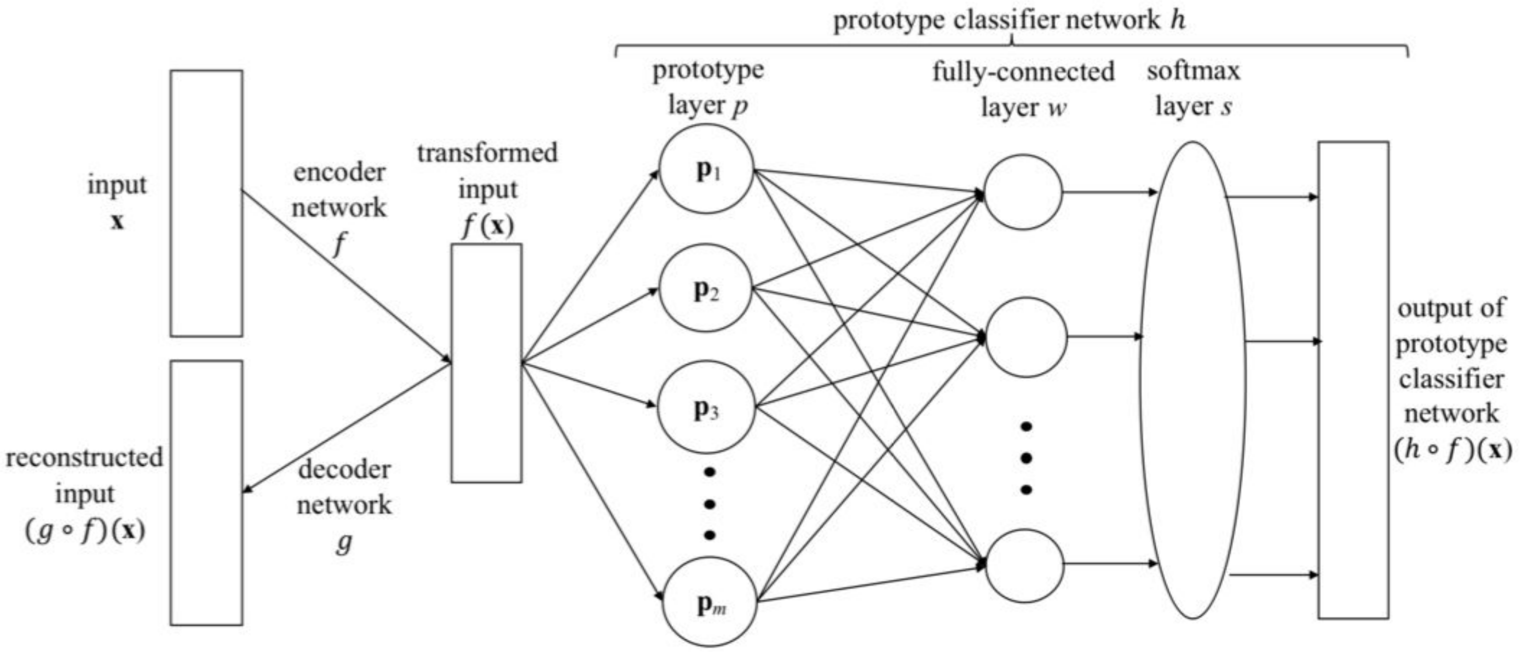


totypes. (2017)

Motivation

Prototype Classifier Network

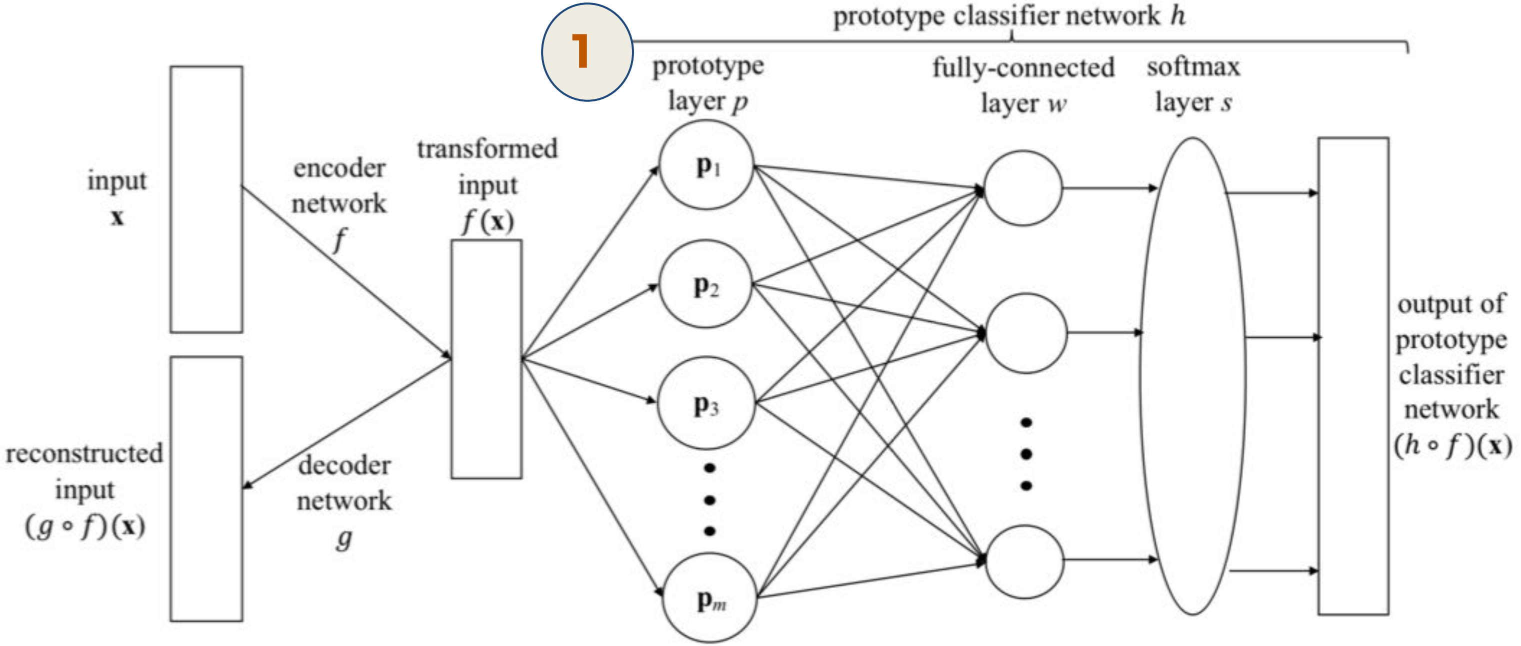
n data points
m prototypes



Motivation

Prototype Classifier Network

1



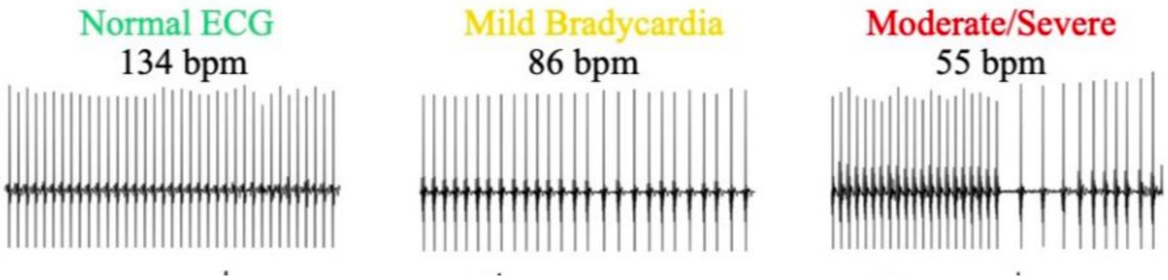
n data points
m prototypes

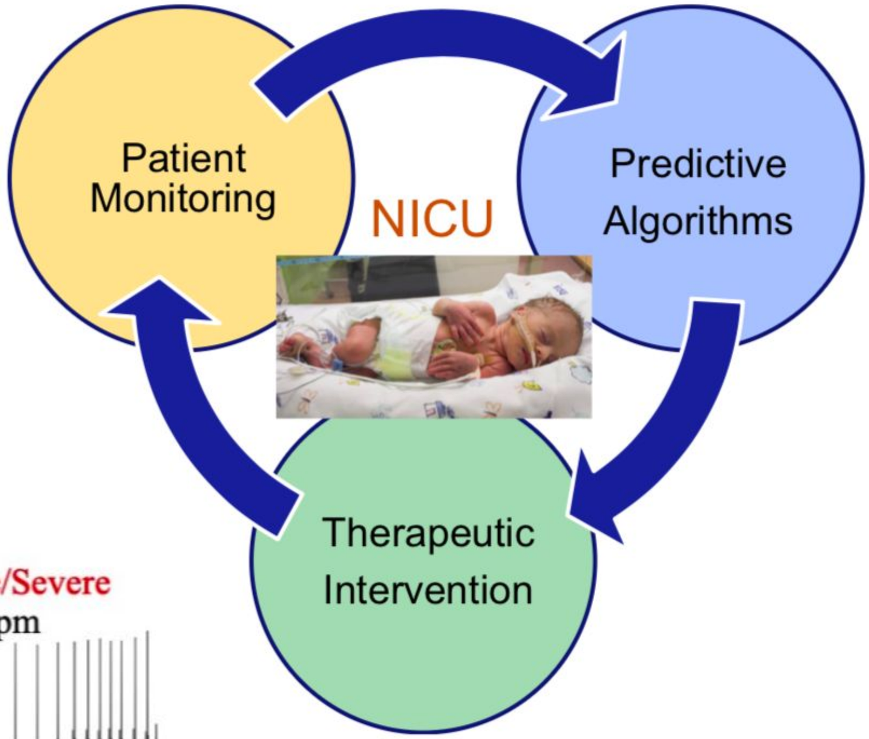
1

$$\underbrace{\text{Feature Vector } f(\boldsymbol{x}) \in \mathbb{R}^q \qquad \text{Prototypes } \boldsymbol{p}_i \in \mathbb{R}^q}_{\|f(\boldsymbol{x}) - \boldsymbol{p}_i\|_2^2 = \boldsymbol{\rho}_i}$$

