# Boosting Dialog Response Generation

**Wenchao Du**
Language Technologies Institute
Carnegie Mellon University
wenchaod@cs.cmu.edu

**Alan W Black**
Language Technologies Institute
Carnegie Mellon University
awb@cs.cmu.edu

# Boosting Dialog Response Generation

- Introduction
  - Seq2Seq suffers from “dull response” problem
  - reinforcement learning, GAN, penalizing dull responses
  - Boosting (1997) and assembling,  having been studied in image generation (also suffered missing model problem) and machine translation (2017)
  - Boosting:  Iteratively train multiple models; reweight according to previous error; combine models

# Preliminaries

**Standard Seq2Seq, MLE**

$$\log p(y \mid x) = \sum_{i=1}^{n} \log p(y_i \mid y_1 \ldots y_{i-1}, x) \quad (1)$$

**Decoding objective base on mutual information of x and y**

$$MMI(x, y) = \log p(y \mid x) - \lambda p(y) \quad (2)$$

**Reward-Augmented Maximum Likelihood (RAML)**

$$s(y \mid y^*; \tau) = \frac{1}{Z(y^*, \tau)} \exp\{r(y, y^*)/\tau\} \quad (3)$$

Objective : Minimize the KL Divergence

$$\sum_{x,y^*} D_{KL}(s(y \mid y^*) \mid\mid p(y \mid x)) = - \sum_{x,y^*} \sum_{y} s(y \mid y^*) \log p(y \mid x) + const \quad (4)$$

# Preliminaries

**Density estimate of each iteration in boosting**

$$q_T = h_T^{\alpha_T} q_{T-1} = \frac{\prod_{t=1}^{T} h_t^{\alpha_t}}{Z_T} \quad (5)$$

if at each iteration, the model can optimize the distribution of this form:

$$d_t \propto \left(\frac{p}{q_t}\right)^{\beta_t} \quad (6)$$

1. Assume the sources are uniformly distributed (均匀分布)
2. This paper **extend** this assumption to the exponential payoff distribution (3)

Then, KL divergence decrease:

$$D_{KL}(P \mid\mid Q_t) \leq D_{KL}(P \mid\mid Q_{t-1}) \quad (7)$$

# Design

$$d_t(x,y) \propto \left(\frac{p(x,y)}{q_t(x,y)}\right)^{\beta_t} \frac{D_t(y)}{1-D_t(y)} \qquad (9)$$

$) = 1$

(8)

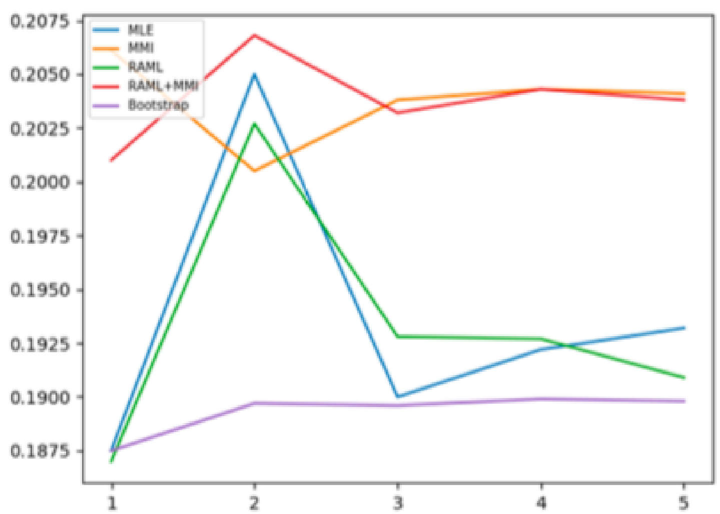
- (6) 式中可知，数据权值与response perplexity成反比; Generic response sometimes has high perplexity; Frequently generated.
- Adopt a rule-based discriminator
  - GAN-like approach is not suitable, as negative samples are too few.
  - Maintain a list of **most frequently generated responses**.
  - Sim(x, y)： n-gram overlap with n ≥ 4
  - Our discriminator: , c = 0
  - Data weight at round t:

