CS8803-MDS

Final Presentation

12/07/22

Group 9

Efficient Query-to-Data Distance Distribution Estimation Add-on for Multi-Probe LSH

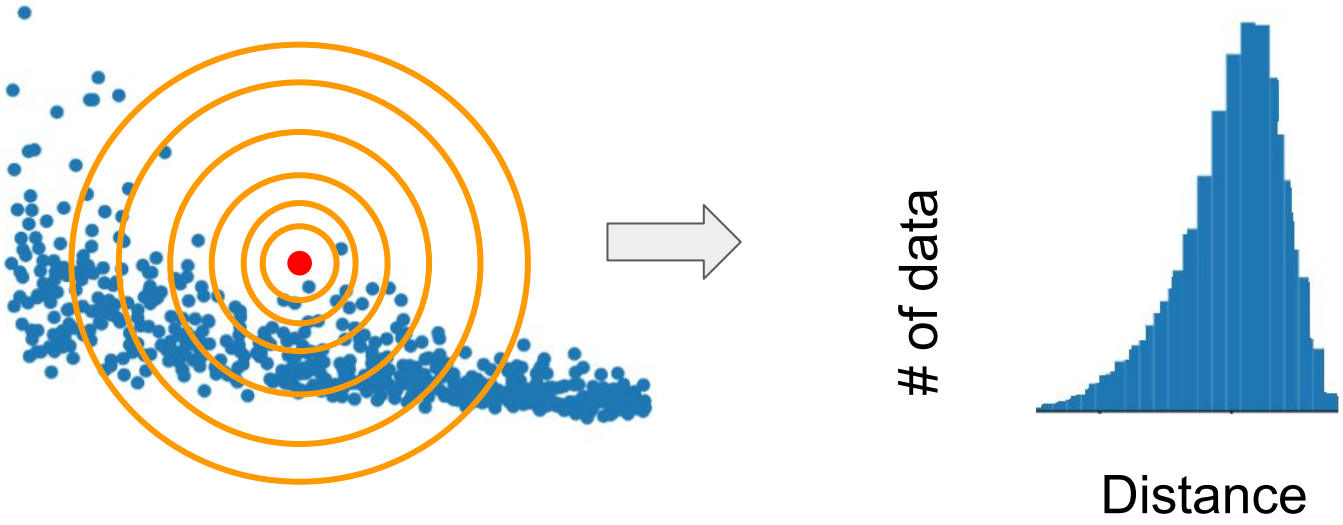
Jingfan Meng

Q2D Distance Distribution Estimation (QDDE)

- It is an important problem to sketch the **local** distribution of a large, high-dimensional dataset around a given query.

Q2D Distance Distribution Estimation (QDDE)

- QDDE asks for the number of data points in each concentric shell centered at query, i.e., a Q2D distance histogram.



Q2D Distance Distribution Estimation (QDDE)

- Efficient QDDE has wide applications in kernel density estimation (KDE), fair near neighbor (FNN) sampling, etc.
- However, it is costly to compute query-to-data (Q2D) distances. For example, on a 1B-scale dataset, it takes several minutes to compute for all data. Even if we sample 1% of data, it still takes seconds.
- The applications above require high-accuracy on **close-range** (neighborhood), so uniform sampling is a bad idea.

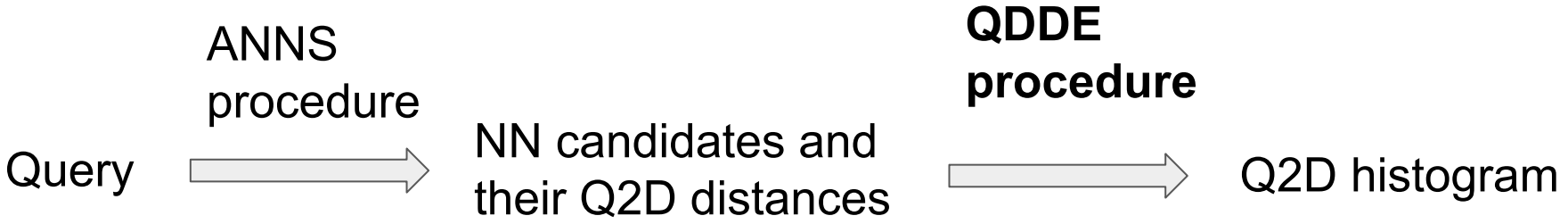
QDDE: The Bit

- Existing solutions use an **ad-hoc index** that is based on locality sensitive hashing (LSH) so that the sampling rates are higher on the close range. This leads to higher accuracy on the close range.
- Limitations:
  - Extra storage space for the ad-hoc index.
  - The query times are still quite long (for computing Q2D distances.

QDDE: The Flip

- We reuse (any) LSH-based approximate nearest neighbor searching (ANNS) index that has been applied to many high-dimensional database systems.
- Benefits:
  - No extra space usage for index.
  - The extra query times are very short (since Q2D distances are computed in ANNS procedure).
  - Our approach is generally applicable to many ANNS solutions.

QDDE: Proposed Solution



- The first step of QDDE is to count the number of candidates, say c, in each histogram bin (distance range).
- If the sampling rate on this bin is p, then the estimated bin height (# of data points) is **c/p**.
- Repeat for each bin, then we get a histogram.

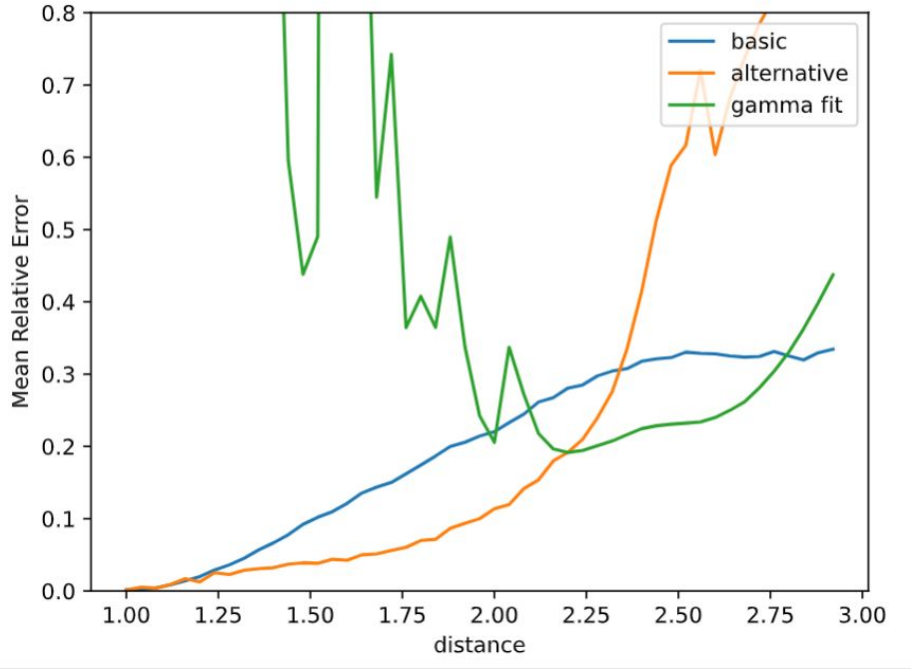
QDDE: Proposed Solution

- How do we know the sampling rates for ANNS candidates?
  - Basic solution: theoretical calculations.
  - Alternative solution: we note that ANNS solutions often use multiple hash tables (T), which return independent sets of candidates.

$$\text{Estimated p of T1} = \frac{\text{\# Candidates from T1 } \mathbf{and} \text{ at least another table}}{\text{\# Candidates from at least another table}}$$

QDDE Evaluation

- High accuracy on the close range (on GIST dataset).



Legends

Blue: Basic
Orange: Alternative
Green: Baseline

QDDE Evaluation

- Extra query time (in milliseconds) over ANNS is very short.

Dataset	Size	ANNS	Basic	Alternative
MNIST	69K	12.2	0.077	0.15
GIST	1M	286.65	2.13	7.26
Deep	10M	299	4.45	50.3

Conclusion

- By reusing the existing index for ANNS, we can estimate Q2D distance histograms both accurately and efficiently.
- Both of our schemes significantly outperform the baseline in terms of accuracy on the close range.
- The extra query time is at most 7.5% that of the underlying ANNS procedure.

Group 11 - Evaluating Language Generation for with Multi-Class Cyberbullying

Bojun Yang

Cyberbullying is Bad

- 87% of young (12-17 y.o.) people have witnessed some kind of cyberbullying, 36.5% of people feel they have been cyberbullied, and 17.4% accept it happened to them within the past month [1]. Estimated 90% of cyberbullying goes unreported [2].

1. https://cyberbullying.org/summary-of-our-cyberbullying-research 2. https://www2.uwstout.edu/content/lib/thesis/2009/2009aunen.pdf

Problem Statement (bit and flip)

- 11 results for binary cyberbullying datasets and only 1 dataset for multi-class cyberbullying on Kaggle
- Cyberbullying is a emerging problem, with no good solution
  - Prominent social media platforms like Facebook, Twitter, Instagram, Snapchat, etc have resources and passive reporting mechanisms but none of them have active anti-cyberbullying functions
- No publications found for multi-class cyberbullying work (detection, augmentation, NLG, evaluation) → use existing techniques to address the lack of work done in multi-class cyberbullying

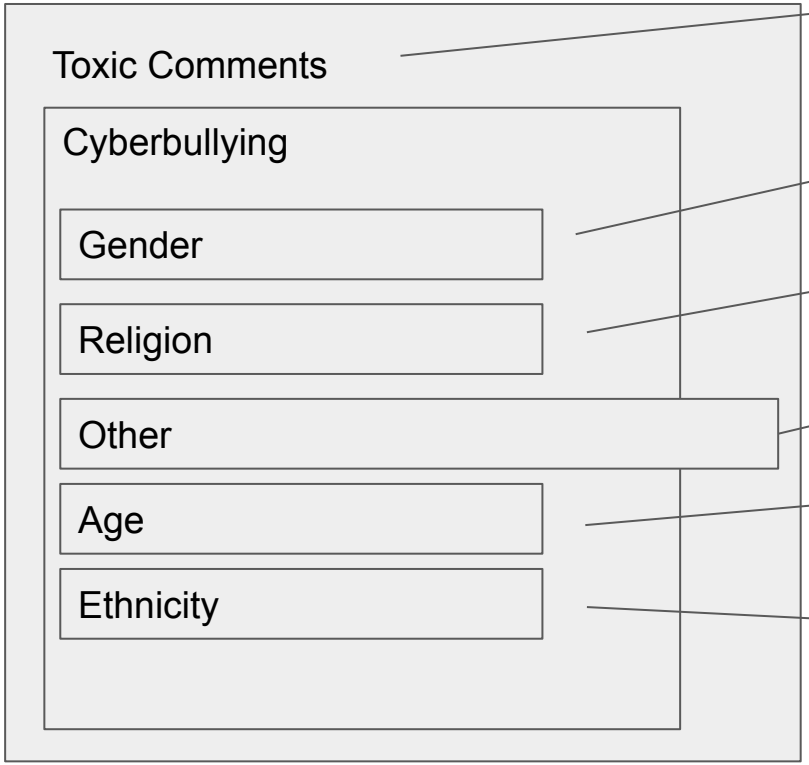
Contributions

- Evaluate NLG methods in generating cyberbullying data that targets a specific quality of the victim (age, ethnicity, gender, religion)
  - Data augmentation
  - Applying NLG to a new dataset/field
- Applying classification methods to multi-class cyberbullying data
  - No literature has done this before
  - Only “multi-class” classification labeled data with different degrees of severity
- Using topic modeling to evaluate generated text
- Ethics [1]
  - Out of a filtered 222,774 tweets that mention WHO from 1/20/2020 - 4/23/2020, 21% of it was labeled as toxic → 46,782 toxic tweets produced specifically mentioning one topic for 4 months
  - We produce 900 generated per positive label per mode → 900*5*2 = 9,000 total generated

1. https://reutersinstitute.politics.ox.ac.uk/volume-and-patterns-toxicity-social-media-conversations-during-covid-19-pandemic

“Tweets either describe a cyberbullying event or are the offense themselves”

Dataset: cyberbullying vs. toxic comments



“F–k off geek, what I said is true. I’ll have your account terminated.”

"RT @- I'm sorry but I can't handle women commentators or women talking about sports on ESPN #NotSexist"

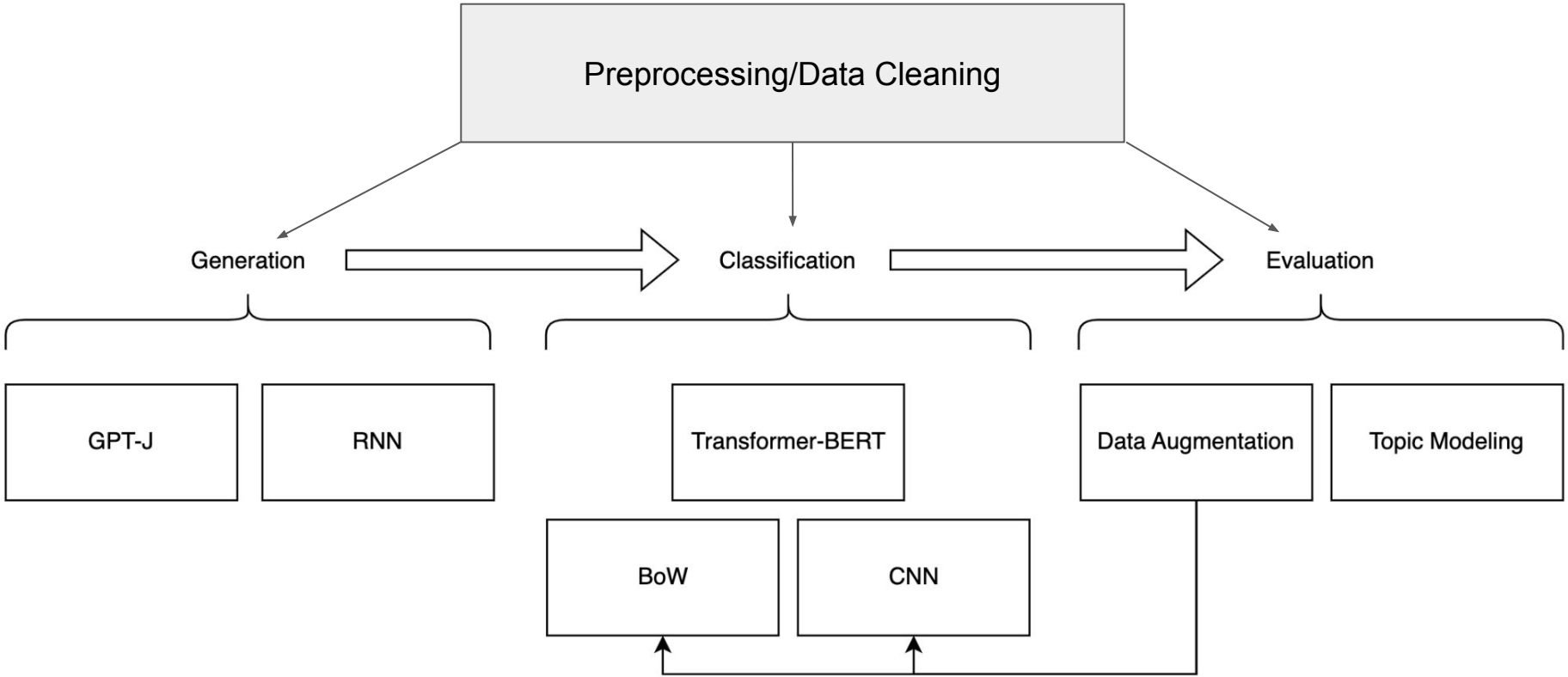
“The terrorist nation of Pakistan has started recruiting kids to wage jihad against India…”

“@- @- who knows. who cares? just more bullshit they are passing around. i don't pay their claims any attention.”