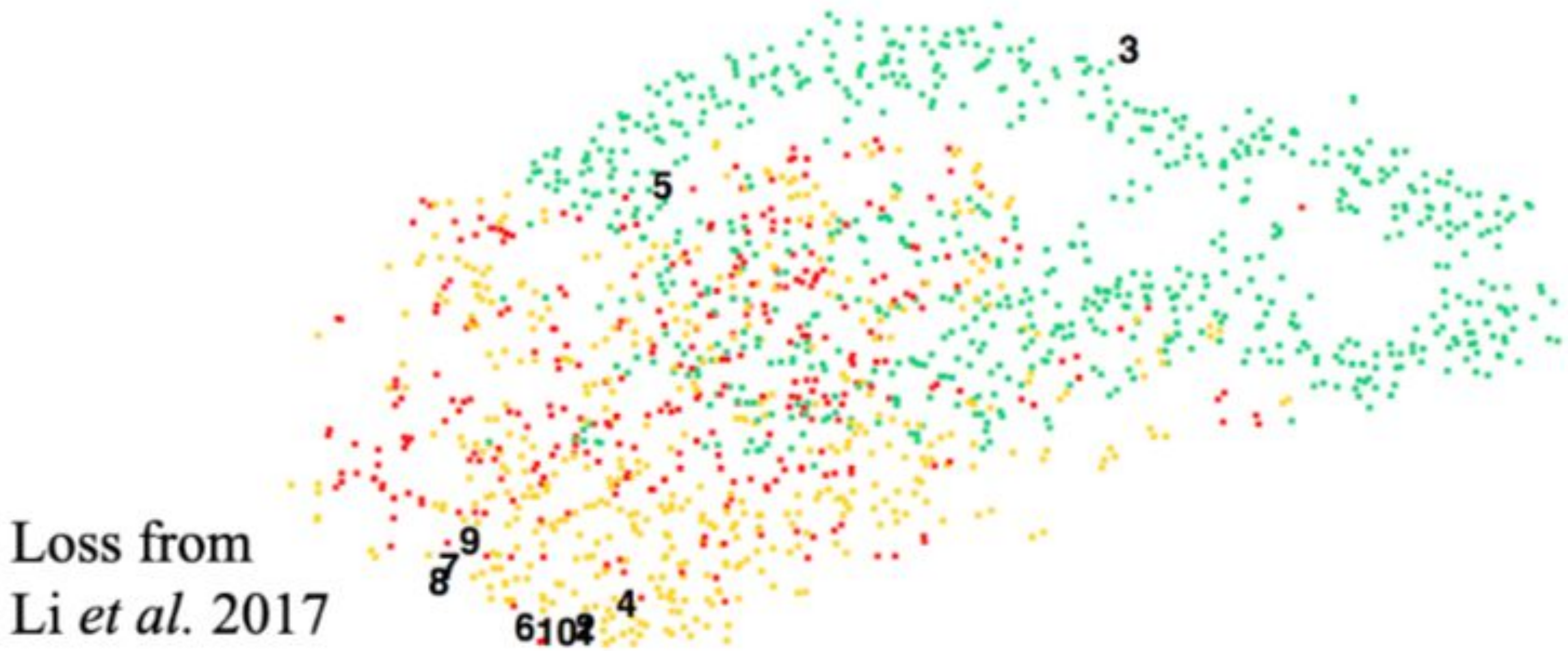
$$\begin{aligned}\mathcal{L}((f, g, h), X) = E(h \circ f, X) &+ \lambda_R R(g \circ f, X) \\ &+ \lambda_1 R_1(p_1, \ldots, p_m, X) \\ &+ \lambda_2 R_2(p_1, \ldots, p_m, X)\end{aligned}$$

Predicting Bradycardia from ECG signals





Prior work
Latent Space Representation for Bradycardia task



Prototype Classifier Network Updated

$$\mathcal{L}((f,g,h),X) = E(h \circ f, X) + \lambda_R R(g \circ f, X) + \lambda_1 R_1(p_1, ..., p_m, X) + \lambda_2 R_2(p_1, ..., p_m, X) + \lambda_{pd} PDL(p_1, ..., p_m) \quad (2)$$

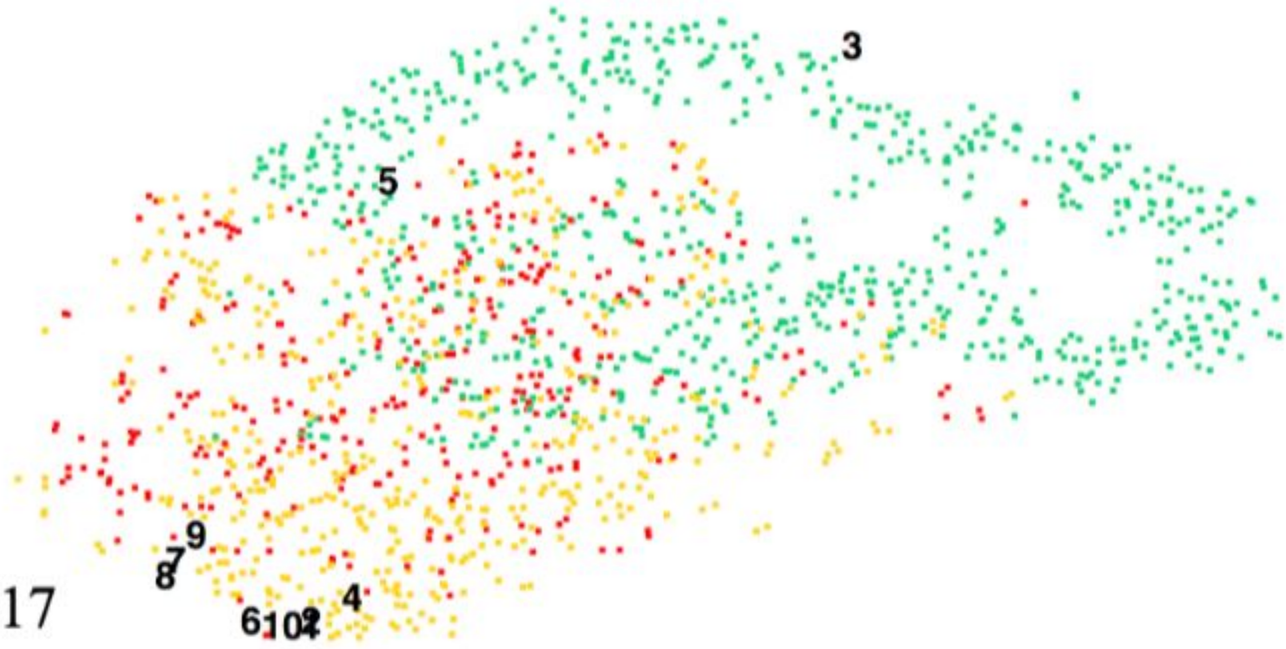
Prototype Diversity Loss

$$\lambda_{pd} PDL(p_1, ..., p_m) = \frac{1}{log(\frac{1}{m}\sum_{j=1}^{m} min_{i>j \in [1,m]} \|p_i - p_j\|_2^2) + \epsilon} \quad (1)$$

$$R_1(p_1, ..., p_m, X) = \frac{1}{m}\sum_{j=1}^{m} min_{i \in [1,n]} \|p_j - f(x_i)\|_2^2, \quad (3)$$