# Design

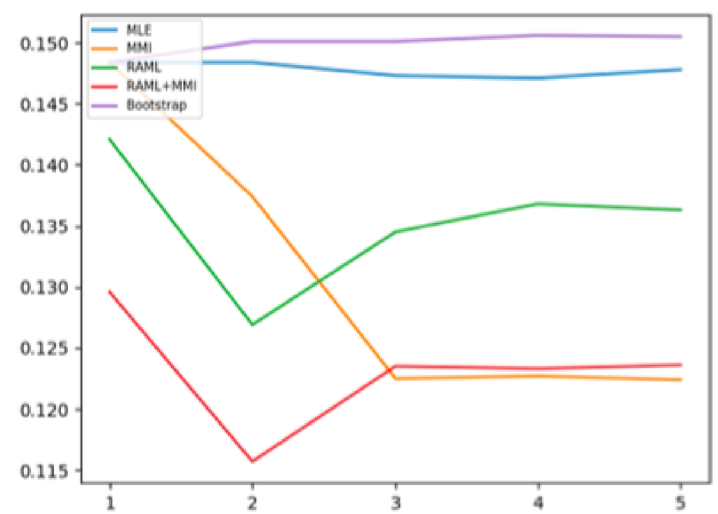
- Model Combination
  - Heuristics
  - Generate candidate responses from single best model using **beam search**.
  - Score the candidate by all models.
  - The one with highest average score is chosen.

# Experiments and Evaluation

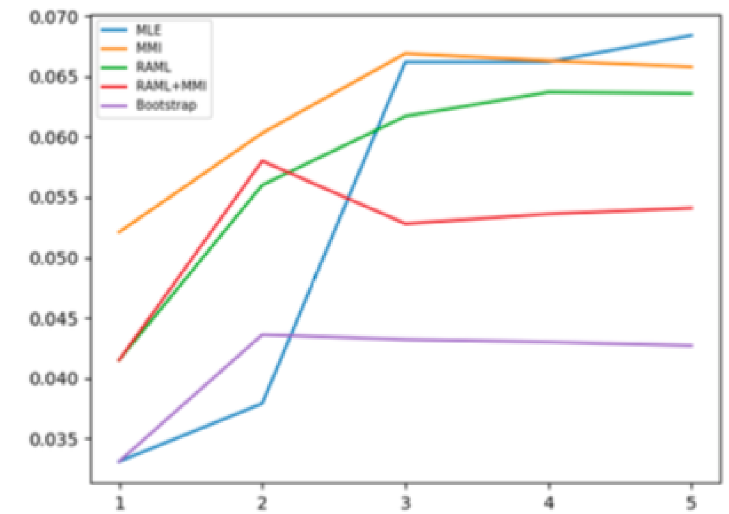
- Dataset: Persona Dataset (2018)