"@- Yeaah I am! and im younger than you, you're being bullied by someone younger, thats even funnier. >:D ahaha"

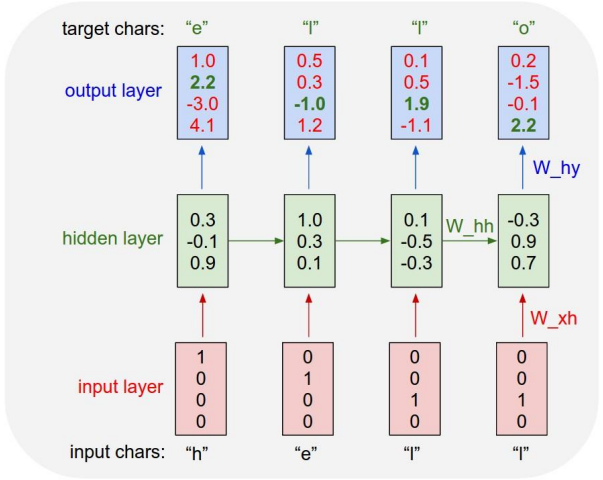
“I was really close to laying out the new black secuirty guard today, stupid dumb f___ n_____ pissed me off.”

Overview

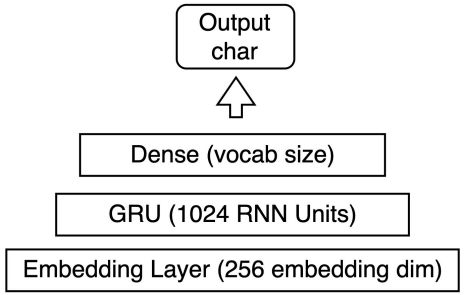

Preprocessing/Data Cleaning
Generation
Classification
Evaluation
GPT-J
RNN
Transformer-BERT
Data Augmentation
Topic Modeling
BoW
CNN

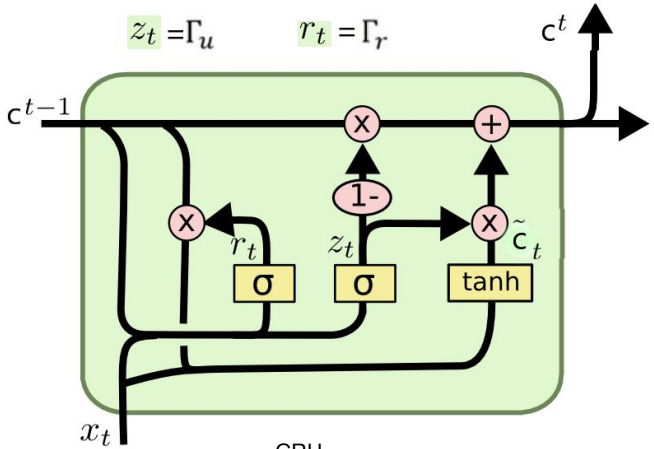
Generation - RNN

- Outputs are influenced by entire past history of inputs using the hidden/RNN units
- We use specifically Gated Recurrent Unit layer (GRU)
  - Uses 2 gates (update and reset) to remember important information from previous cell states



RNN



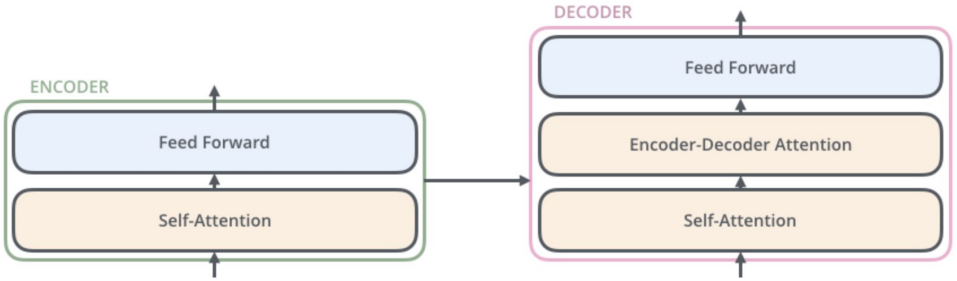


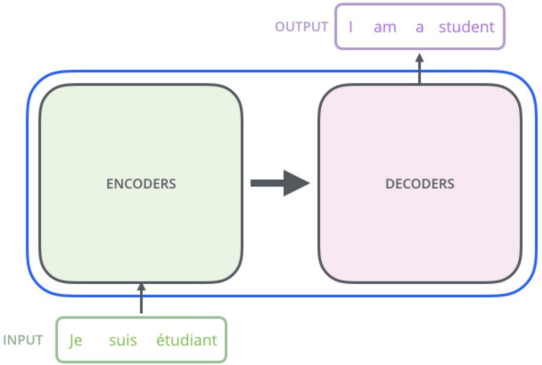
GRU

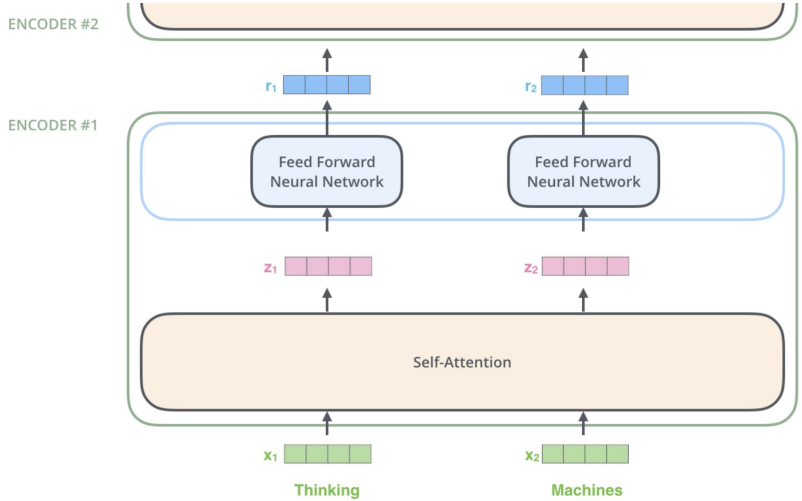
Generation - GPT-J

- 6B parameter text completion model trained on The Pile (dataset known to contain profanity, lewd, and otherwise abrasive language)
- Probabilistic model → chooses next word using sampling
  - Temp: how confident the model is in words it views as likely
  - Top-p: how much of the list of possible words get excluded
    - 0.9 means the 10% most unlikely words in the list will get excluded
- Transformer language model

Transformer Architecture



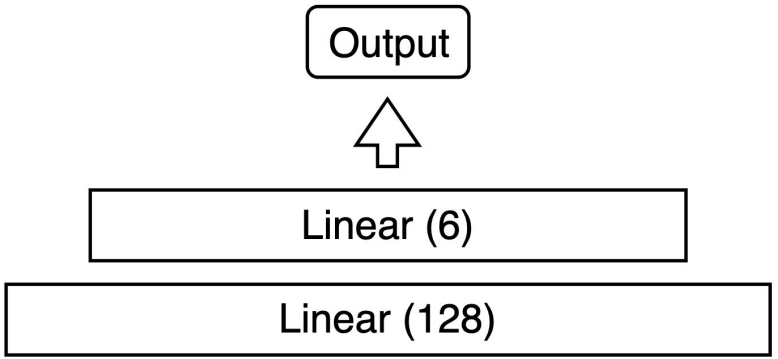




Classification - Bag of Words

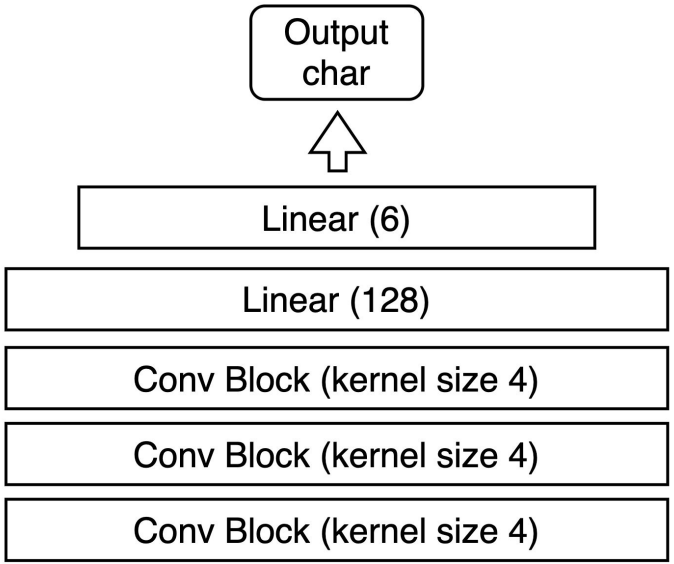
Document	the	cat	sat	in	hat	with
the cat sat	1	1	1	0	0	0
the cat sat in the hat	2	1	1	1	1	0
the cat with the hat	2	1	0	0	1	1

- Vectorizes each tweet as a vocabulary-length vector with elements corresponding the the amount of times a token appears in the tweet
- Does not capture sequence information



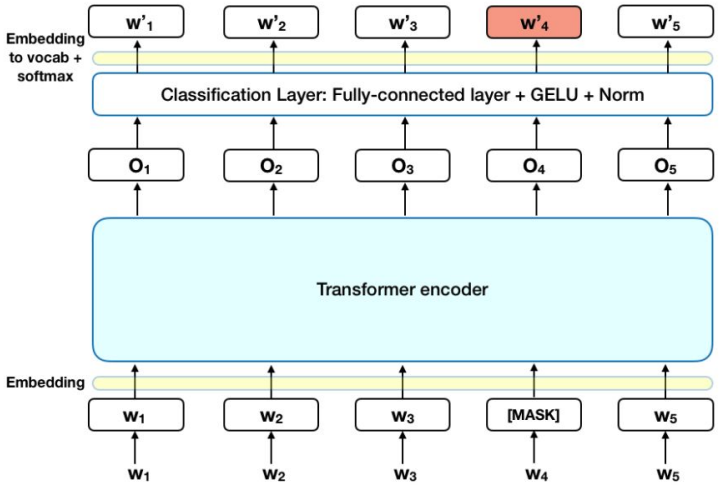
Classification - CNN

- Convolves with a 1d kernel over the input
- Tries to capture some relationship between words



Classification - Transformer-Bert

- Bidirectional Encoder Representations from Transformers (BERT)
- Attention works similar in Transformer models → can look other words to make decisions
- 2 implementations
  - Huggingface's AutoModelForSequenceClassification
  - Using BERT to get embeddings and running it through a MLP



Evaluation - BERTopic

1. Converts documents to vector representations using BERT
2. Cluster Embeddings
    a. Reduce dimensions of vector representations using UMAP (i.e. embedding of 384 to 5)
    b. Find semantically similar documents with HDBSCAN and extract topics
3. Topic Representation
    a. Tokenize each topic into BoW representations
    b. Calculate words that are interesting to each topic using class-based TF-IDF (Adjusted TF-IDF to work on a cluster/topic/categorical level instead of document level)

Results - Data Augmentation

Full Data	F1 score							
	Label	BoW	CNN	Transformer	BoW - Raw Aug	CNN - Raw Aug	BoW - Clean Aug	CNN - Clean Aug
not	0	0.5778	0.5169	0.5796	**0.587**	0.5297	0.5863	0.5338
gender	1	0.8588	0.8436	**0.8757**	0.8588	0.8435	0.8602	0.8464
religion	2	0.9557	0.9346	0.9547	**0.9584**	0.9472	0.9571	0.9441
other	3	**0.6563**	0.6533	0.642	0.6463	0.62	0.6346	0.6284
age	4	**0.98**	0.9715	0.9753	0.9775	0.9672	0.9756	0.9638
ethnicity	5	0.9798	0.9631	**0.9867**	0.9798	0.9674	0.9798	0.9642
	avg	0.834	0.8143	**0.8365**	0.8331	0.813	0.8304	0.8145

- RNN aug have similar accuracies as GPT-J aug
- RNN CNN avg accuracy: 0.8205

Results - Data Augmentation

Partial	F1 score							
	Label	BoW	CNN	Transformer	BoW - Raw Aug	CNN - Raw Aug	BoW - Clean Aug	CNN - Clean Aug
not	0	0.8516	0.8272	**0.8663**	0.8496	0.8411	0.8536	0.8286
gender	1	0.8855	0.8579	**0.905**	0.8848	0.873	0.8904	0.8673
religion	2	0.9544	0.9461	0.952	**0.957**	0.9444	0.957	0.9494
age	4	0.9799	0.9767	**0.9861**	0.9774	0.9709	0.9786	0.9703
ethnicity	5	**0.9836**	0.9792	0.9817	0.9817	0.9792	0.9823	0.9735
	avg	0.9303	0.9162	**0.9376**	0.9293	0.9208	0.9315	0.917

Results - Common topics between input and raw augmented

not	not, of, que, you, and, to, por, in, de, se, is, that
gender	but, and, call, is, are, **woman, gay,** funny, not, about, be, **feminist**, my, said, his, **women**, fu____g, it, **rape**, you, **her**, **man**, jokes, racist, **men**, joke, an, for, announcer, **b_ch**, as, can
religion	the, **palestinian, muslims, christian,** was, and, she, to, **muslim**, is, are, not, **palestinians, india,** president, on, **islam**, it, they, you, her, racist, we, of, in
other	he, about, from, voice, games, of, tweets, you, her, have, to, twitter, am, is, that, are, women
age	the, he, **school**, like, was, and, she, is, are, lesbian, my, his, on, **bullied**, it, this, they, you, her, mean, weight, for, in, **girls**, bully
ethnicity	the, **n____rs,** obama, was, and, to, is, that, are, ass, f__k, not, **black**, **racism**, president, **white**, my, him, his, by, it, **n___er**, they, you, her, sh_t, your, so, of, in

Results - Common topics between input and cleaned augmented

not	do, he, and, she, to, at, is, that, not, about, my, his, on, this, they, you, her, shirt, bullying, your, of, que, in, am, de, se
gender	**gay**, when, not, it, for, you, and, **her**, my, is, are, **women**
religion	the, he, **muslims**, right, **christian**, was, and, she, to, is, are, vote, not, **india**, white, **jews, islam, hindus**, it, you, her, racist, for, of, in
other	he, to, here, is, their, that, are, wear, me, women, **abuse**, by, im, best, you, her, of, am, **bully**
age	the, he, **school**, **high**, got, was, and, she, like, is, are, now, me, about, my, his, him, on, im, bullied, it, this, they, you, her, **girl**, for, in, **bully**
ethnicity	the, **n____rs**, obama, was, and, to, at, first, is, ass, are, that, f__k, not, **black**, dumb, president, people, **white**, my, his, him, by, it, **n____r**, they, you, her, your, **racist**, we, so, b__ch, of, as, in, yall