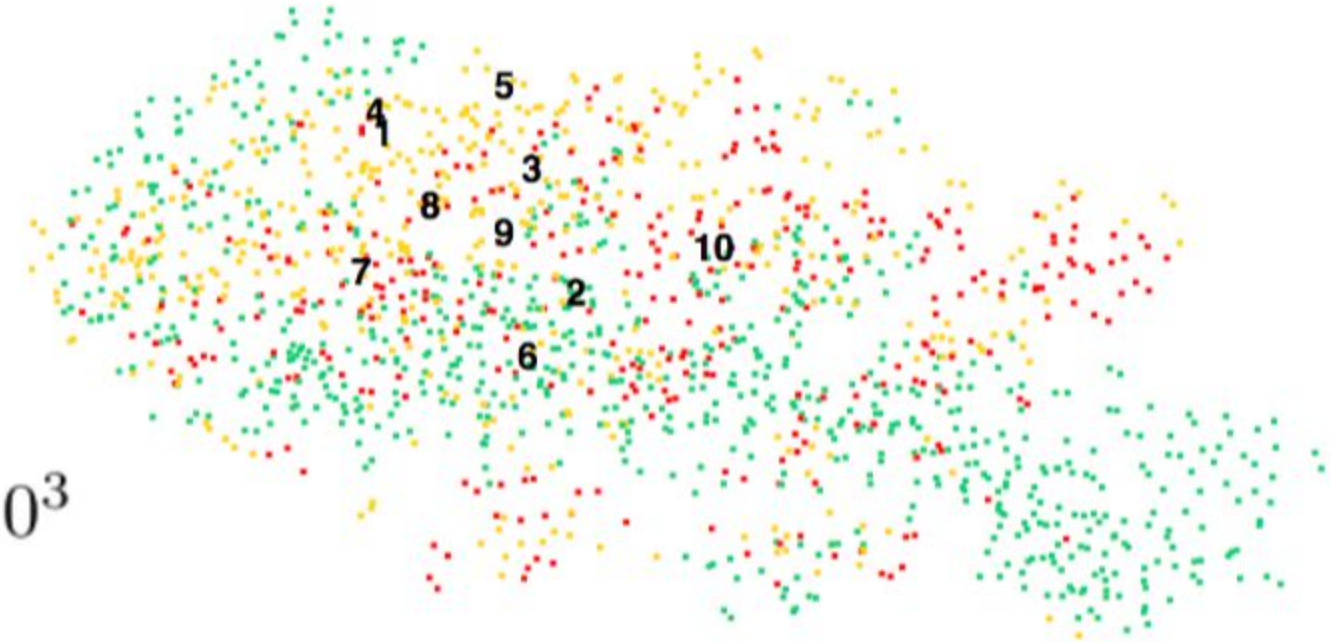
$$R_2(p_1, ..., p_m, X) = \frac{1}{n}\sum_{i=1}^{n} min_{j \in [1,m]} \|f(x_i) - p_j\|_2^2 \quad (4)$$

Prior work:
Latent Space Representation for Bradycardia task



Loss from
Li *et al.* 2017
$(\lambda_{pd} = 0)$

Our work:
Latent Space Representation for Bradycardia task



$$\lambda_{pd} = 10^3$$

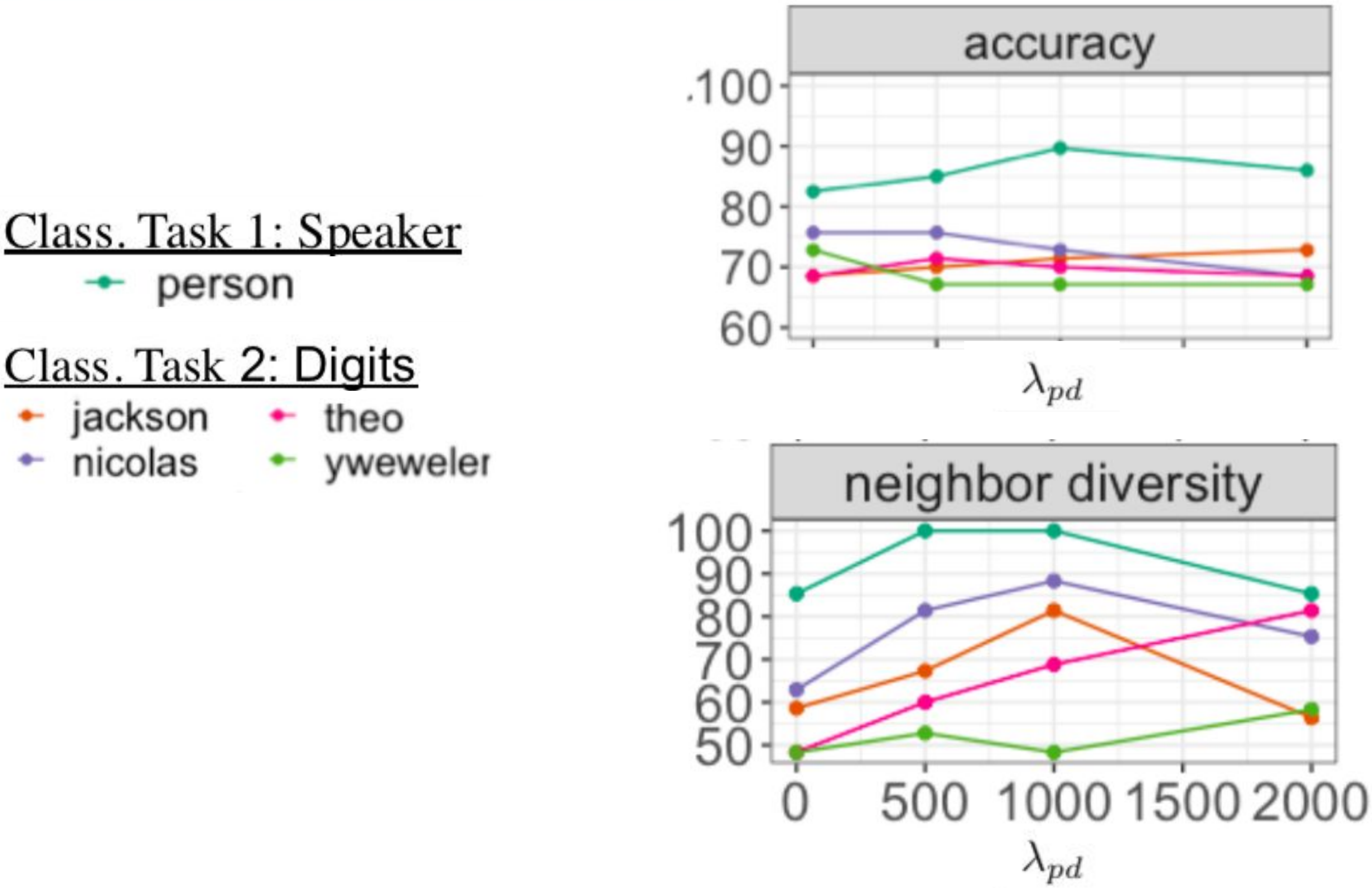
Results and Prototype Diversity

ECG: Bradycardia			
λ_{pd}	Accu.	Ψ_N	Ψ_C
0	$92.1 \pm 0.1\%$	0.83 ± 0.04	0.78 ± 0.19
500	$92.7 \pm 1.0\ \%$	0.86 ± 0.07	0.89 ± 0.19
$1e3$	$92.4 \pm 1.3\%$	0.87 ± 0.11	0.89 ± 0.19
$2e3$	$\mathbf{93.1 \pm 0.4\%}$	$\mathbf{0.90 \pm 0.04}$	$\mathbf{1.00 \pm 0.00}$