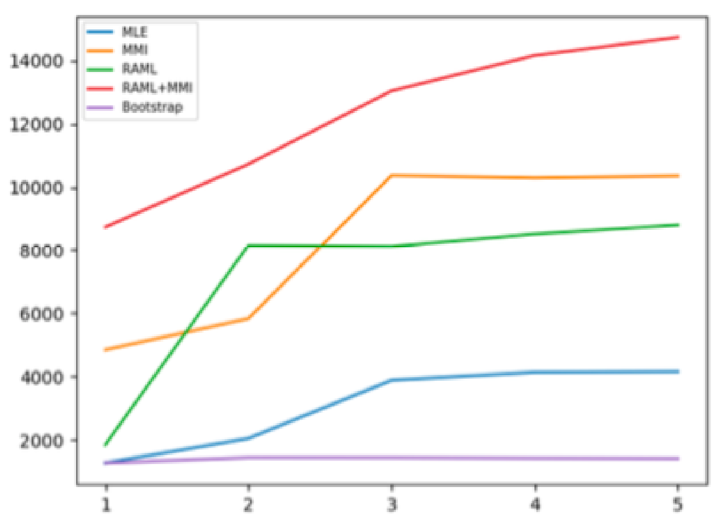

(a) BLEU



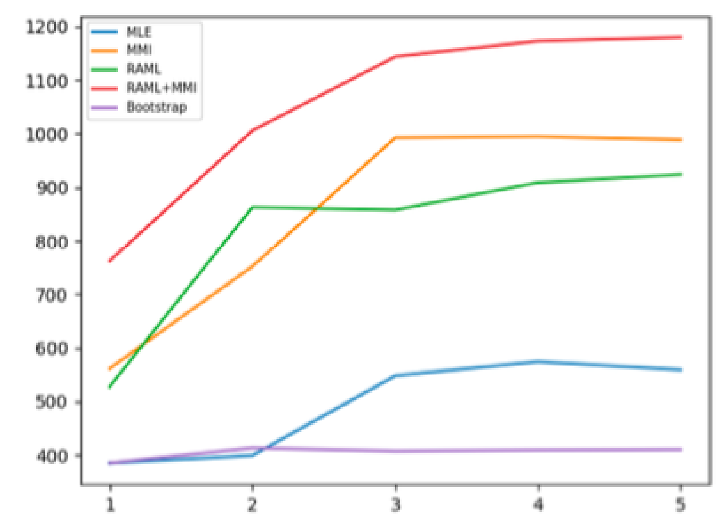

(b) ROUGE-L



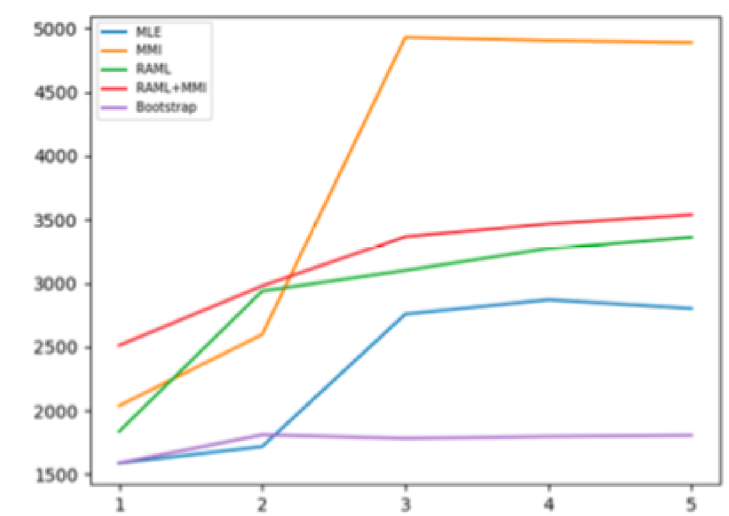

(c) Cosine Similarity




(d) Inertia




(e) Number of unigrams




(f) Number of trigrams

Figure 1: Quantitative results. X-axis is for iteration and y-axis for metrics. The numbers at iteration 1 represent the base models.

| Model | Win | Loss | Tie |
|---|---|---|---|
| MLE | $37.6 \pm 6.4\%$ | $17.6 \pm 4.0\%$ | $44.8 \pm 6.4\%$ |
| MMI | $36.0 \pm 9.2\%$ | $16.8 \pm 6.8\%$ | $47.2 \pm 8.8\%$ |
| RAML | $44.8\% \pm 10.8\%$ | $16.8 \pm 4.8\%$ | $38.4 \pm 12.4\%$ |

Table 1: Human evaluation results. "Win" stands for the boosted model winning.

| | |
|---|---|
| **Context** | my family lives in alaska . it is freezing down there . |
| **Human** | i bet it is oh i could not |
| **Baseline** | what do you do for a living |
| **Boosted** | do you live near the beach ? i live in canada |

Table 2: Examples of generated responses from baseline sequence-to-sequence model and its boosted counterpart.

# Result Analysis

- Boosting drastically improves performance, far better than bootstrapping. BLEU fluctuated in a tight range, while ROUGE-L suffered from boosting a little.
- Diversity of the response is **significantly improved**.
- Qualitative evaluation shows boosting models beat base models.

# Conclusion

- We novelly combine boosting and RAML for response generation.
- Combining boosting with MMI gives some of the most diversified results.
- This method can improve diversity without harming the quality of the responses, and the quality may be better than base models.

# Learning Compressed Sentence Representations for On-Device Text Processing

**Dinghan Shen[1*], Pengyu Cheng[1*], Dhanasekar Sundararaman[1]**
**Xinyuan Zhang[1], Qian Yang[1], Meng Tang[3], Asli Celikyilmaz[2], Lawrence Carin[1]**
[1] Duke University [2] Microsoft Research [3] Stanford University
dinghan.shen@duke.edu

# Introduction

- Sentence embeddings require large storages or memory footprint. It is computationally expensive to retrieve semantically-similar sentences.

- In mobile devices, only a relatively tiny memory footprint and low computational capacity are typically available.

- This paper’s method: **binarizing** the continuous sentence embeddings, using three alternative strategies. **2%** performance drop while reduces **98%** storage requirement.