Results - Common topics between input and RNN augmentation (partial)

gender	**rape**, and, joke, is, rt, are, **gay**, not, **sexist**, jokes, but, you, it, **b__ch**, for, **female**, am, call
religion	and, are, you, **islamic**, racist, is, not, idiot, the, she, **islam, christian**, support, he, **terrorists, white**, for, **muslim**, who, in, **india**, right, **muslims**, of, to, **palestine**, idiots, it
ethnicity	and, are, you, her, is, the, f_ck, ass, **n___ers**, rt, **black**, people, dumb, **racism**, f__king, that, **white**, **n___er**, in, obama, of, to

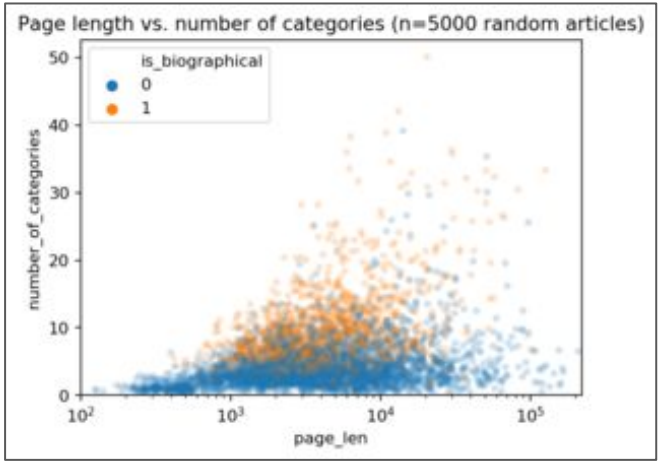
Results - RNN

- Seems at least as good as GPT-J from observations
- Random generated tweet RNN for gender
  - "Hey, probably gay can insult, and 2936: angress mental illnesses and guy planton making sh_tty rape jokes...but f__k Gay lol He's being block, you’re a compliment.\nCan someone using me? @WillTrail @RamesBeter70 This is exactly what is outrage. Rape jokes, said Be? Probably jokes"
- Random extended tweet GPT-J for gender
  - “**RT @TayRaeAye: Call me sexist or whatever but men shouldn't telecast women sports, and women shouldn't telecast men sports. #sorrynotsorry** #1am #hilariouslyashamed #altoff #sports
    @orlandotimes Julie Chatard trying to be positive isnt it”

Group 2

QuickScatter
Online Sampling Analysis For Scatter Plots
Eric Martin, Akshay Iyer

The Problem



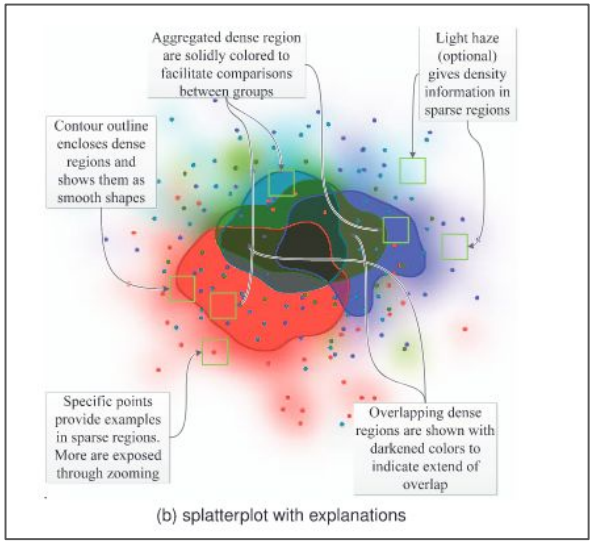
- Scatter Plots Are Popular
  - Good For...
    - Correlation / Trend Analysis
    - Cluster / Outlier Detection
    - Density Estimation
    - Relating High Dimensional Data (PCA)

- Explosion of data has reduced the efficacy of scatter plots.
  - Loading 1M data is latent.
  - Rendergin 1M data is computationally expensive.
  - Overdraw Issue -> How to view so much data?

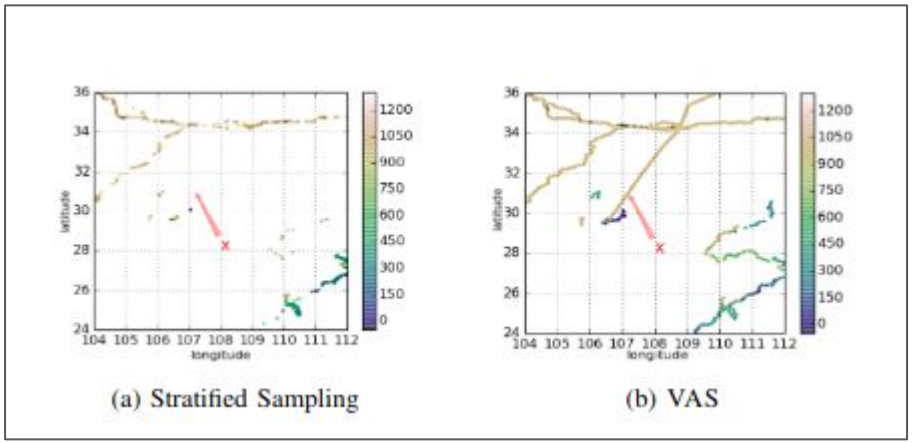
- Can scatter plots be adopted for exploratory data analysis?

Prior Work To Adopt Scatter Plots For Larger Datasets...

- Splatter Plot -> Novel Data Abstraction
- Visualization Aware Sampling -> Offline Sample Construction
- Different Sampling Methods
  - Farthest Point Sampling
  - Blue Noise Sampling
  - Density Biased Sampling
  - etc...



(b) splatterplot with explanations



(a) Stratified Sampling (b) VAS

Shortcomings of Previous Approaches...

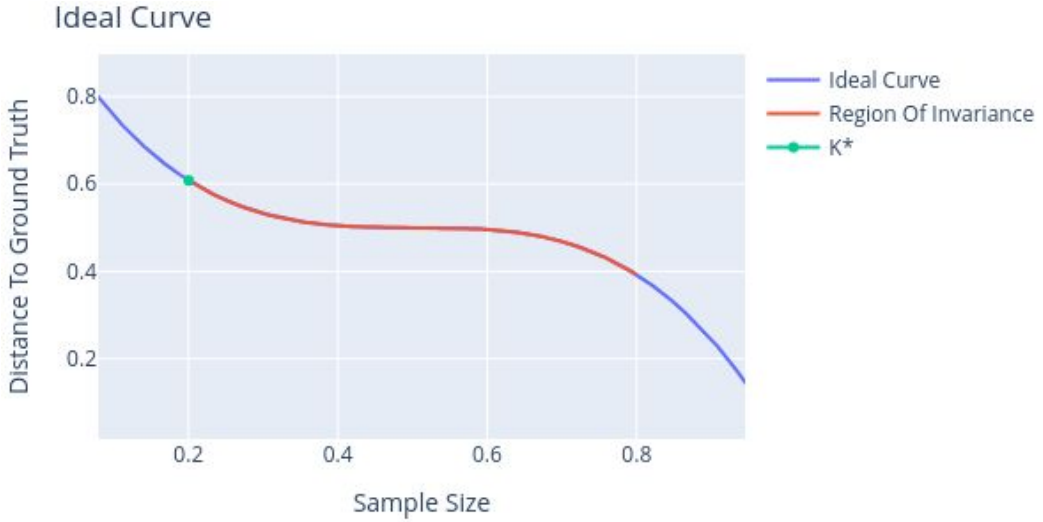
- Splatter Plot
  - Tries to visualize all points.
  - Requires GPU support.

- Visualization Aware Sampling (Offline Sampling)
  - Limited Exploration Space
  - Expensive To Build Samples
  - Requires Backend Server Support

- Complex Sampling...
  - Not well known.
  - Can be time consuming.
  - Not always valid for online sampling.

The Bit-flip...

- **The Bit:** Earlier works assumes overplotting is an ‘absolute bad’ to be avoided at all costs.
- **The Flip:** Can we turn overplotting into a positive feedback signal to uncover ideal sampling rates any novice user can use?
- Explore different scenarios to understand when oversampling of data points occurs. Report the maximum threshold for sampling rate given any significantly large scatterplot.



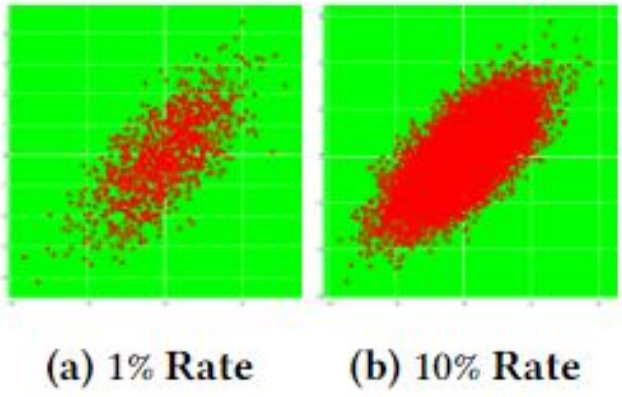
The Ideal User Interface...

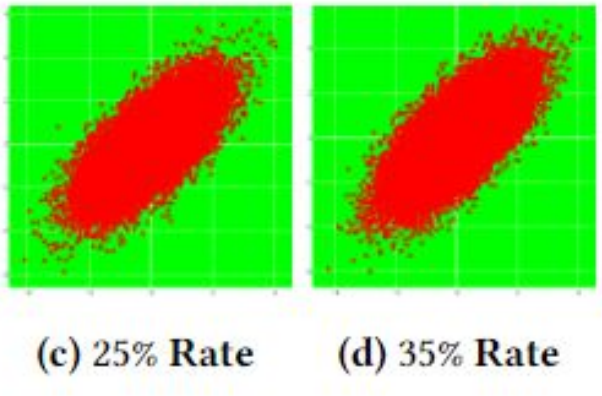
```
sample_rate = find_effecitve_communicable_bandwidth(
    sampling_method="Uniform",
    task="Trend",
    dataset_size=size_of_target_dataset,
    plot_width=500,
    plot_height=500,
    marker_radius=5)

data_to_plot = submit_interactive_query(
    dataset="url_to_database",
    query=query, rate=sample_rate)
```

✓ 0.6s

Technical Details I

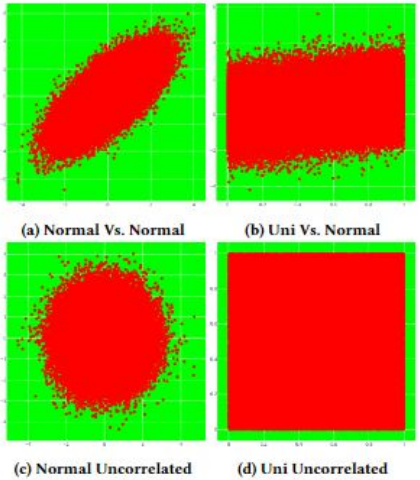
- Establish this idea of K* – a value much less than total sample points where sampling beyond this point does not give any better results
- Two types of metrics: image-based and data-based
- Image-based:
  - Manhattan Distance, RMSE (compare pixels between sampled and ground truth image)
  - SSIM (computer vision derived metric to analyze how close images are)
- Data-based:
  - VAS–calculus based loss objective function between sample and full dataset, using projections





Technical Details II

- Main two tasks were trend analysis and cluster detection

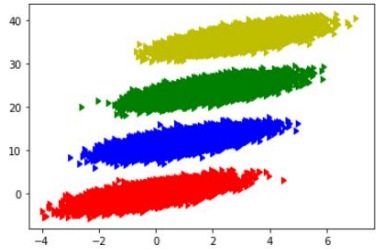


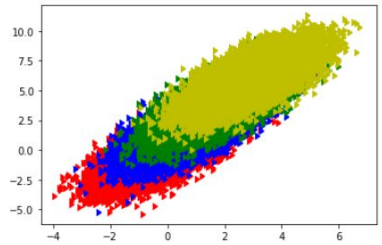
Trend Analysis

- Generated 4 different data distributions with varying degrees of randomness
- Examined how correlation aligned with the other metrics and visually derived a K* value

Cluster Detection

- Generated a total of 8 different graphs, of 2-5 clusters with both overlapping and non-overlapping components
- Looked at how the presence of clusters led to any differences and how the metrics reflected such changes





Results... The Opposite of Expectation.
Trend Analysis. Plot Size (500, 500). Marker Radius = 5

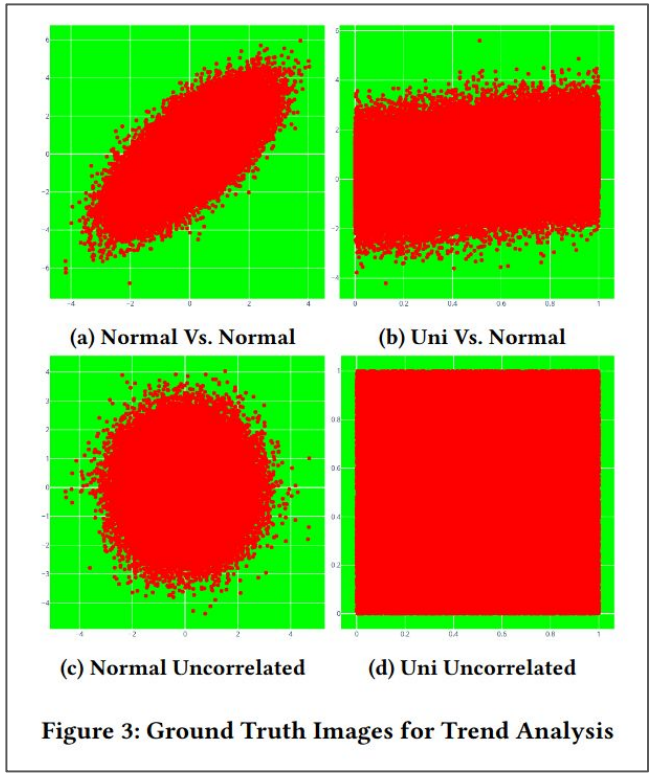
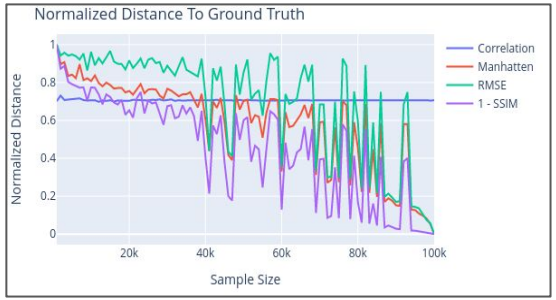
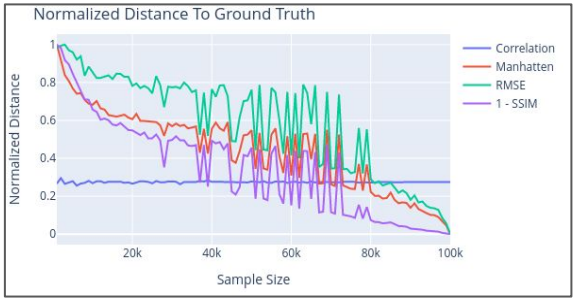