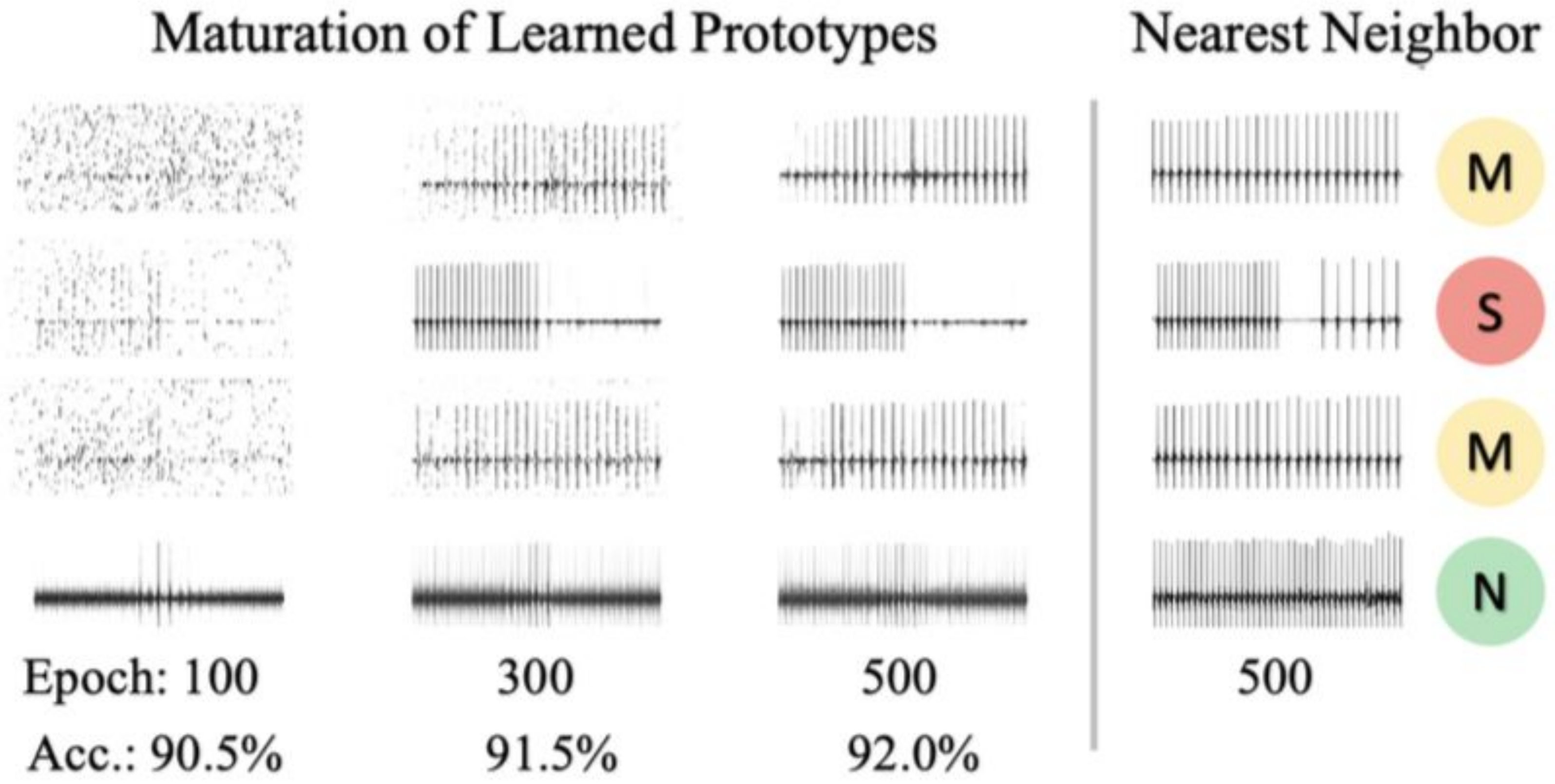
Prototype neighbor diversity Ψ_N

Prototype class diversity Ψ_C

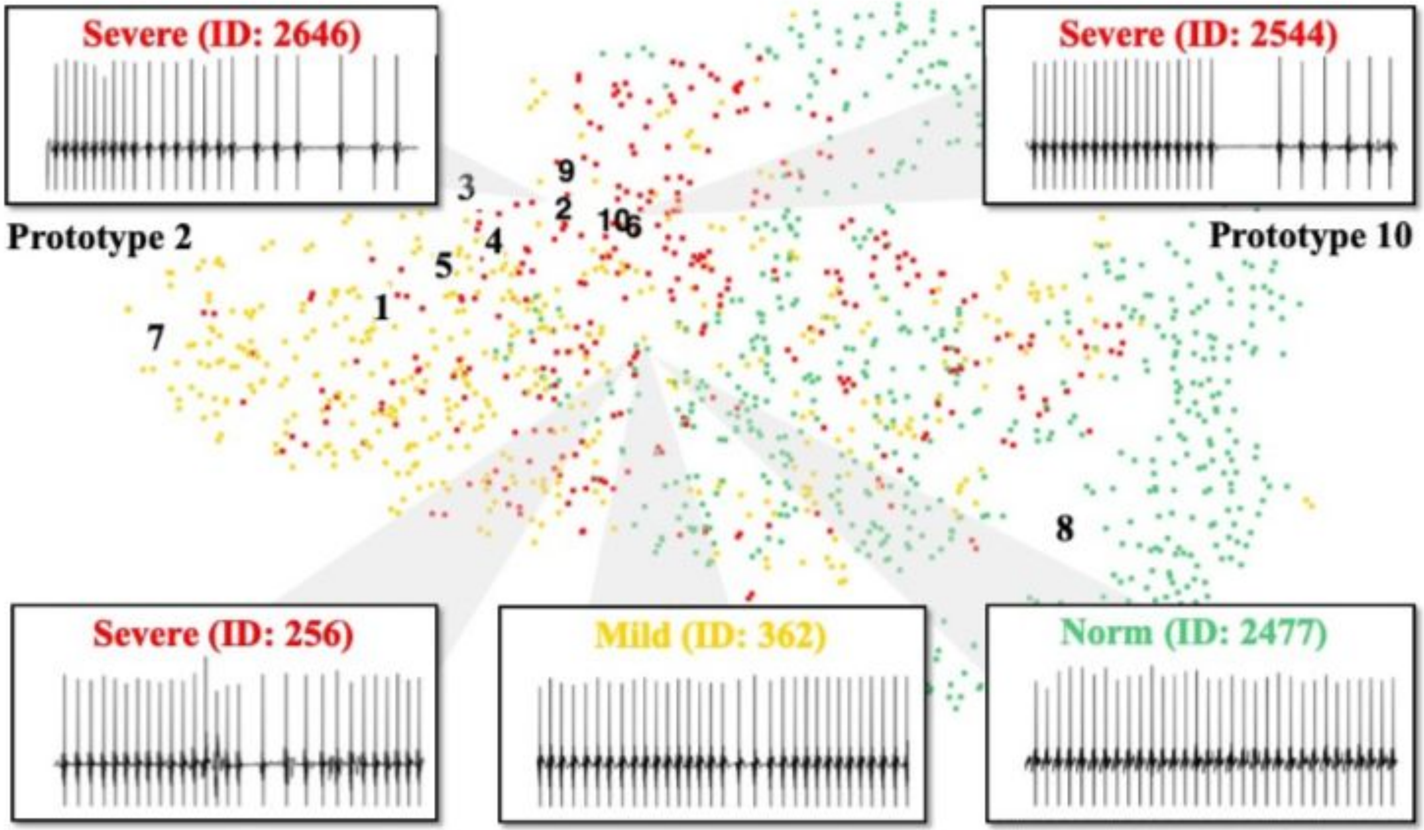
Spoken MNIST Performance



Explainability via Prototypes



Decoded Representations of Prototypes



Explainability via Prototypes

Spoken Digit Global Explainability

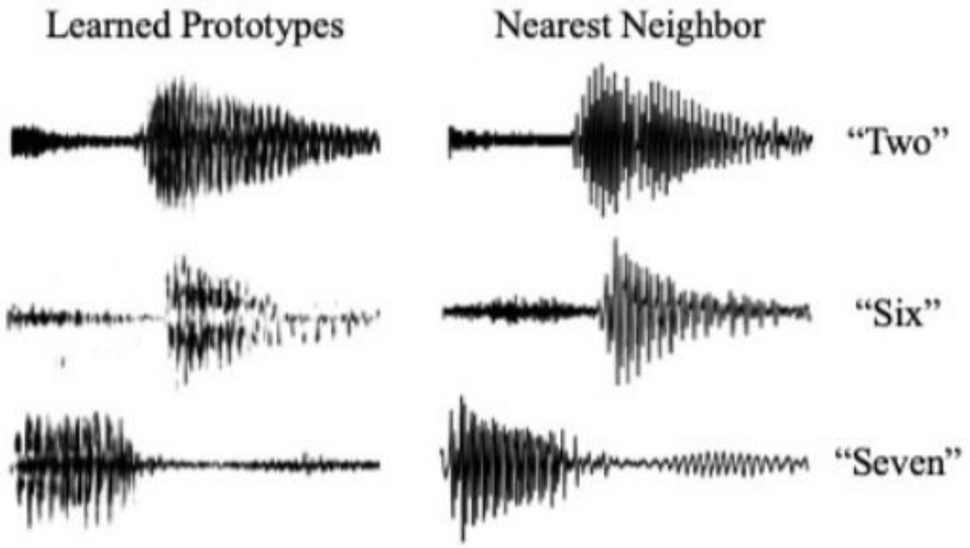
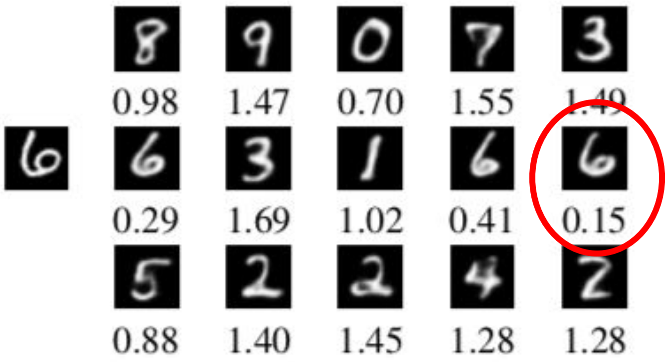


Figure 8: Learned prototypes from audio waveforms of spoken digits by Nicolas from the FSDD $(\lambda_{pd} = 500)$.