- This paper found that the Hamming distance between the binary code can measure the relatedness between a sentence pair even **better** than cosine similarity between continuous embeddings.

# Binarized embeddings

- Memory efficient, relative to discrete embeddings.
- Fast retrieval based on a Hamming distance calculation.
- Previous work focuses on word-level, this paper works on extracting binarized embeddings at the **sentence-level**.

# Proposed Approach

- **f(x)**, continuous embeddings extracted by the encoder.
- Target: find a transformation **g**, which can convert f(x) to highly informative **binary** sentence representations. Four strategies:
  - Hard Threshold
  - Random Projection
  - Principal Component Analysis
  - Autoencoder Architecture

# Hard Threshold

- Suppose that h, b denote continuous and binary sentence embeddings respectively.
- Suppose s is the hard threshold, L is the dimension of h, for i = 1, 2, …, L:

$$b^{(i)} = \mathbf{1}_{h^{(i)} > s} = \frac{\operatorname{sign}(h^{(i)} - s) + 1}{2}, \qquad (1)$$

- Each dimension is converted to 0 or 1
- Lost information.

# Random Projection

- Simple applying a random projection over pre-trained continuous representation.

- Initialize the value of the resulting binary representations matrix Wi,j **uniformly** (D denotes the dimension of W):

$$W_{i,j} \sim \text{Uniform}(-\frac{1}{\sqrt{D}}, \frac{1}{\sqrt{D}}), \qquad (2)$$

- Then apply hard threshold operation to binarize it into compact form. D can be set arbitrarily.

# Principal Component Analysis (PCA)

- PCA can reduce the dimensionality of pre-trained embeddings.
- A set of sentences $\{x_i\}_{i=1}^N$, and their corresponding continuous embeddings $\{h_i\}_{i=1}^N \subset \mathbb{R}^L$, learn a projection to reduce dimensions.
- After centralizing $h_i = h_i - \frac{1}{N}\sum_{i=1}^N h_i$, the matrix H, has a singular value decomposition (SVD), $H = U\Lambda V^T$. The first D rows can be used as projection of W. Then apply the hard threshold operation.

# Autoencoder architecture

Extract useful features

$$b^{(i)} = \mathbf{1}_{\sigma(W_i \cdot h + k^{(i)}) > s^{(i)}} = \frac{\operatorname{sign}(\sigma(W_i \cdot h + k^{(i)}) - s^{(i)}) + 1}{2}, \quad (3)$$

Reconstruct the original continuous embedding with linear transformation.

$$\hat{h}^{(i)} = W_i' \cdot b + k'^{(i)}, \quad (4)$$

**Reconstruction loss**:
The mean square error of h

$$\mathcal{L}_{rec} = \frac{1}{D} \sum_{i=1}^{D} (h^{(i)} - \hat{h}^{(i)})^2, \quad (5)$$

This objective **imposes** the binary vector b to encode more information from h,

# Autoencoder architecture

- **Target**: To preserve **similarity information** of the original embeddings and improve the binary embedding's **semantic-preserving** property.

- Define a semantic-preserving regularizer as:

$$\mathcal{L}_{sp} = \sum_{\alpha,\beta,\gamma} \max\{0, l_{\alpha,\beta,\gamma}[d_h(b_\alpha, b_\beta) - d_h(b_\beta, b_\gamma)]\}, \tag{6}$$

$l_{\alpha,\beta,\gamma} = 1$ if $d_c(h_\alpha, h_\beta) \geq d_c(h_\beta, h_\gamma)$ -1 otherwise

- Entire objective function $\mathcal{L} = \mathcal{L}_{rec} + \lambda_{sp}\mathcal{L}_{sp}$