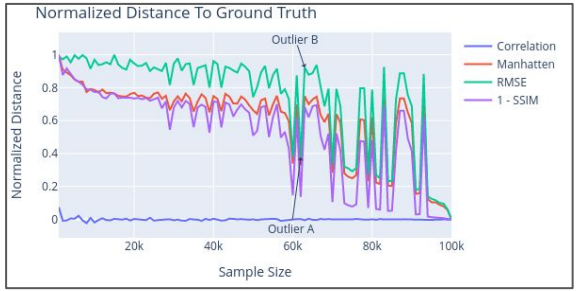
Figure 3: Ground Truth Images for Trend Analysis



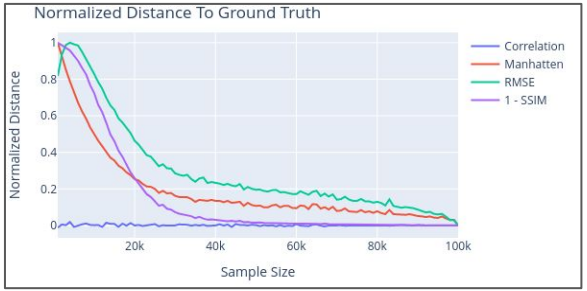
Normal Vs Normal



Uniform Vs Normal

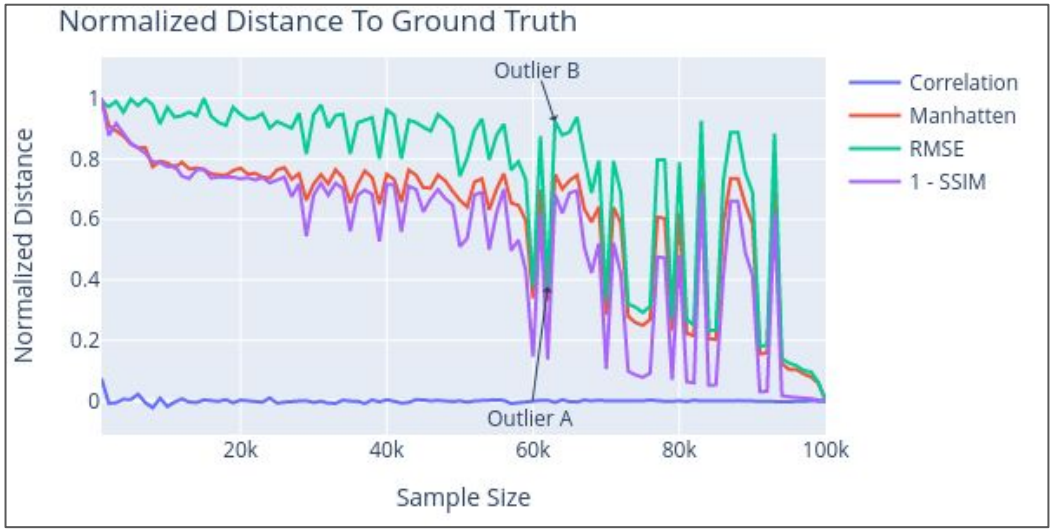


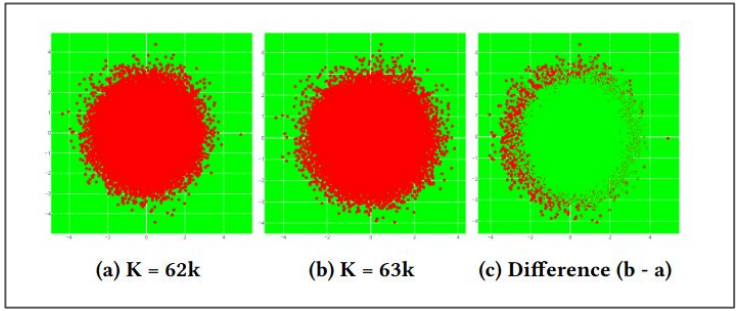
Normal Uncorrelated



Uniform Uncorrelated

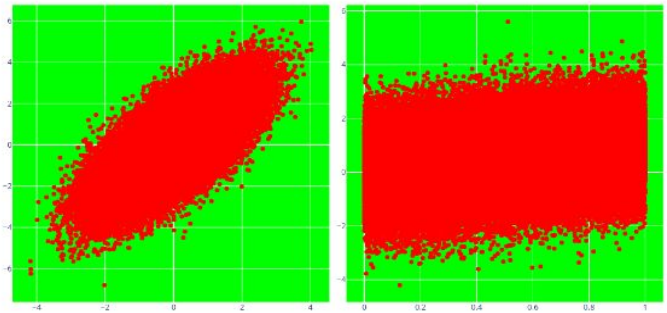
Results... A Closer Look At The Volatility





The Image Difference From Going To 63% Percent Sample From A 62% Sample.

Results: VAS Data-Based Metric

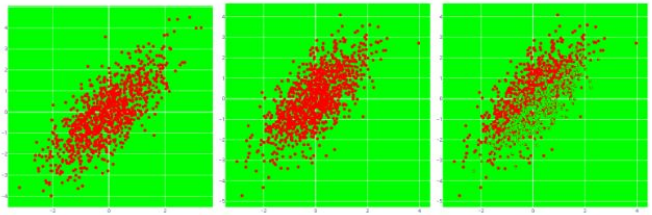


(a) Normal Vs. Normal (b) Uni Vs. Normal

(c) Normal Uncorrelated (d) Uni Uncorrelated

Figure 3: Ground Truth Images for Trend Analysis

Figure 8: VAS Loss Curves

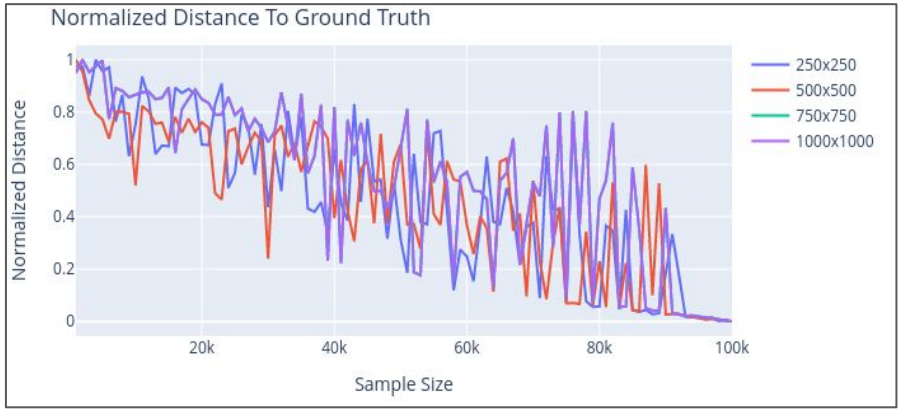


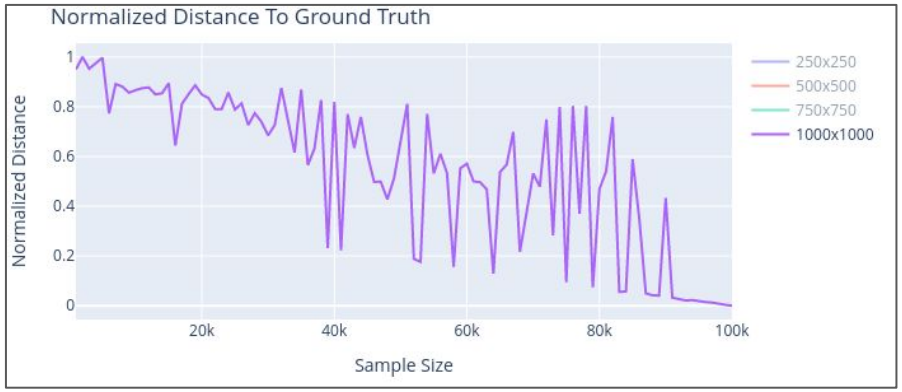
(a) $K = 1000$ (b) $K = 1100$ (c) Difference

Figure 9: Image Differences For VAS

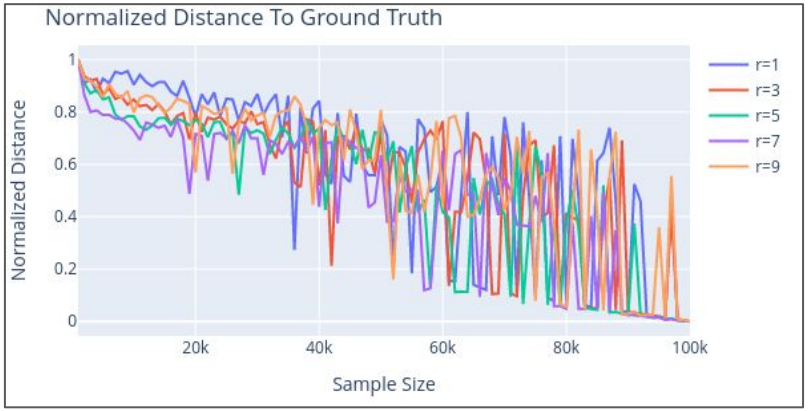
Further Results For Tend Analysis...