Instance Explainability



Biomedical Interpretable Entity Representations

Garcia-Olano, D., Onoe, Y., Baldini, I., Ghosh, J., Wallace, B., Varshey, K. “Biomedical Interpretable Entity Representations”. Findings of the Association for Computational Linguistics (ACL-IJCNLP 2021)

Motivation

Entities over text = typically embedded in dense vector spaces
with pre-trained language models (BERT,etc).

```
[0.519, 0.917, -0.935, 0.891, 0.396, 0.711, 0.479, 0.417, 0.744, -0.254,
-0.174, 0.233, -0.315, 0.497, -0.516, 0.22, -0.679, 0.389, -0.683, 0.909,
23, 0.528, 0.116, 0.334, 0.717, 0.857, -0.262, 0.624, -0.178, -0.045, -0.
 -0.952, 0.4, 0.356, 0.091, 0.976, -0.337, -0.002, 0.054, 0.512, -0.312,
.278, -0.409, -0.655, -0.294, -0.453, 0.735, 0.461, 0.282, -0.43, -0.838,
3, -0.736, -0.001, 0.889, -0.228, 0.645, 0.883, 0.805]
```

```
[0.656, 0.407, 0.568, -0.035, -0.842, -0.257, 0.202, -0.31, 0.886, 0.386,
34, -0.823, -0.929, -0.068, -0.238, 0.236, -0.463, 0.56, -0.687, -0.521,
88, 0.54, 0.047, -0.434, -0.009, 0.59, 0.971, 0.798, 0.202, 0.225, 0.131,
88, 0.44, -0.835, -0.032, -0.935, 0.318, 0.72, -0.23, -0.903, 0.912, -0.8
0.981, -0.23, 0.797, -0.785, -0.583, 0.055, -0.511, 0.413, -0.757, 0.914,
943, 0.62, -0.78, 0.888, 0.288, 0.807, -0.207, -0.284]
```

Motivation

Entities over text = typically embedded in dense vector spaces
with pre-trained language models (BERT,etc).

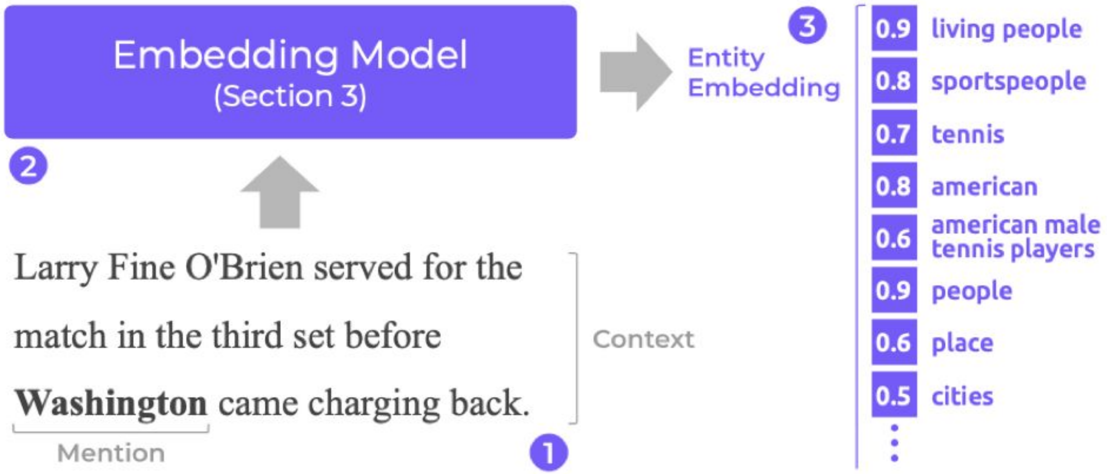
```
>>> word_embedding_for_happy
[0.519, 0.917, -0.935, 0.891, 0.396, 0.711, 0.479, 0.417, 0.744, -0.254,
-0.174, 0.233, -0.315, 0.497, -0.516, 0.22, -0.679, 0.389, -0.683, 0.909,
23, 0.528, 0.116, 0.334, 0.717, 0.857, -0.262, 0.624, -0.178, -0.045, -0.
 -0.952, 0.4, 0.356, 0.091, 0.976, -0.337, -0.002, 0.054, 0.512, -0.312,
.278, -0.409, -0.655, -0.294, -0.453, 0.735, 0.461, 0.282, -0.43, -0.838,
3, -0.736, -0.001, 0.889, -0.228, 0.645, 0.883, 0.805]
```

```
>>> word_embedding_for_sad
[0.656, 0.407, 0.568, -0.035, -0.842, -0.257, 0.202, -0.31, 0.886, 0.386,
34, -0.823, -0.929, -0.068, -0.238, 0.236, -0.463, 0.56, -0.687, -0.521,
88, 0.54, 0.047, -0.434, -0.009, 0.59, 0.971, 0.798, 0.202, 0.225, 0.131,
88, 0.44, -0.835, -0.032, -0.935, 0.318, 0.72, -0.23, -0.903, 0.912, -0.8
0.981, -0.23, 0.797, -0.785, -0.583, 0.055, -0.511, 0.413, -0.757, 0.914,
943, 0.62, -0.78, 0.888, 0.288, 0.807, -0.207, -0.284]
```

Not immediately interpretable.

Dense Entity Embeddings = Give good performance for entity-related tasks, but using them in those tasks requires additional processing in neural models.

Onoe et al* learn human readable interpretable entity representations that achieve high performance without additional learning ("out of the box")

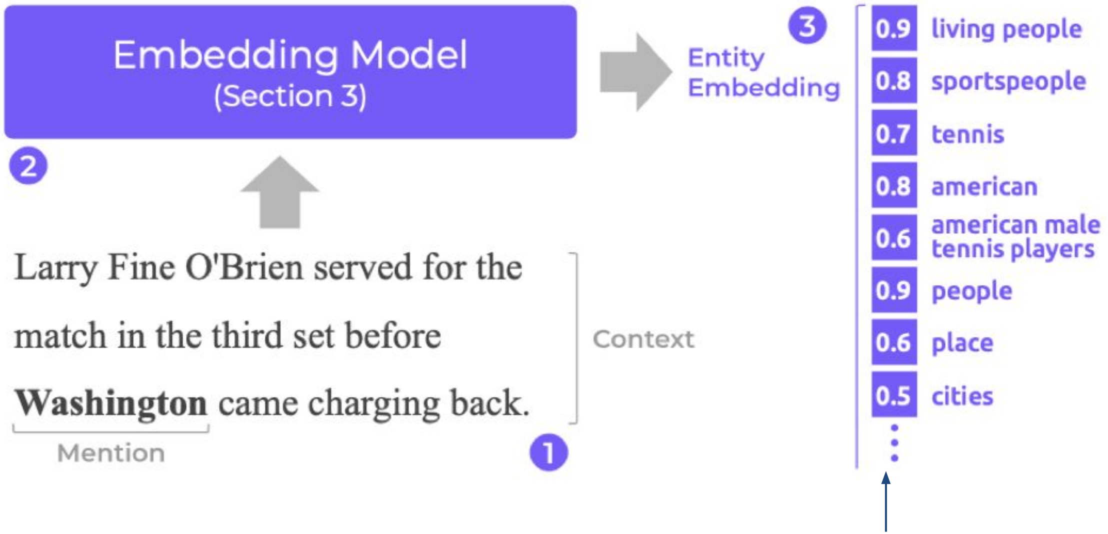


"Interpretable Entity Representations Through Large Scale Typing"
Yasumasa Onoe & Greg Durrett . Findings of EMNLP 2020

IERs

Onoe et al* learn human readable interpretable entity representations
that achieve high performance without additional learning ("out of the box")

fine grained entity types



represent probability of entity have corresponding properties

experiments using Ultra Fine Entity Type system **(10k)**
and Wikipedia Categories Type System **(60k)**

Problem setup: Interpretable Entity Representations

s = a sequence of **context words**,
m = an **entity mention span in s**.
t $\in [0, 1]^{T}$ binary vector of **entity types** over types in T

Goal: Learn parameters θ of a function f that
maps the mention m and its context s
⇒ to a vector t
that captures salient features of the entity mention in its context

High dimensional Multi-label classification task over entity types

Can we adapt IERs for the **Biomedical Domain?**

[Glesatinib] is a dual inhibitor of c-Met and SMO
that is under phase II clinical trial for non-small cell lung cancer.