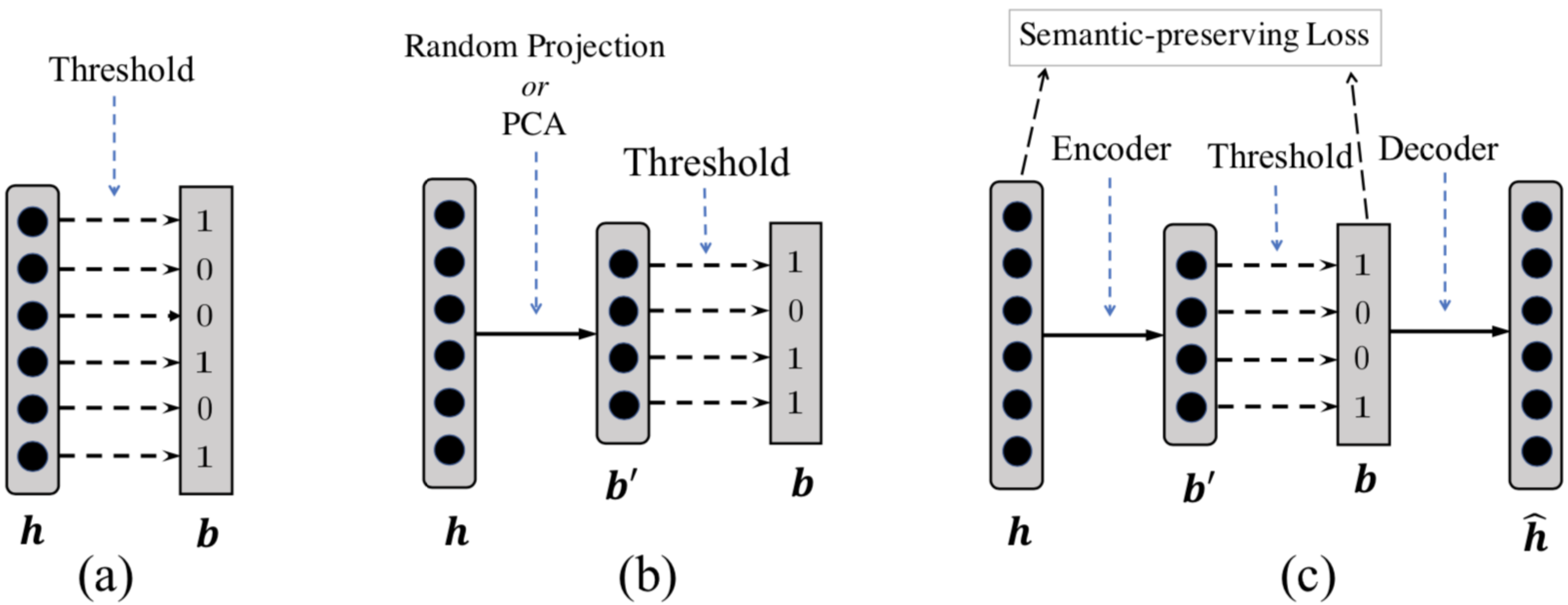



Figure 1: Proposed model architectures: (a) direct binarization with a hard threshold $s$; (b) reducing the dimensionality with either a random projection or PCA, followed by a binarization step; (c) an encoding-decoding framework with an additional semantic-preserving loss.

# Experiment

- Using **InferSent** as the continuous embeddings
- Sentence encoder: bidirectional LSTM architecture with max-pooling operation over hidden units.
- Train on Stanford Natural Language Inference (SNLI) and MultiNLI datasets.

# Result

| Model | Dim | MR | CR | SUBJ | MPQA | SST | STS14 | STSB | SICK-R | MRPC |
|---|---|---|---|---|---|---|---|---|---|---|
| ***Continuous (dense)** sentence embeddings* | | | | | | | | | | |
| fastText-BoV | 300 | 78.2 | 80.2 | 91.8 | 88.0 | 82.3 | .65/.63 | 58.1/59.0 | 0.698 | 67.9/74.3 |
| SkipThought | 4800 | 76.5 | 80.1 | 93.6 | 87.1 | 82.0 | .29/.35 | 41.0/41.7 | 0.595 | 57.9/66.6 |
| SkipThought-LN | 4800 | 79.4 | 83.1 | **93.7** | 89.3 | 82.9 | .44/.45 | - | - | - |
| InferSent-FF | 4096 | 79.7 | 84.2 | 92.7 | 89.4 | 84.3 | .68/.66 | 55.6/56.2 | 0.612 | **67.9/73.8** |
| InferSent-G | 4096 | **81.1** | **86.3** | 92.4 | **90.2** | **84.6** | **.68/.65** | **70.0/68.0** | **0.719** | 67.4/73.2 |
| ***Binary (compact)** sentence embeddings* | | | | | | | | | | |
| InferLite-short | 256 | 73.7 | 81.2 | 83.2 | 86.2 | 78.4 | 0.61/- | 63.4/63.3 | 0.597 | 61.7/70.1 |
| InferLite-medium | 1024 | 76.3 | 83.2 | 87.8 | 88.4 | 81.3 | 0.67/- | 64.9/64.9 | 0.642 | 64.1/72.0 |
| InferLite-long | 4096 | 77.7 | 83.7 | 89.6 | 89.1 | 82.3 | 0.68/- | 67.9/67.6 | 0.663 | 65.4/**72.9** |
| HT-binary | 4096 | 76.6 | 79.9 | **91.0** | 88.4 | 80.6 | .62/.60 | 55.8/53.6 | 0.652 | 65.6/70.4 |
| Rand-binary | 2048 | 78.7 | 82.7 | 90.4 | 88.9 | 81.3 | .66/.63 | 65.1/62.3 | 0.704 | 65.7/70.8 |
| PCA-binary | 2048 | 78.4 | 84.5 | 90.7 | 89.4 | 81.0 | .66/.65 | 63.7/62.8 | 0.518 | 65.0/ 69.7 |
| AE-binary | 2048 | 78.7 | **84.9** | 90.6 | 89.6 | 82.1 | .68/.66 | 71.7/69.7 | 0.673 | 65.8/70.8 |
| AE-binary-SP | 2048 | **79.1** | 84.6 | 90.8 | **90.0** | **82.7** | **.69/.67** | **73.2/70.6** | **0.705** | **67.2**/72.0 |