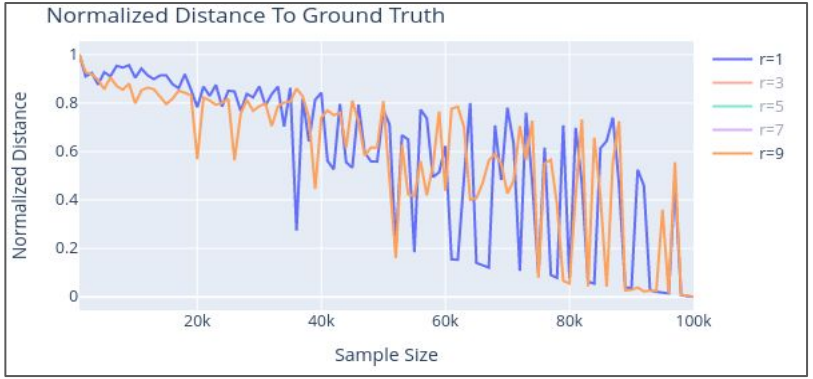
Changes In Plots Sizes



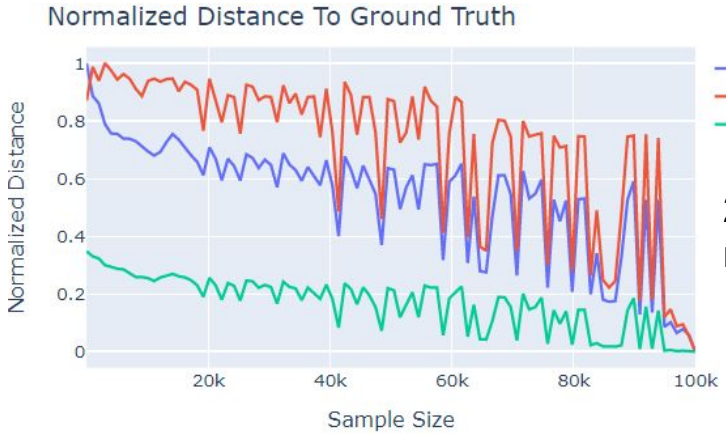


Changes In Marker Radius

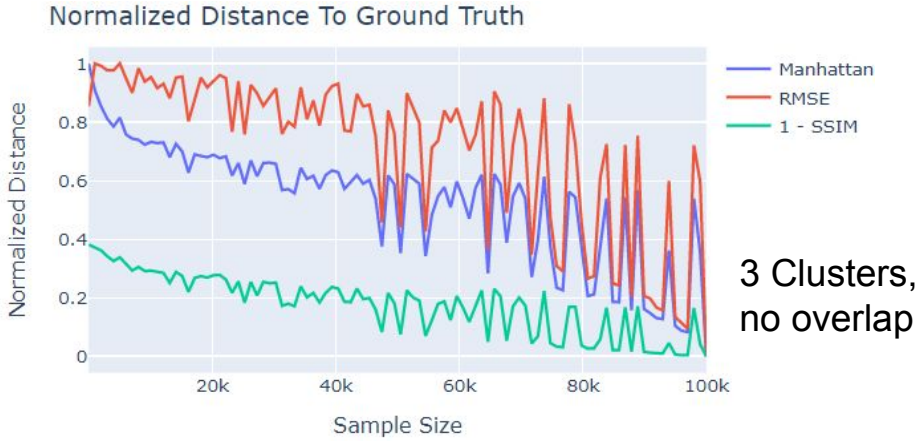




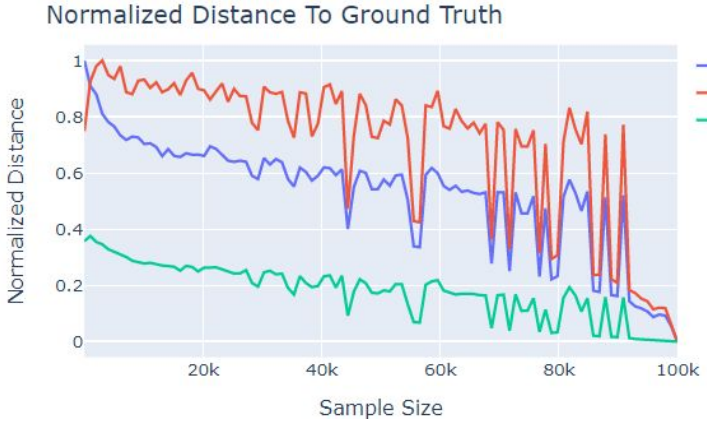
Cluster Graphs



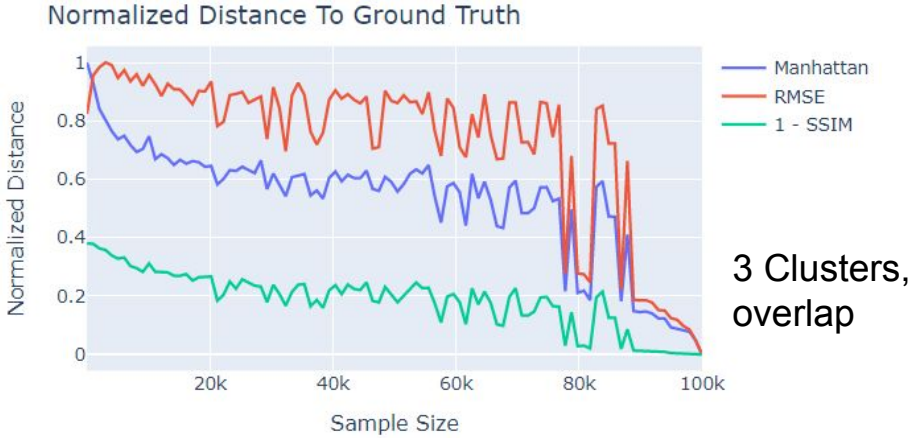
2 Clusters, no overlap



3 Clusters, no overlap

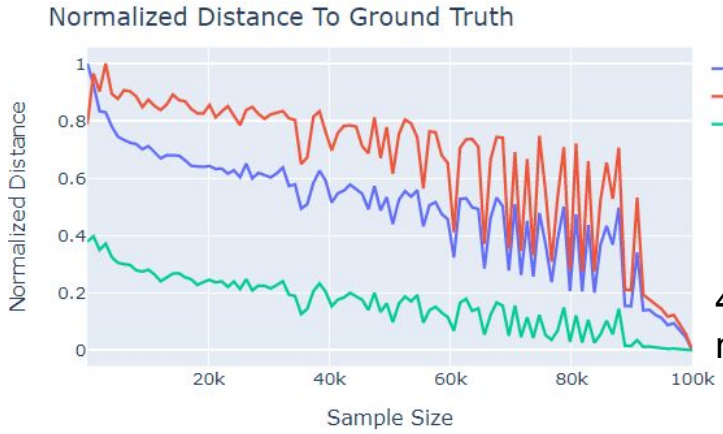


2 Clusters, overlap



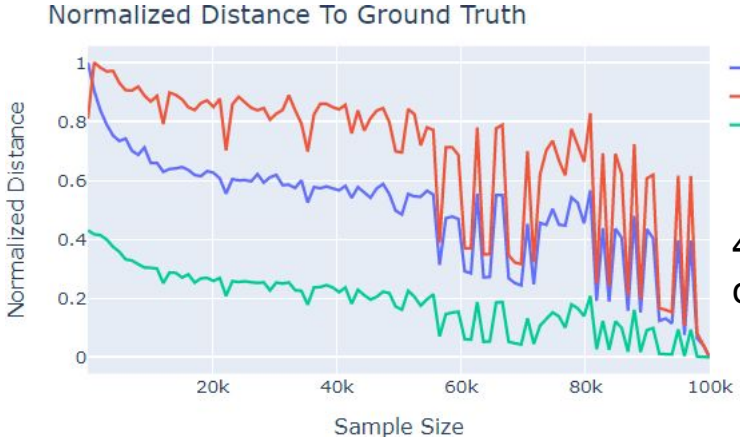
3 Clusters, overlap

Cluster Graphs II

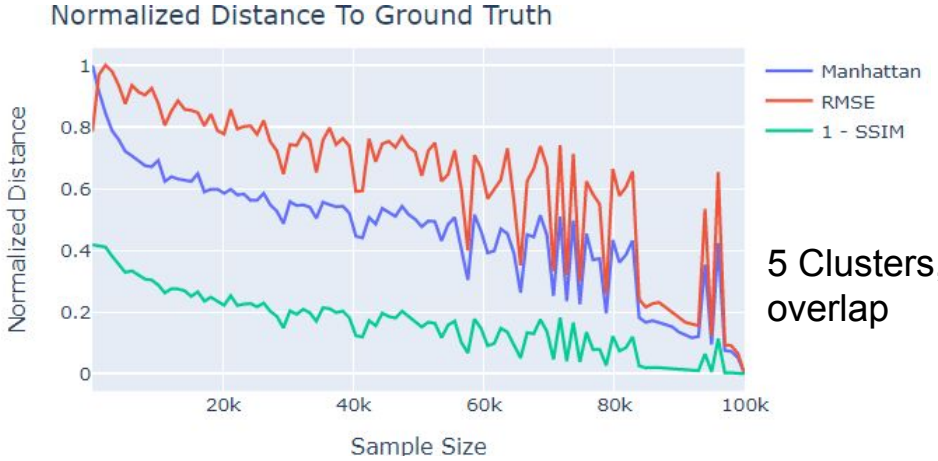


4 Clusters, no overlap

5 Clusters, no overlap



4 Clusters, overlap



5 Clusters, overlap

Results In Brief

- All the graphs have a region of high volatility especially in the latter portion of the sample size.
- Before sample rate of 40% the normalized distance values seem to decrease at a fairly steady rate.
- Afterwards, random sampling can produce wild swings in the distance to the ground truth indicating that the inclusion of exclusion of outliers or border points should not be ignored.

Conclusion And Future Work...

- Current image and data based metrics cannot produce the expected ideal curve.
  - Images produced in the lower samping regions appear invariant which suggests we need a more targeted objective function at sample rates < 10%.

- Qualitatively recommend not going above K* = 10% sampling rate with a bias toward including outliers.

- Outliers and random distribution shifts create much measured volatility without much change resulting change in the sample images.
  - Results consistent across multiple data distributions and plots.
  - Should explore if outlier based sampling reduces this affect.

- The best signal for overplotting will likely be the end user.
  - Should explore the feasibility of creating quick scatter plots from streaming data.
  - Users can interactively update up to 10% of data.

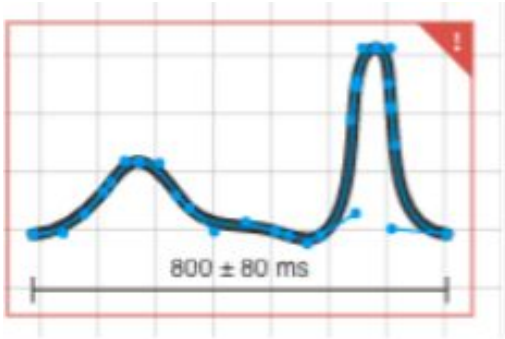
Group 3

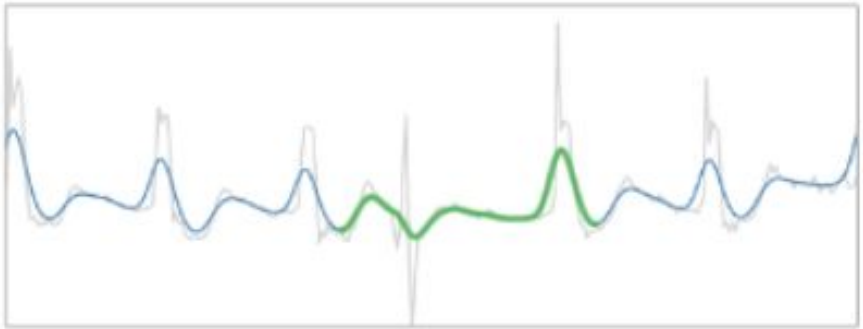
Review and Evaluation of Similarity Measures for Query-by-sketch Pattern Matching in Time Series

By Haotian Sun and Vishnu Krishnan

What is Query-by-sketch Pattern Matching in Time Series?

As the name suggests, it is using users' hand-drawn sketches to query time series data.





PROBLEM - BIT

- Query-by-sketching in time series is essential in many application scenarios but matching sketches with data often comes with non-optimal efficiency and accuracy.
- In terms of efficiency, most approaches are based on local characteristics and sliding window that is used to best match computation or similarity ranking which may lead to time consuming comparison. While there are other approaches like Qetch, which utilize shape comparisons, there is a lack of empirical evaluation on the performance between these pattern matching metrics.
- To the best of our knowledge, there is a lack of research work on the empirical evaluation of time series pattern matching techniques for the query-by-sketch applications.

RELATED WORK

- While there have been advancements in pattern matching techniques for query-by-sketching in time series applications such as Qetch, most of their focus is more towards attaining qualitative evaluations often utilizing user studies to compare to previous implementations.
- There have been papers on the comparison of time series pattern matching techniques for example, “*Using time-series similarity measures to compare animal movement trajectories in ecology*”. While these papers highlight the differences between time-series pattern matching techniques, they do not consider recent query-by-sketching algorithms such as Qetch for their comparisons. And not to mention, their focus could be application based such as in this animal movement trajectory paper.

BIT-FLIP

Our goal is to provide a system that allows the user to determine the optimal matching algorithm for their query-by-sketch use-case scenarios.

We conducted four distinct tests namely accuracy and running time tests, precision test, sensitivity to frequency components and sensitivity to noise.

We then evaluate a select few algorithms according the to these measures and provided an in-depth analysis on them

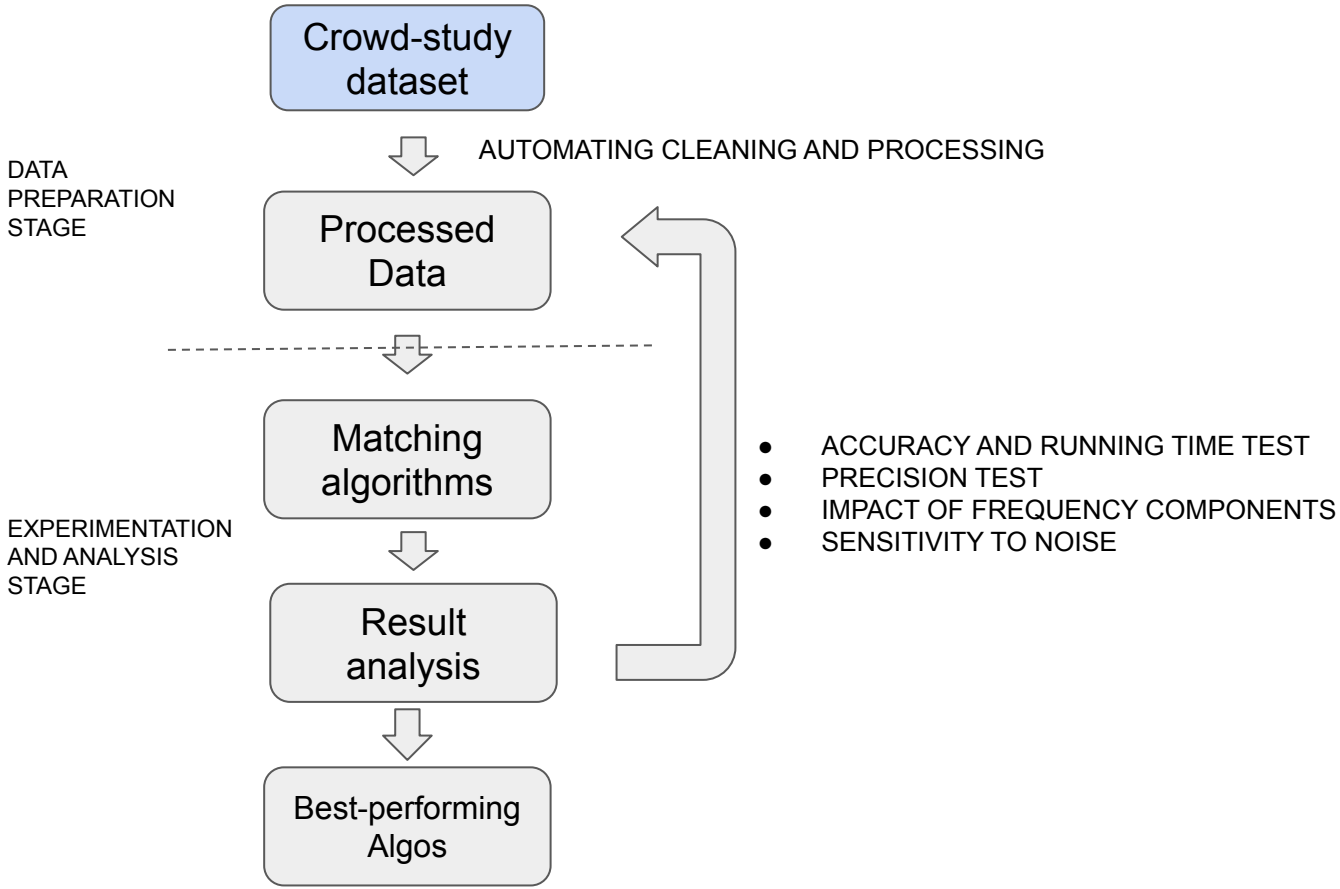
Pattern Matching Tests

- Accuracy and Running Time
- Precision
- Sensitivity to Frequency Components
- Sensitivity to Noise

Similarity Metrics

- Euclidean Distance
- Manhattan Distance
- Dynamic Time Warping(DTW) Distance
- Soft-DTW Distance
- Longest Common Subsequence(LCSS)
- Global Alignment Kernel
- Qetch Distance

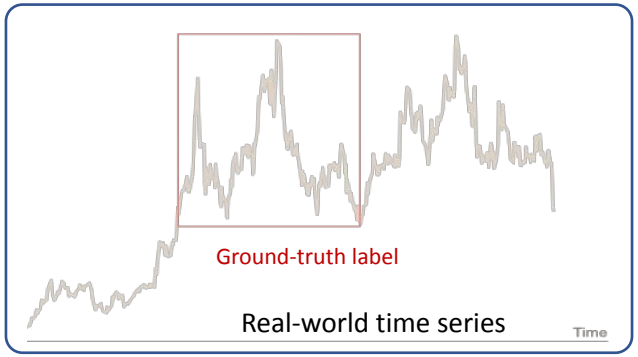
EXPERIMENT PIPELINE

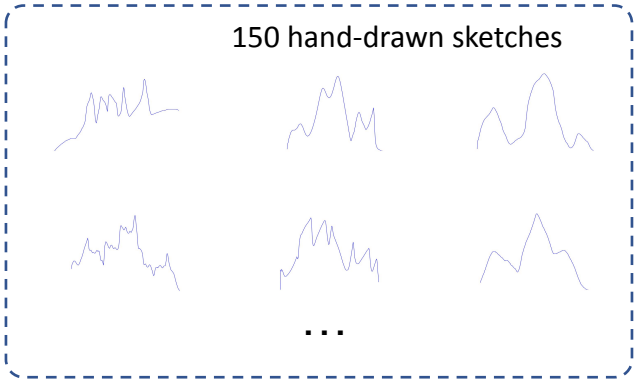


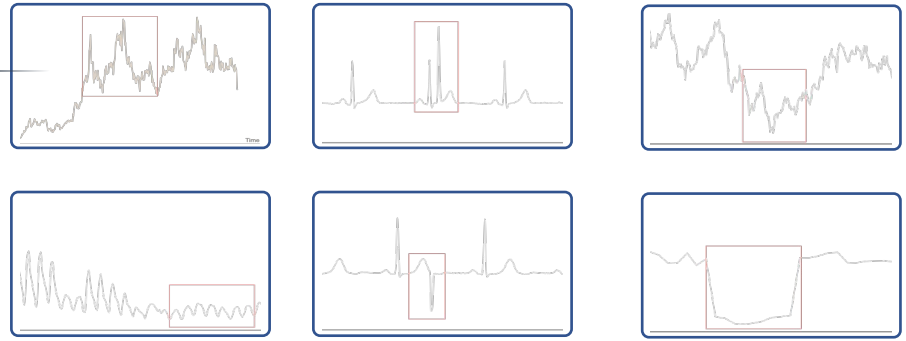
SIMILARITY MEASURES

1. Euclidean Distance(ED)
2. Manhattan Distance(MD)
3. Dynamic Time Warping(DTW) Distance
4. Soft-DTW Distance
5. Longest Common Subsequence(LCSS)
6. Global Alignment Kernel(GAK)
7. Qetch Distance

METHOD: DATA PREPARATION







Crowd-study dataset

- 8 real-world time series from various domains
- 100 labeled sketches for each time series
- Contains some duplicates, corrupted sketches
- Stored in png
  → need data preprocessing