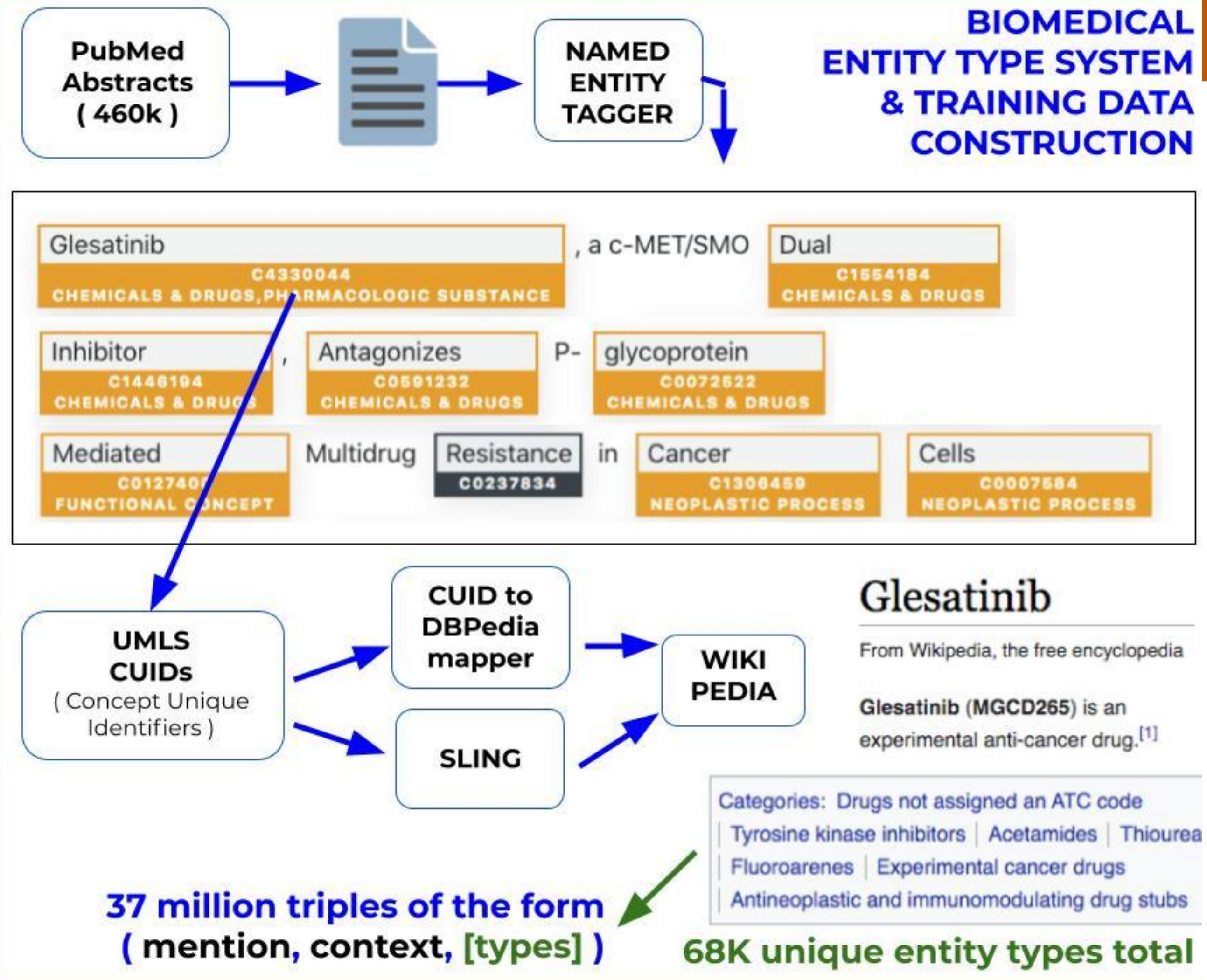
Can we adapt IERs for the **Biomedical Domain?**

[Glesatinib] is a dual inhibitor of c-Met and SMO
that is under phase II clinical trial for non-small cell lung cancer.

Most probable entity types for mention/context

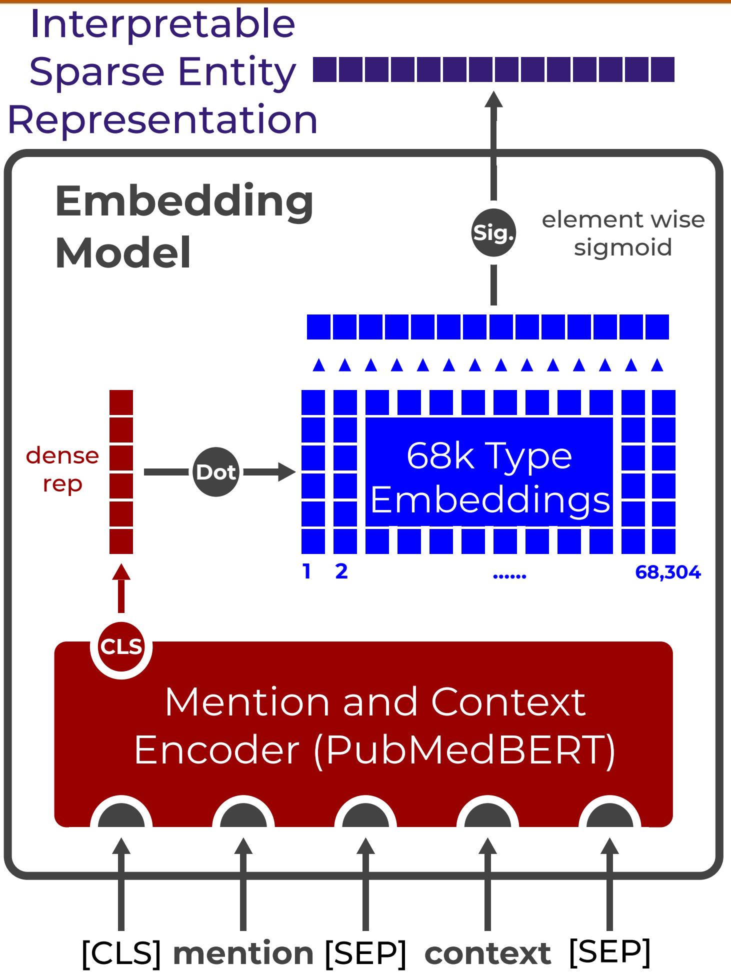
world health organization essential medicines	0.4941
pyridines	0.4073
diols	0.3539
cancer treatments	0.3260
carboxylate esters	0.2376
chloroarenes	0.1984
rtt	0.1879
hormonal antineoplastic drugs	0.1768
antineoplastic drugs	0.1037
alcohols	0.0771
prodrugs	0.0315
peptides	0.0300
methyl esters	0.0223
merck	0.0191
transgender and medicine	0.0135
teratogens	0.0130
world anti-doping agency prohibited substances	0.0124
peripherally selective drugs	0.0103
human proteins	0.0099
ureas	0.0090
withdrawn drugs	0.0089
iarc group 2a carcinogens	0.0073
prostate cancer	0.0066
mechanisms	0.0066
chemotherapy	0.0058
aromatase inhibitors	0.0057

of 60k wiki entity types

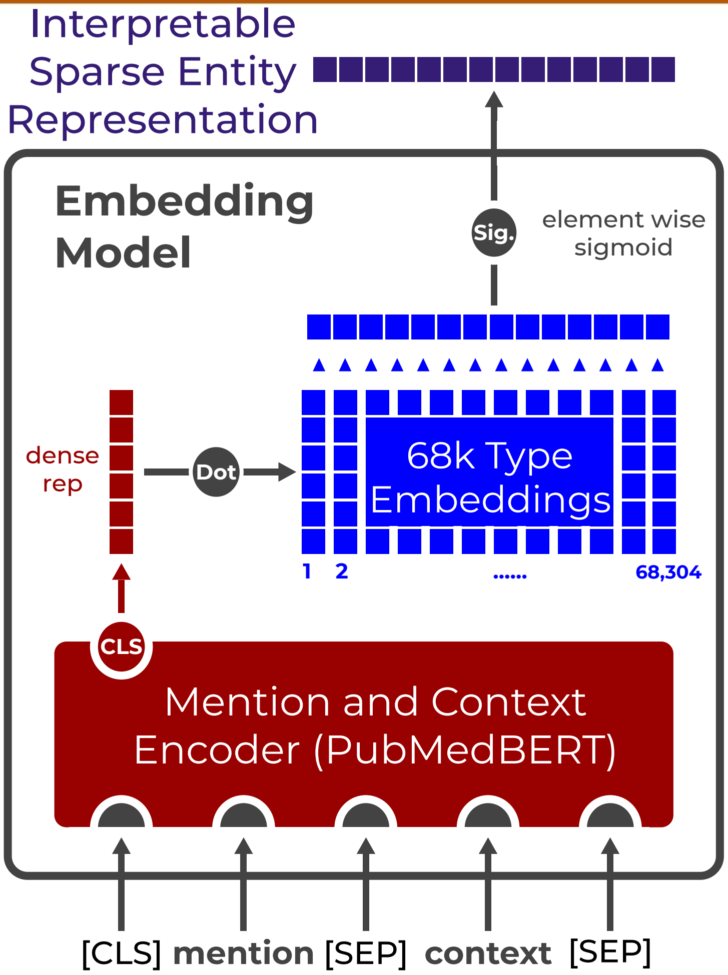


Distant Supervision to **construct** **Entity Type System** and **Training Data**.