Table 1: Performance on the test set for 10 downstream tasks. The STS14, STSB and MRPC are evaluated with Pearson and Spearman correlations, and SICK-R is measured with Pearson correlation. All other datasets are evaluated with test accuracy. InferSent-G uses Glove (G) as the word embeddings, while InferSent-FF employs FastText (F) embeddings with Fixed (F) padding. The empirical results of InferLite with different lengths of binary embeddings, *i.e.*, 256, 1024 and 4096, are considered.

# Nearest Neighbor Retrieval

| Hamming Distance (binary embeddings) | Cosine Similarity (continuous embeddings) |
|---|---|
| **Query: Several people are sitting in a movie theater .** | |
| A group of people watching a movie at a theater . | A group of people watching a movie at a theater . |
| A crowd of people are watching a movie indoors . | A man is watching a movie in a theater . |
| A man is watching a movie in a theater . | Some people are sleeping on a sofa in front of the television . |
| **Query: A woman crossing a busy downtown street .** | |
| A lady is walking down a busy street . | A woman walking on the street downtown . |
| A woman is on a crowded street . | A lady is walking down a busy street . |
| A woman walking on the street downtown . | A man and woman walking down a busy street . |
| **Query: A well dressed man standing in front of piece of artwork .** | |
| A well dressed man standing in front of an abstract fence painting . | A man wearing headphones is standing in front of a poster . |
| A man wearing headphones is standing in front of a poster . | A man standing in front of a chalkboard points at a drawing . |
| A man in a blue shirt standing in front of a garage-like structure painted with geometric designs . | A man in a blue shirt standing in front of a garage-like structure painted with geometric designs . |
| **Query: A woman is sitting at a bar eating a hamburger .** | |
| A woman sitting eating a sandwich . | A woman is sitting in a cafe eating lunch . |
| A woman is sitting in a cafe eating lunch . | A woman is eating at a diner . |
| The woman is eating a hotdog in the middle of her bedroom . | A woman is eating her meal at a resturant . |
| **Query: Group of men trying to catch fish with a fishing net .** | |
| Two men are on a boat trying to fish for food during a sunset . | There are three men on a fishing boat trying to catch bass . |
| There are three men on a fishing boat trying to catch bass . | Two men are trying to fish . |
| Two men pull a fishing net up into their red boat . | Two men are on a boat trying to fish for food during a sunset . |

Table 2: Nearest neighbor retrieval results on the SNLI dataset. Given a a query sentence, the left column shows the top-3 retrieved samples based upon the hamming distance with all sentences' binary representations, while the right column exhibits the samples according to the cosine similarity of their continuous embeddings.

# Conclusion

- Among the four distinct strategies to convert pre-trained continuous sentence embeddings into binarized form, the regularized autoencoder augmented with semantic-preserving loss exhibits the best empirical results.
- Random projection or PCA transformation require no training, but demonstrate competitive embedding quality.
- The nearest-neighbor sentence retrieve further validate this framework.