METHOD: DATA PREPARATION

- Data Preprocessing:
  - Extracted corresponding one-dimensional data and ground-truth labels from the image files
  - Perform de-noising by setting an appropriate filter threshold

- Data Cleaning:
  - Remove the corrupted and deformed data samples
  - Detected and eliminated the duplicate sketch samples

METHOD: PATTERN MATCHING METRICS

- Accuracy and Running Time Test -

$$\bar{E} = \frac{1}{m} \sum_{i=1}^{m} \frac{|s_i - s_i^*|}{L_i} \times 100\%$$

- Precision Test -

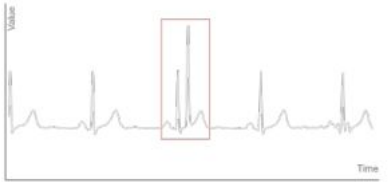
$$Precision@k(SK_i) = \frac{1}{|S_i|} \sum_{s \in SK_i} \begin{cases} 1, & rank_i(s) \leq k, \\ 0, & otherwise. \end{cases}$$

- Sensitivity to Noise Test -

$$\text{Noise} \sim N(0, \sigma)$$

METHOD: PATTERN MATCHING METRICS

- Sensitivity to Frequency Components



$$\mathcal{S}(TS) = \frac{1}{|FFT(TS)|} \sum_{f \in FFT(TS)} f^2$$

Dataset: sharp rise (sr)

Dataset: rounded bottom (rb)

Power spectrum: 7.8

Power spectrum: 224.9

- Larger power spectrum
- Fewer frequency components
- Fewer repetitive patterns
- (intuitively) less challenging for pattern matching

EVALUATION

$$\bar{E} = \frac{1}{m} \sum_{i=1}^{m} \frac{|s_i - s_i^*|}{L_i} \times 100\%$$

- Accuracy and Computation Time

Distance measure	Average matching error (%)	Running time (s)
ED	10.71	16.15
MD	10.33	<u>15.91</u>
DTW	<u>9.31</u>	35.47
Soft-DTW	15.47	136.33
LCSS	35.77	24.93
GAK	9.59	135.16
Qetch	14.21	33.97

- DTW, GAK: winner of accuracy
- ED, MD: winner of computation time
- ED, MD: reasonable choice in practice

EVALUATION

$$Precision@k(SK_i) = \frac{1}{|S_i|} \sum_{s \in SK_i} \begin{cases} 1, & rank_i(s) \leq k, \\ 0, & otherwise. \end{cases}$$

- Precision @ k

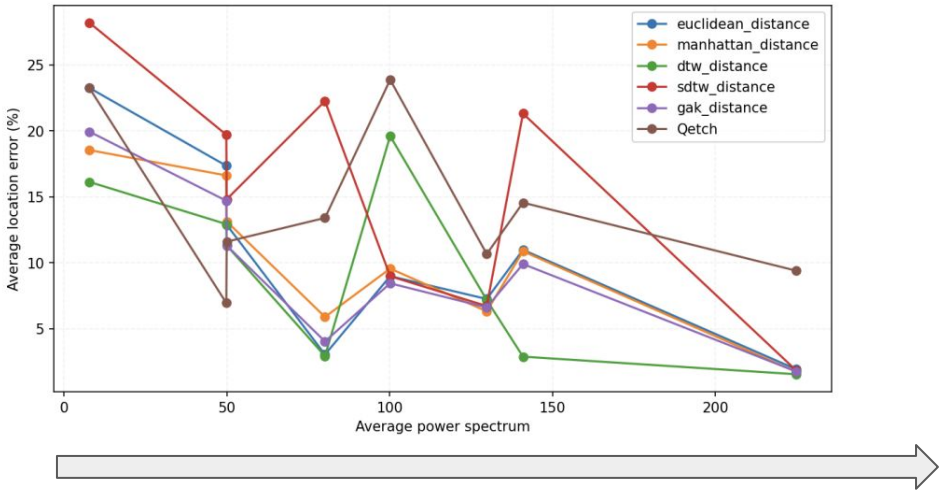
Measure	Precision@1	Precision@3	Precision@5
ED	41.375	63.125	81.250
MD	33.625	59.500	81.125
DTW	48.625	77.000	88.625
Soft-DTW	12.500	37.500	62.500
GAK	42.250	62.875	80.625
Qetch	29.625	64.375	77.000

- DTW: best precision performance
- ED, MD, GAK, Qetch: good performance when k is large

EVALUATION

$$\mathcal{S}(TS) = \frac{1}{|FFT(TS)|} \sum_{f \in FFT(TS)} f^2$$

- Impact of Frequency Components

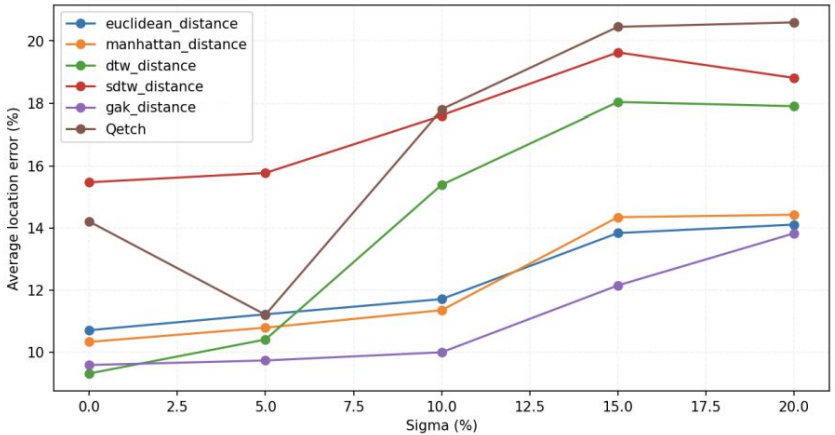


- Larger power spectrum
- Fewer frequency components
- Fewer repetitive patterns
- (intuitively) less challenging for pattern matching

- ED, MD, and GAK show excellent robustness against the varying frequency components;
- Adopt different approaches considering the power spectrum range of the target time series.

EVALUATION

- Sensitivity to Noise



Zero-mean Gaussian noise with sigma proportional to original series' std

- ED, MD, and GAK are less prone to noise
- DTW suffers from a significant degradation in matching accuracy when the error level goes up

DISCUSSION

- Qualitative observations of previous evaluation results

Similarity measure	Category	Accuracy	Precision@k	Compute Time	Frequency Sensitivity	Noise Sensitivity	Parameters
Euclidean Distance (ED)	Point-wise	good	good	best	lowest	low	sliding window step size
Manhattan Distance (MD)	Point-wise	good	good with large k	best	lowest	low	sliding window step size
Dynamic Time Warping (DTW)	Point-wise	best	best	medium	low & unstable	medium	sliding window step size
Soft-DTW	Point-wise	medium	worst	worst	medium & unstable	highest	sliding window step size& soft-smoothing factor
Longest Common Subsequence (LCSS)	Point-wise	worst	-	medium	-	-	sliding window step size
Global Alignment Kernel (GAK)	Point-wise	best	good	worst	lowest	lowest	sliding window step size& internal Gaussian bandwidth
Qetch	Shape-matching	medium	good with large k	medium	high & unstable	high	smoothing level factor

DISCUSSION

Similarity measure	Category	Accuracy	Precision@k	Compute Time	Frequency Sensitivity	Noise Sensitivity
Euclidean Distance (ED)	Point-wise	good	good	best	lowest	low
Manhattan Distance (MD)	Point-wise	good	good with large k	best	lowest	low
Dynamic Time Warping (DTW)	Point-wise	best	best	medium	low & unstable	medium
Soft-DTW	Point-wise	medium	worst	worst	medium & unstable	highest
Longest Common Subsequence (LCSS)	Point-wise	worst	-	medium	-	-
Global Alignment Kernel (GAK)	Point-wise	best	good	worst	lowest	lowest
Qetch	Shape-matching	medium	good with large k	medium	high & unstable	high

Choice of Methods in Different Scenarios

- Not recommended: Soft DTW, LCSS
-

DISCUSSION

Similarity measure	Category	Accuracy	Precision@k	Compute Time	Frequency Sensitivity	Noise Sensitivity
Euclidean Distance (ED)	Point-wise	good	good	best	lowest	low
Manhattan Distance (MD)	Point-wise	good	good with large k	best	lowest	low
Dynamic Time Warping (DTW)	Point-wise	best	best	medium	low & unstable	medium
Soft-DTW	Point-wise	medium	worst	worst	medium & unstable	highest
Longest Common Subsequence (LCSS)	Point-wise	worst	-	medium	-	-
Global Alignment Kernel (GAK)	Point-wise	best	good	worst	lowest	lowest
Qetch	Shape-matching	medium	good with large k	medium	high & unstable	high

Choice of Methods in Different Scenarios

- Not recommended: Soft DTW, LCSS
- In most scenarios without special requirement: ED and MD

DISCUSSION

Similarity measure	Category	Accuracy	Precision@k	Compute Time	Frequency Sensitivity	Noise Sensitivity
Euclidean Distance (ED)	Point-wise	good	good	best	lowest	low
Manhattan Distance (MD)	Point-wise	good	good with large k	best	lowest	low
Dynamic Time Warping (DTW)	Point-wise	best	best	medium	low & unstable	medium
Soft-DTW	Point-wise	medium	worst	worst	medium & unstable	highest
Longest Common Subsequence (LCSS)	Point-wise	worst	-	medium	-	-
Global Alignment Kernel (GAK)	Point-wise	best	good	worst	lowest	lowest
Qetch	Shape-matching	medium	good with large k	medium	high & unstable	high

Choice of Methods in Different Scenarios

- Not recommended: Soft DTW, LCSS
- In most scenarios without special requirement: ED and MD
- In noise-free scenarios with an emphasis on accuracy and precision: DTW

DISCUSSION

Similarity measure	Category	Accuracy	Precision@k	Compute Time	Frequency Sensitivity	Noise Sensitivity
Euclidean Distance (ED)	Point-wise	good	good	best	lowest	low
Manhattan Distance (MD)	Point-wise	good	good with large k	best	lowest	low
Dynamic Time Warping (DTW)	Point-wise	best	best	medium	low & unstable	medium
Soft-DTW	Point-wise	medium	worst	worst	medium & unstable	highest
Longest Common Subsequence (LCSS)	Point-wise	worst	-	medium	-	-
Global Alignment Kernel (GAK)	Point-wise	best	good	worst	lowest	lowest
Qetch	Shape-matching	medium	good with large k	medium	high & unstable	high

Choice of Methods in Different Scenarios

- Not recommended: Soft DTW, LCSS
- In most scenarios without special requirement: ED and MD
- In noise-free scenarios with an emphasis on accuracy and precision: DTW
- In scenarios with an emphasis on accuracy and precision, but in the presence of noise and frequency disturbance, one can trade the running time for accuracy by using the GAK-based approach;

DISCUSSION

Similarity measure	Category	Accuracy	Precision@k	Compute Time	Frequency Sensitivity	Noise Sensitivity
Euclidean Distance (ED)	Point-wise	good	good	best	lowest	low
Manhattan Distance (MD)	Point-wise	good	good with large k	best	lowest	low
Dynamic Time Warping (DTW)	Point-wise	best	best	medium	low & unstable	medium
Soft-DTW	Point-wise	medium	worst	worst	medium & unstable	highest
Longest Common Subsequence (LCSS)	Point-wise	worst	-	medium	-	-
Global Alignment Kernel (GAK)	Point-wise	best	good	worst	lowest	lowest
Qetch	Shape-matching	medium	good with large k	medium	high & unstable	high

Choice of Methods in Different Scenarios

- Not recommended: Soft DTW, LCSS
- In most scenarios without special requirement: ED and MD
- In noise-free scenarios with an emphasis on accuracy and precision: DTW
- In scenarios with an emphasis on accuracy and precision, but in the presence of noise and frequency disturbance, one can trade the running time for accuracy by using the GAK-based approach;
- In the scenarios where the pointwise sliding window is computationally costly: Qetch

CONCLUSION

- We proposed a comprehensive review and evaluation of various similarity measures for query-by-sketch pattern matching in time series;

- We then designed concrete experiments on seven widely-used similarity measures and evaluated them with five performance metrics;

- Based on the experimental results, we further summarized five different real-world scenarios and their corresponding best choices of methods.

Group 1

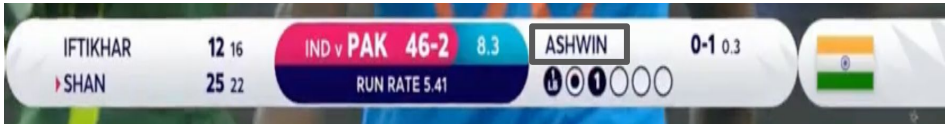
Interactive Compositional Querying of Video Data

Ashmita Raju
Gaurav T Kakkar
Myna Prasanna Kalluraya

Support Exploratory Video Analytics using domain hints

Query: Find where player (Ashwin) is bowling in a cricket game

- Approach 1: Run face recognition
- Approach 2: Look for Ashwin's jersey number
- **Alternate Approach : Use scoreboard and apply OCR – domain hint**



Prior Works - The missing piece

- Optimize simple selection/aggregate queries

```
SELECT id, Facedetection(data) FROM   CRICKETGAMES;
```

Computation Expensive

- Optimize complex action queries

```
SELECT First(id), Last(id), Segment(data, 16f) AS seg
FROM   JACKSONHOLE
HAVING Actionrecognizer(seg) = "pedestriancrossing";
```