Training Biomedical IERs



Training Biomedical IERs

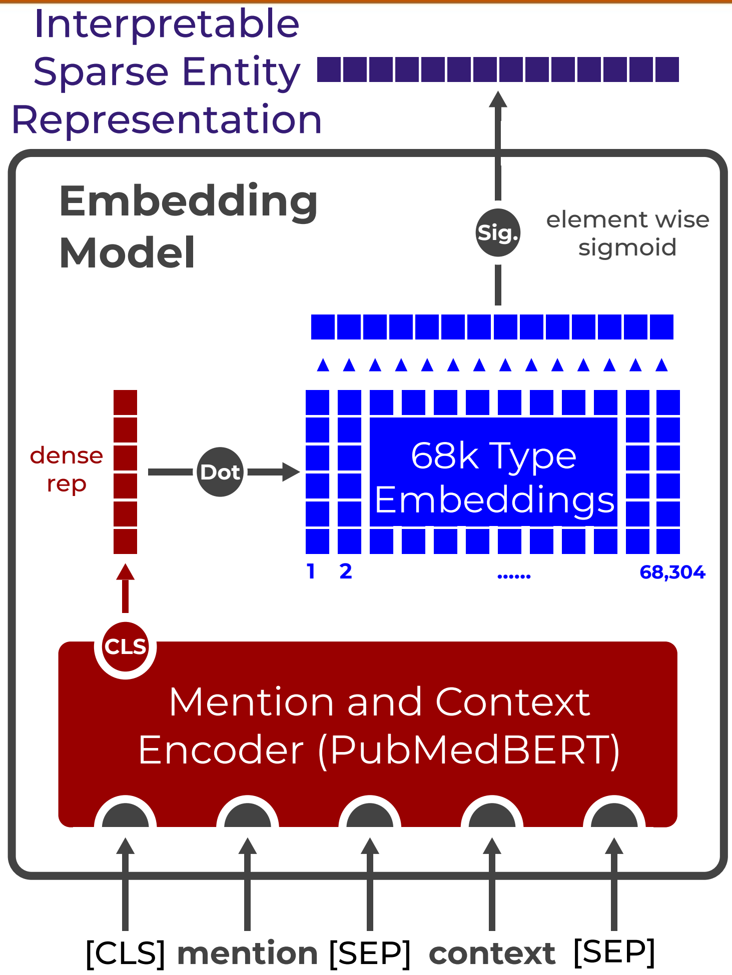


Training loss:

Independent sum
of binary cross entropy losses
over all all entity types T
over all training examples D.

$$-\sum_{i}^{D} \sum_{j}^{T} t^*_{ij} \cdot \log(t_{ij}) + (1 - t^*_{ij}) \cdot \log(1 - t_{ij}),$$

Inference using Biomedical IERs



Training loss:

Independent sum
of binary cross entropy losses
over all all entity types T
over all training examples D.

$$-\sum_{i}^{D}\sum_{j}^{T} t_{ij}^{*} \cdot \log(t_{ij}) + (1 - t_{ij}^{*}) \cdot \log(1 - t_{ij}),$$

Inference via distance metric (cosine sim, dot prod)
between Biomedical IERs
without fine-tuning
(*leverages quantized based*
efficient similarity search)

Using BIERs

(1) **Named Entity Disambiguation** (NED) on Clinical Entities.

Given a entity mention, context & set of candidate entities,
identify which of the candidates is the true one linked to the mention.

(1) **Named Entity Disambiguation** (NED) on Clinical Entities.

Given a entity mention, context & set of candidate entities identify which of the candidates is the true one linked to the mention.

Model	Test Acc. Dot Prod	Test Acc. Cosine Sim
BIER-PubMedBERT (ours)	80.1	**84.0**
BIER-SciBERT (ours)	76.4	77.3
BIER-BioBERT (ours)	71.9	75.9
Onoe and Durrett (2020)	63.6	69.8
Popular Prior	73.9	-
PubMedBERT (Gu et al., 2020)	77.6	-
SciBERT (Beltagy et al., 2019)	77.4	-
BioBERT (Lee et al., 2019)	77.9	-

Table 2: BIER zero shot test results vs Logistic Regression Baselines trained on task data for NED task

Using BIERs

(2) Entity label Classification for Cancer Genetics

Model	Test Acc. L2 Dist Dense	Test Acc. L2 Dist Sparse	Test Acc. Dot Prod Dense	Test Acc. Dot Prod Sparse
BIER-PubMedBERT	85.5	86.8	**88.2**	**87.5**
BIER-SciBERT	70.8	77.0	72.8	76.8
BIER-BioBERT	83.4	85.9	85.6	86.8
Onoe and Durrett (2020)	63.9	55.1	60.0	59.9
PubMedBERT	77.3	-	69.3	-
SciBERT	74.4	-	75.2	-
BioBERT	67.6	-	59.6	-

Table 3: Test accuracy on Cancer Genetics data using a nearest neighbor classifier (k=1) without fine-tuning based on sparse output or intermediate dense embeddings using L2 or Dot Product distance metrics.

Using BIERs

(2) Entity label Classification for Cancer Genetics

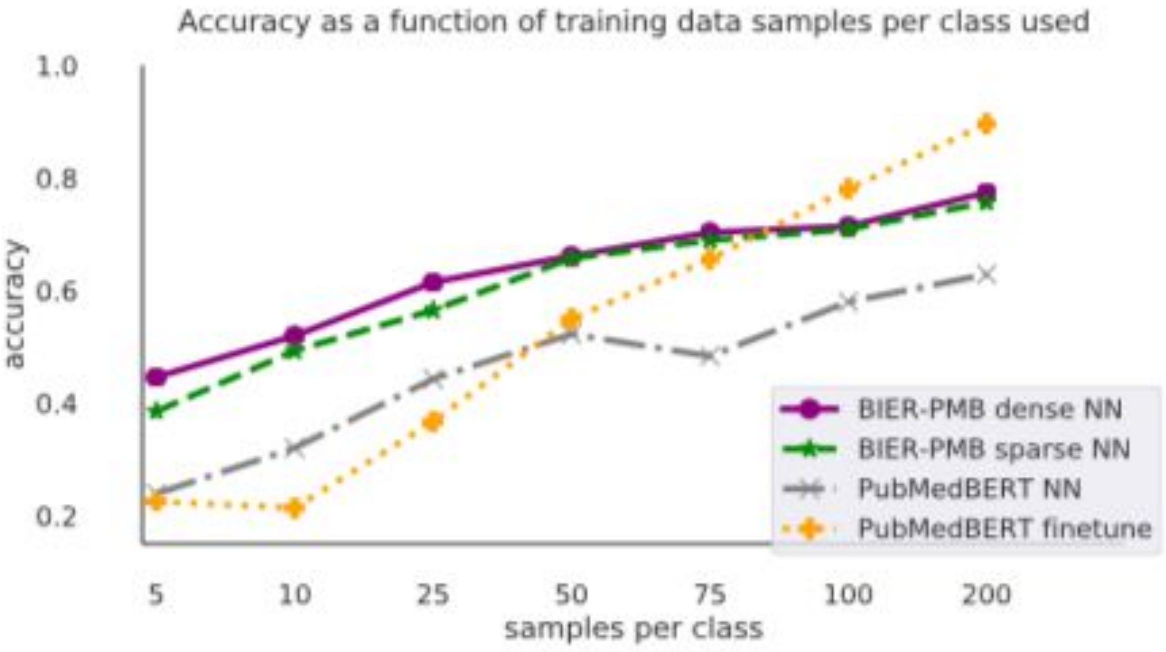
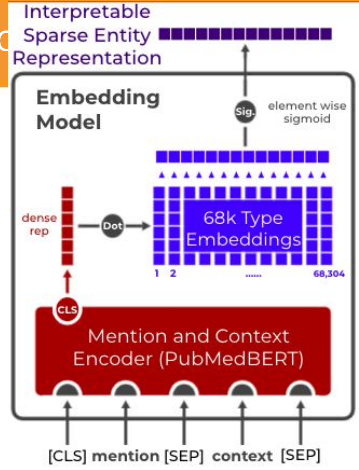


Figure 3: Results for the entity label classification task under varying amounts of supervision.

Debugging with BIERs

Allows for error analysis at the component level
to identify areas lacking in supervision
and/or possible changes to the type system.

Debugging with BIERs



Allows for error analysis at the component level to identify areas lacking in supervision and/or possible changes to the type system.

How well the model could have done had it known to fallback to using the intermediate dense embedding in cases where the sparse representation led to an **incorrect prediction**

	Test Acc.			
Task	Dense	Sparse	Combined	Δ
NED	84.0	81.0	**91.7**	+7.7
ELC	87.5	88.2	**91.9**	+3.7

Table 5: Results for both tasks showing improvements that could have been achieved by combining intermediate dense and interpretable sparse output embeddings generated by the same BIER-PubMedBERT model.

Motivation for **future work** on developing a dynamic approach to making predictions that is a function of model confidence.