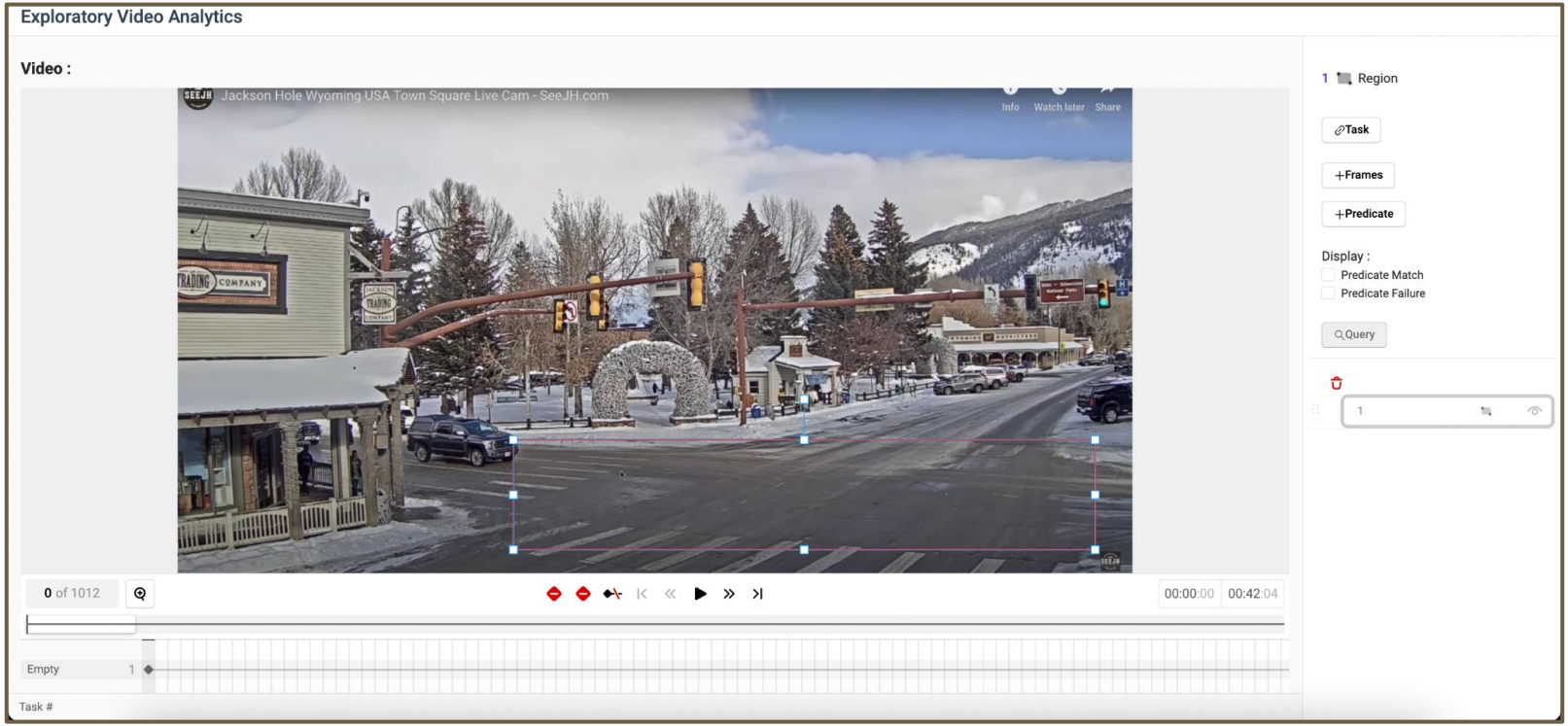
Requires training custom models

System Architecture - Pivot

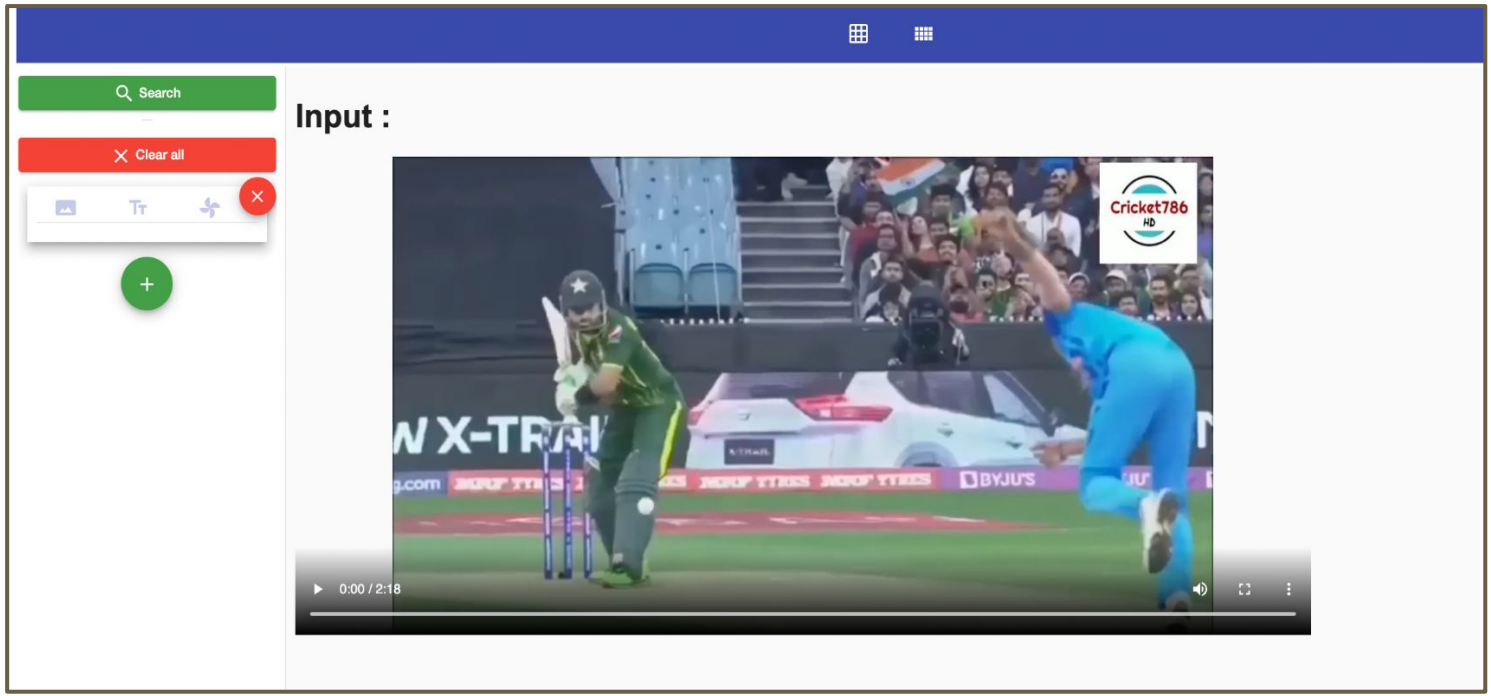
Custom UI*

* vitrivr Multimodal Multimedia Retrieval with vitrivr

Initial UI (based on Label Studio)

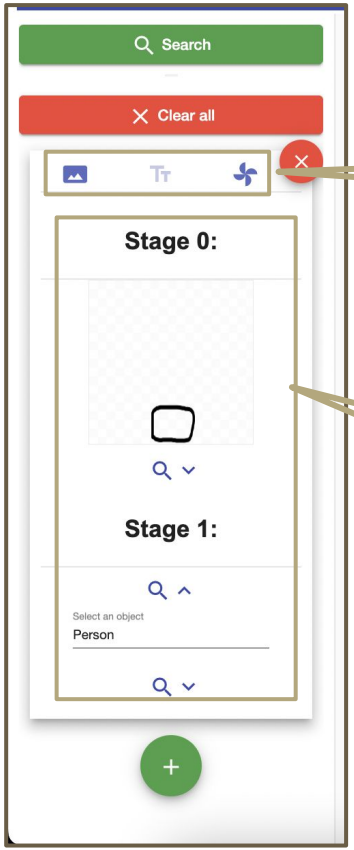


UI to support compositional queries with domain hints



* vitrivr Multimodal Multimedia Retrieval with vitrivr

UI to support compositional queries with domain hints

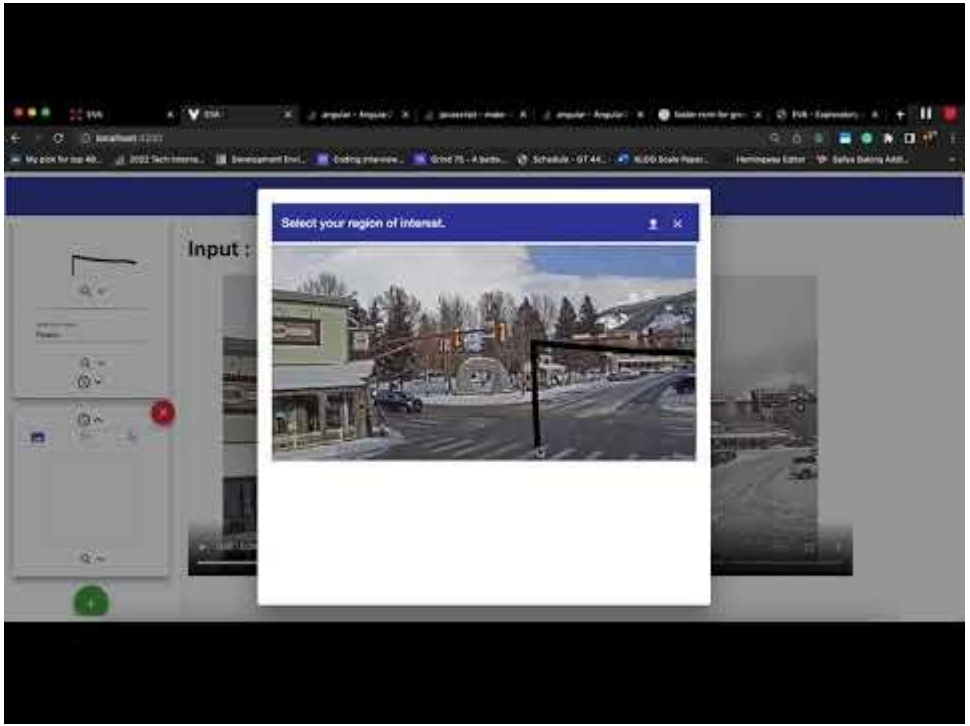


Query Type: OCR, Object Detection

Supports the concept of **staging**. (Similar to predicate re-ordering)

Demo - Sports

Demo - Traffic



Qualitative Results

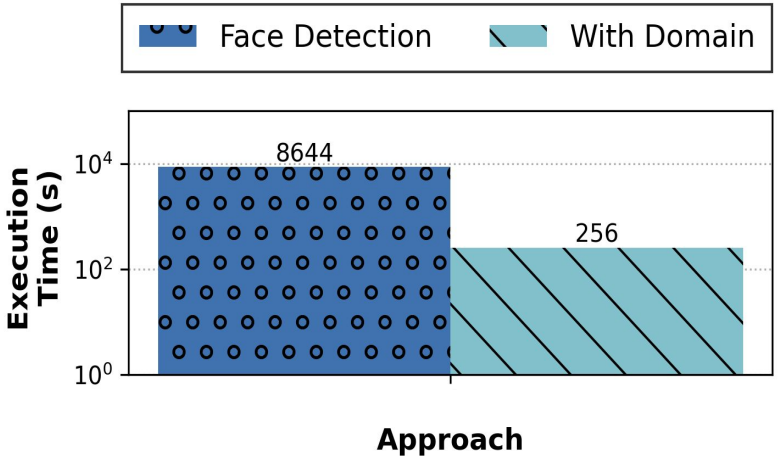
Positives

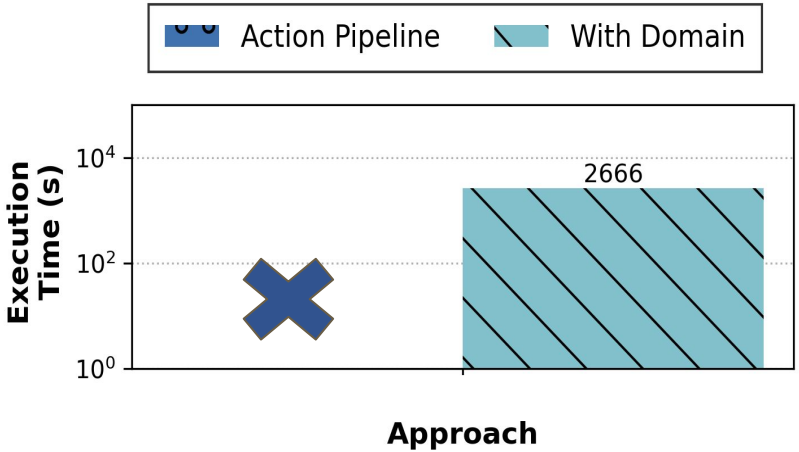
- Simple UI with straight-forward steps.
- Powerful tool to search for objects in region of interest.
- Supports OCR and object detection.

Opportunities

- Add tooltips to help the user learn about the components.
- No support for filtering dataset based on metadata like VIDEOS, IMAGES etc.
- Concept of staging is unintuitive.

Quantitative Results





Challenges Faced

EVADB

- Did not have support for querying dataset
  - Each video was one table (bad from UI perspective)
  - Added support for loading using regex - *"games/*.mp4"*
- Added support for user provided region of interest
- Does not support trajectory based search (future work)

Label Studio

- Fundamental difference from our requirements
  - Focused on labelling one video instead of dataset
  - Showing results to user was a nightmare
- Large code base with no documentation

Future Steps

- Support trajectory based search.
- Support more tasks such as Face Recognition.
- Support asynchronous visualization of results.

Thank You

Food for thought: Why is our class named MDS?

SQL Q-Suggest: Context-Aware SQL Prediction and Auto-Completion with Q-Learning

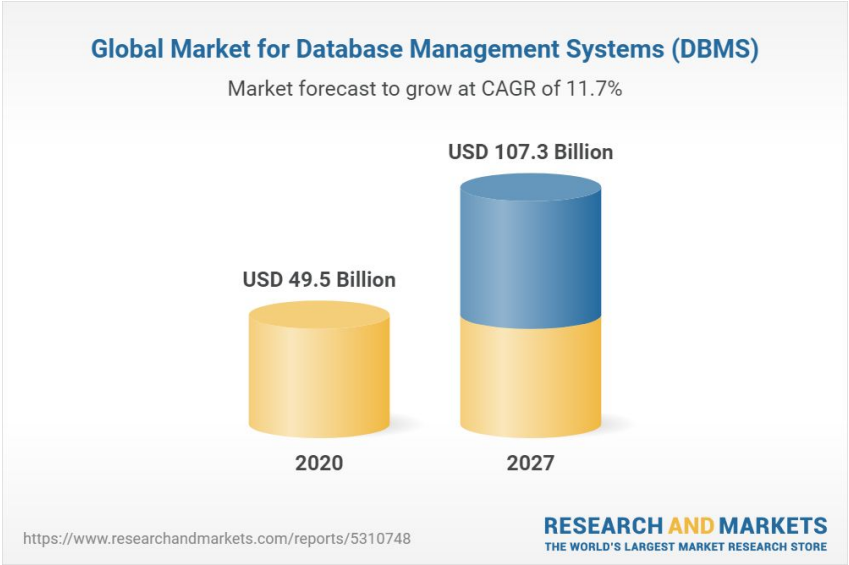
Group 6: Cangdi Li, Ting Yu, Yiheng Mao

Why SQL Autocompletion?

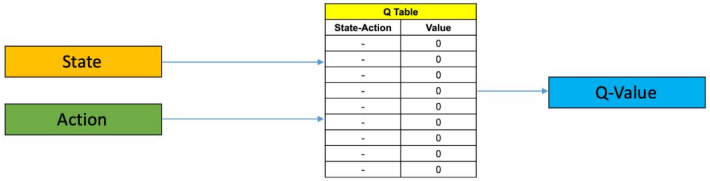
With **ever increasing** amount of data stored in scalable database management systems, analytical SQLs are becoming more frequently adopted to access and modify DBMSs.

Writing SQL queries is **not easy**. Even for veteran analysts, it can still be cumbersome to work with a new database with unfamiliar schemas or a database with overly complicated relations.

An query autocompletion and prediction system **could help alleviate** some of the difficulties of query writing.



Bit: Q-Learning and SnipSuggest



Q Learning

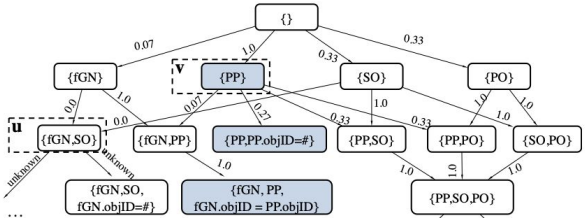
Prior Work in Query Prediction:

1. Q-Learning

A recent evaluation paper demonstrates Q-Learning to be one of **the most efficient** algorithms in query predicting. It is slightly less effective than RNN but runs significantly faster.