Error analysis using BIERS

context: The presence of activating TSH-R mutations has also been demonstrated in differentiated **thyroid carcinomas.** At present, the percentage of such a modification is low, unless referred to selected series of tumors.

mention: **thyroid carcinomas**

label: **Cancer**

Sparse NN model pred	Dense NN model pred
thyroid **(label: Organ)**	**esophageal carcinomas** **(label: Cancer)**
Types	**Types**
('gland', 0.99965), ('thyroid', 0.99932), ('rtt', 0.999), ('head_and_neck_cancer', 0.99093), ('neck', 0.97243), ('head_and_neck_anatomy', 0.93763), ('head', 0.86131), ('squamous-cell_carcinoma', 0.0024), ('ingredient', 0.00078), ('thyroid disease', 0.00047), ('nitrous_oxide', 0.00034), ('thyroid cancer', 0.0003), ('endocrine diseases', 0.00019),	('thyroid cancer', 0.99994), ('squamous-cell_carcinoma', 0.9998), ('thyroid', 0.99925), ('cancer', 0.99133), ('gland', 0.99039), ('nitrous_oxide', 0.01965), ('pancreatic_cancer', 0.00152), ('neck', 0.00023), ('thyroid_neoplasm', 0.00019), ('rtt', 0.00014), ('endocrine diseases', 2e-05), ('head', 1e-05), ('malignancy', 1e-05),

Completed Work

Learning Dense Representations for Entity Retrieval. (CoNLL 2019)	Constructed a **dual mention-entity encoder** that learns dense representations for efficient neural **Entity Retrieval** with an **in-process, iterative hard-negatives procedure** for **model learning and inference time inspection**.
Deep Classification of Time-Series Data with Learned Prototype Explanations. (ICML 19)	Adapted a **prototypical autoencoder** classifier to be compatible with **time series data**; allowing for **tunable prototype diversity** and improved accuracy and **global and instance level explanations**.
Biomedical Interpretable Entity Representations. (ACL-IJCNLP 2021)	Learned a distantly supervised entity type system and data set for use in training a **Biomedical Interpretable Entity model** whose representations exist in a **semantically meaningful vector space** & whose **predictions may be diagnosed** with an oracle method.

Summary

This proposal shows in-process diagnostic techniques are useful for sequential data tasks both in accuracy & interpretability.

1. We constructed a **dual mention-entity encoder** that learns dense representations for efficient neural Entity Retrieval with an **in-process, iterative hard-negatives procedure** that can be inspected.

Summary

This proposal shows in-process diagnostic techniques are useful for sequential data tasks both in accuracy & interpretability.

1. We constructed a **dual mention-entity encoder** that learns dense representations for efficient neural Entity Retrieval with an **in-process, iterative hard-negatives procedure** that can be inspected.

2. We adapted a **prototypical autoencoder** classifier to be compatible with **time series data**; allowing for **tunable prototype diversity** for improved **global and instance level explanations**.

Summary

This proposal shows in-process diagnostic techniques are useful for sequential data tasks both in accuracy & interpretability.

1. We constructed a **dual mention-entity encoder** that learns dense representations for efficient neural Entity Retrieval with an **in-process, iterative hard-negatives procedure** that can be inspected.

2. We adapted a **prototypical autoencoder** classifier to be compatible with **time series data**; allowing for **tunable prototype diversity** and improved **global and instance level explanations**.

3. We learned a distantly supervised entity type system and data set for use in training a **Biomedical Interpretable Entity model** whose representations exist in a **semantically meaningful vector space** & whose **predictions may be diagnosed** with an oracle method.

List of Publications

- Garcia-Olano, D., Onoe, Y., Baldini, I., Ghosh, J., Wallace, B., Varshey, K. "Biomedical Interpretable Entity Representations". Findings of the Association for Computational Linguistics (ACL-IJCNLP), Bangkok, Thailand, 2021

- Gillick, D., Kulkarni, S., Lansing, L., Presta, A., Baldridge, J., Ie, Eugene., Garcia-Olano, D. "Learning Dense Representations for Entity Retrieval". Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL), Hong Kong, China, 2019.

- Garcia-Olano, D., Gee, A., Ghosh, J., Paydarfar, D. "Explaining Deep Classification of Time-Series Data with Learned Prototypes". Proceedings of the 4th International Workshop on Knowledge Discovery in Healthcare Data co-located with International Joint Conference on Artificial Intelligence, (IJCAI) , Macao, China, 2019

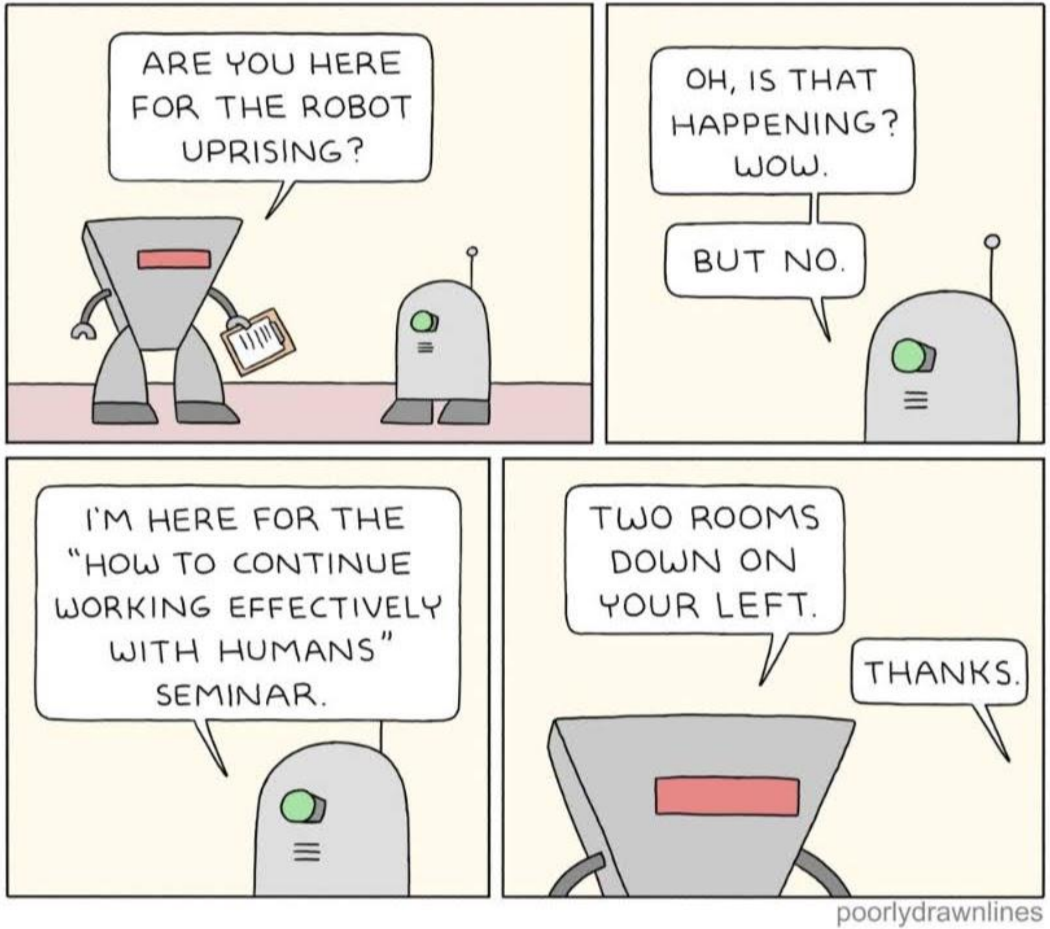
- Garcia-Olano, D., Gee, A., Ghosh, J., Paydarfar, D. "Deep Classification of Time-Series Data with Learned Prototype Explanations". Proceedings of the 36th International Conference on Machine Learning (ICML), Long Beach, California, PMLR 97, 2019

- Sankaran, K., Garcia-Olano, D., Javed, M., Alcala-Durand, M., De Unánue, A., van der Boor, P., Potash, E., Avalos, R., Encinas, L., Ghani, R., "Applying Machine Learning Methods to Enhance the Distribution of Social Services in Mexico". Presented at UChicago Data Science for Social Good. arXiv:1709.05551. 2017.

- Garcia-Olano, D. Arias, M, Larriba Pey, J. "Automated construction and analysis of political networks via open government and media sources". European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML). Riva del Garda, Italy, 2016

- Paz-Ortiz, García-Olano, D., Gay-García, C. (2015. July) "Term-frequency Inverse Document Frequency for the Assessment of Similarity in Central and State Climate Change Programs: An Example for Mexico". In Proceedings of the 5th International Conference on Simulation and Modeling Methodologies, Technologies and Applications (MSCCES-2015), Colmar, France 2015.

Thank you!


ARE YOU HERE FOR THE ROBOT UPRISING?
OH, IS THAT HAPPENING? WOW.
BUT NO.
I'M HERE FOR THE "HOW TO CONTINUE WORKING EFFECTIVELY WITH HUMANS" SEMINAR.
TWO ROOMS DOWN ON YOUR LEFT.
THANKS.
poorlydrawnlines