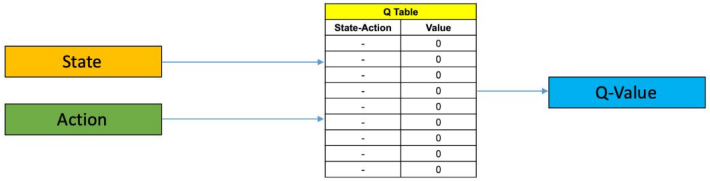
2. SnipSuggest

SnipSuggest is a **popular** partial query prediction system that utilizes a workload directed acyclic graph (DAG) and a bayesian-based ranking algorithm. It is effective at predicting query fragments (snippets).



Workload DAG from SnipSuggest

Bit: Q-Learning and SnipSuggest



Q Learning

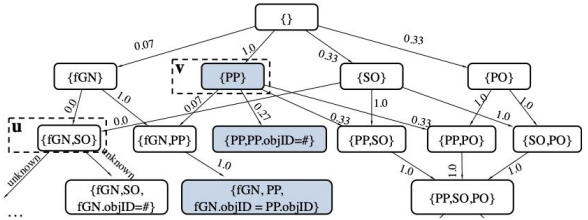
However,

1. **Q-Learning**

   Can only generate predictions for next **full** queries and is less effective when some prefix is given.

2. **SnipSuggest**

   Does not capture the sequential nature (context) of query logs that Q-Learning does.

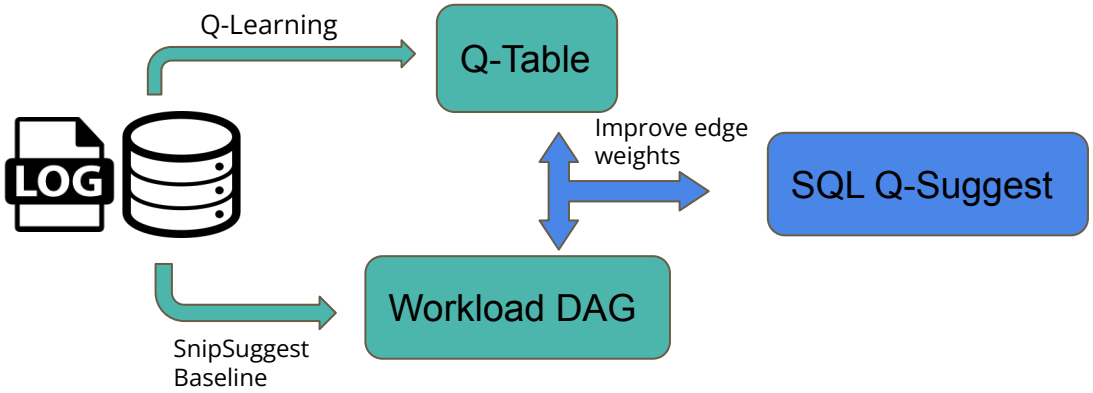


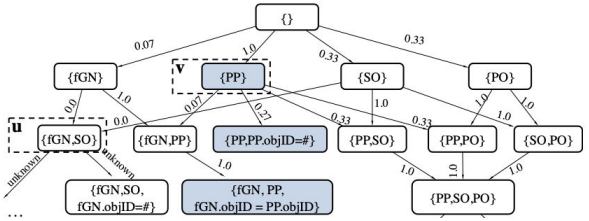
Workload DAG from SnipSuggest

SQL Q-Suggest (SQLQS) Auto-completion and Prediction System

After training the Q-Learning model, we will derive a Q-table. We then update the workload DAG from baseline SnipSuggest with knowledge from the Q-table: we **update DAG edge weights** with corresponding **Q-values**.

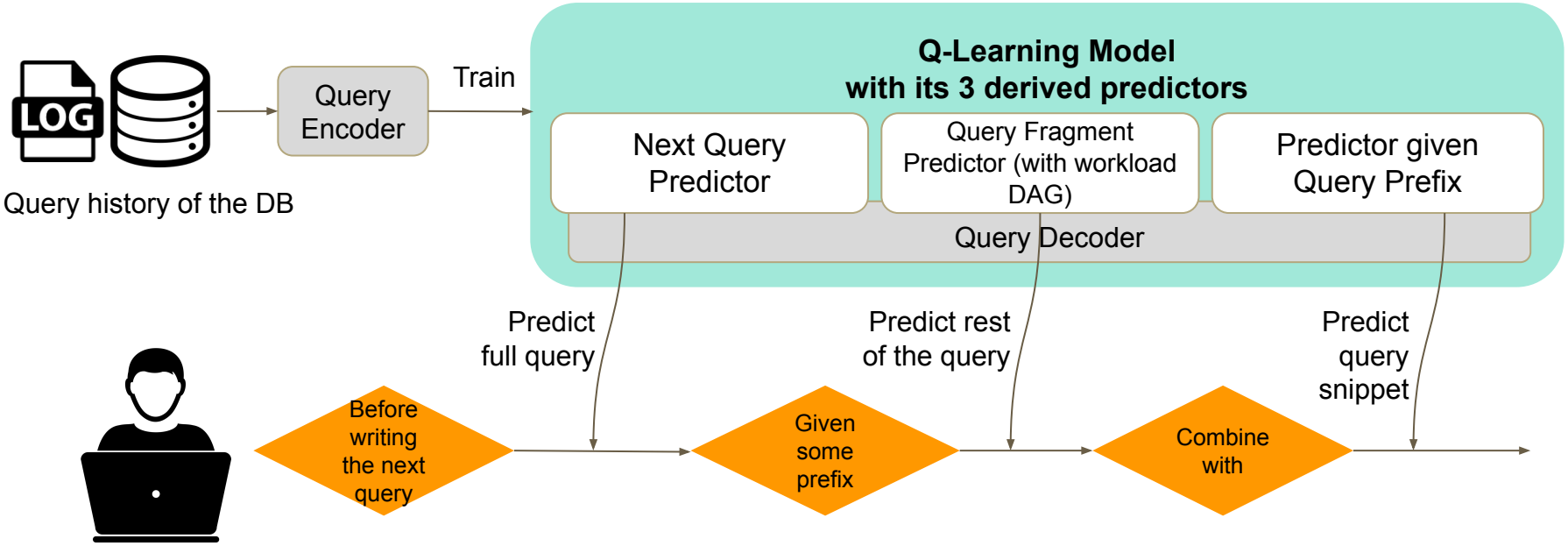
System Scope: Single DB, multiple user, multiple sessions with past query logs. The system is able to predict full queries as well as fragments while capturing the sequential nature of past queries.





Workload DAG from SnipSuggest

System Architecture



Predictions available throughout the query input cycle

How do we determine the similarity of two queries?

- Similarity Function Definition:
  - Cosine $$cosine - similarity\,(A, B) = \frac{A \cdot B}{|A||B|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$
  - Jaccard $$jaccard - similarity\,(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

Evaluation Plan

1. Cosine Similarity VS Jaccard Similarity

2. Prediction with Query Prefix VS without

3. SQL Q-Suggest VS SnipSuggest

Evaluation Dataset Setup

We adapt **IDEBench**, a benchmark tool for interactive data exploration, to generate **100 human–like database sessions, each containing 100 SQL queries** in a meaningful sequence (modeled by Markov chains).

To simulate our use case, where our system is trained on past queries on a database and used to give predictions for a future session, we **train the model on 90 sessions and test on the other 10 sessions**. We use the 10 fold cross validation idea to repeat the testing 10 times on each 9:1 session split.

Evaluation Scope: SELECT attributes FROM tables WHERE conditions GROUP BY attributes

Parsing the query as a dictionary.

{"SELECT": [attr1, ...], "FROM": [table1], "WHERE": [condition1,...], "GROUP BY": [attr1,...]}

Query Encoder/Decoder

Evaluation 1: Cosine Similarity VS Jaccard Similarity

	Cosine	Jaccard
Mean Precision	**0.370**	**0.383**
std(Precision)	0.202	0.187
Mean Recall	**0.280**	**0.279**
std(Recall)	0.224	0.210
Mean F1 Score	**0.287**	**0.291**
std(F1)	0.202	0.187
Mean Predict Time	65 sec	58 sec
Mean Training Time	0.04 sec	0.04 sec
Mean Model Size	1.1GB	650MB

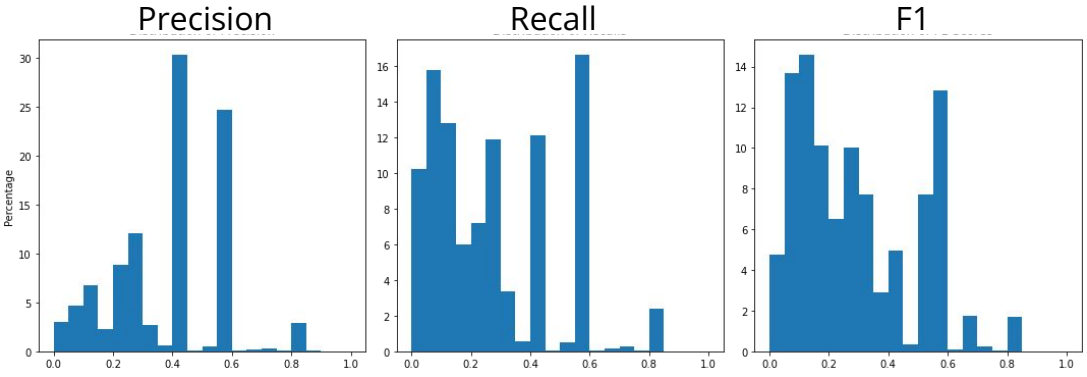


Fig. Query level precision, Recall, F1 distribution

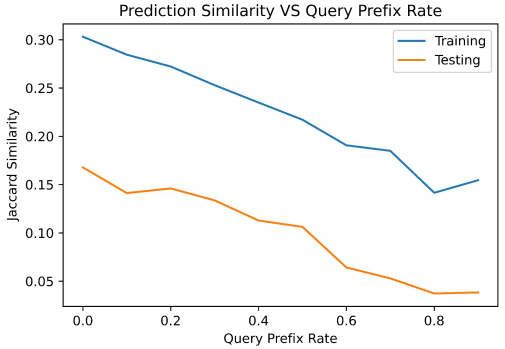
Each model is trained on 90 sessions (90 * 100 queries), evaluated on 10 sessions (10 * 99 queries)

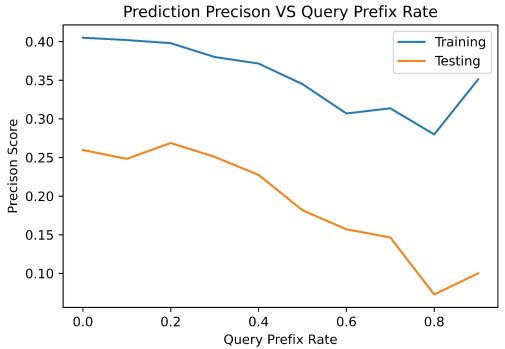
Not much difference in performance. Query level performance distributions are also highly similar.

Jaccard model size is proportional to #past queries used. Cosine model size is proportional to #query snippets in past queries, potentially huge for very complex database.

Evaluation 2: Prediction with Query Prefix VS without

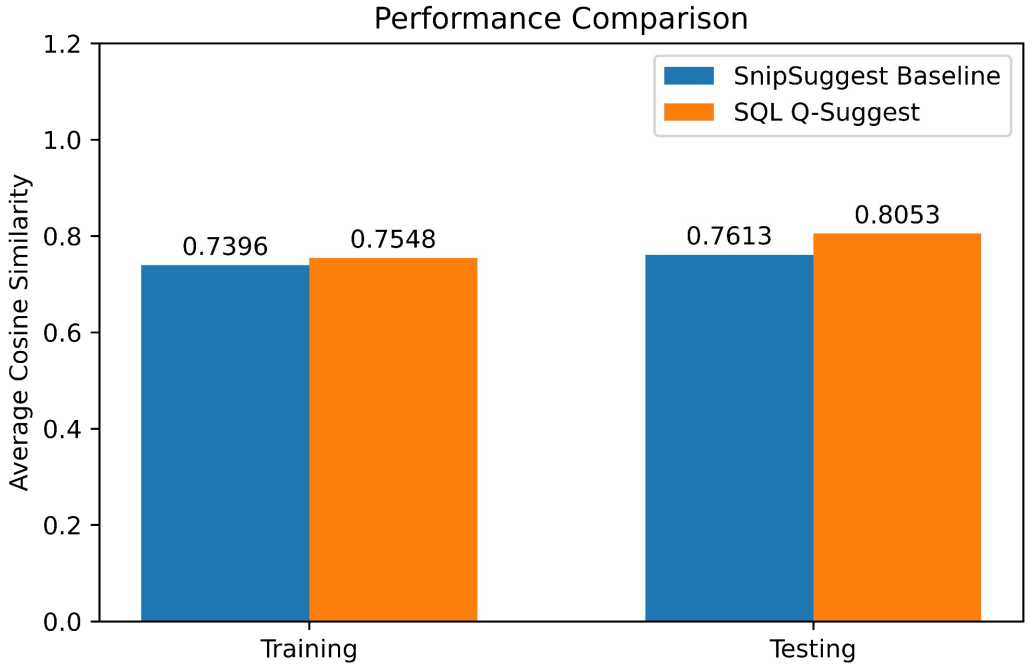
- In order to see how useful **SQL Q-suggest (SQLQS)** is for user, we evaluate the prediction performance **with user partial input query prefix** VS **without**.
- Using **Jaccard Similarity**.
- **Query Prefix Rate:** We use a percentage to split the query into **prefix** and **rest,** with the prefix feeding into the Q-Learning table when selecting the similar query, and evaluate on the prediction of the rest query.
- **2 Evaluation Metrics:**
- (1). Jaccard Similarity
- (2). Jaccard Precision
-
- **Observation**:
    - Negative relation: less data to compare
    - No prefix rate to 0.2 prefix rate is of good performance
    - 0.9 prefix rate is outlier (almost no data to compare)





Evaluation 3: SQL Q-Suggest vs SnipSuggest Baseline

With 10 instances each using 9 100-query sessions as training data and 1 session as testing, we observed roughly **2%** improvement in average training similarity and **4%** in average testing similarity over SnipSuggest Baseline model.



Conclusion

- In general, we present SQL Q-Suggest (SQLQS) which provides SQL recommendation and auto-completion for multiple users that query from the same database(s).
- Compared with prior work in Q-learning, SQLQS uses Jaccard similarity, a more lightweight, scalable score with the same performance.
- Given 0% to 20% user input prefix in SQLQS can achieve its best performance.
- SQLQS performs slightly better than the SnipSuggest baseline model.

Future Work

- Improving the current model performance with better parameter tuning.
- Support more SQL Query Clause (Insert, update, create...) across multiple databases.

Thank you