Generalized Boundary Detection Using Compression-Based Analytics

Christina Ting, Richard Field, Tu-Thach Quach, and Travis Bauer

ABSTRACT

We present a new method for boundary detection within sequential data using compression-based analytics. Our approach is to approximate the information distance between two adjacent sliding windows within the sequence. Large values in the distance metric are indicative of boundary locations. A new algorithm is developed, referred to as sliding information distance (SLID), that provides a fast, accurate, and robust approximation to the normalized information distance. A modified smoothed z-score algorithm is used to locate peaks in the distance metric, indicating boundary locations. A variety of data sources are considered, including text and audio, to demonstrate the efficacy of our approach.

DATASETS

We used synthesized datasets with known ground truth in order to quantify algorithm performance:

1. Text randomly selected from English and Spanish translations of United Nations (UN) documents
2. Audio randomly selected from male and female speakers from the LibriSpeech ASR corpus

Each data set contains 100 synthesized boundaries with a section length of 512 tokens.

METHODS

A. Sliding Information Distance (SLID)

- Approximate the information distance (ID) between adjacent subsequences x_k and y_k located at byte k in sequence z [Algorithm 1]

$$z_0 \cdots z_{k-w-1} \underbrace{z_{k-w} \cdots z_{k-1}}_{x_k} \underbrace{z_k \cdots z_{k+w-1}}_{y_k} z_{k+w} \cdots$$

  - Form Lempel-Ziv (L-Z) dictionaries $D(x_k)$ and $D(y_k)$ [Algorithm 2]
  - Compute S_k, the Jaccard distance between them at byte k

$$S_k(z; w) = 1 - \frac{|D(x_k) \cap D(y_k)|}{|D(x_k) \cup D(y_k)|}$$

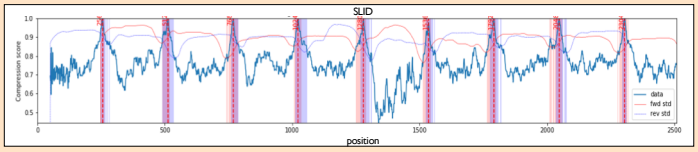
- Slide the local window to the right and repeat the calculation to estimate this distance along z
  - Update to $D(x_{k+1})$ and $D(y_{k+1})$ by drop/append for efficiency [Algorithm 3]

Algorithm 1 Sliding information distance.

function SLID(sequence z, window size w)
  $S \leftarrow [0]$
  for $k = w, \ldots$ **do**
    $x_k \leftarrow z_{k-w}, \ldots, z_{k-1}$
    $y_k \leftarrow z_k, \ldots, z_{k+w-1}$
    if $k == w$ **then**
      $Lx, Dx \leftarrow$ makeLZdict(x_k)
      $Ly, Dy \leftarrow$ makeLZdict(y_k)
    else
      $Lx, Dx \leftarrow$ updateLZdict$(x_k[-1], Lx)$
      $Ly, Dy \leftarrow$ updateLZdict$(y_k[-1], Ly)$
    end if
    S.append$(1 - |Dx \cap Dy|/|Dx \cup Dy|)$
  end for
  return S
end function

B. Boundary Detection

- Any significant change in the SLID score indicates a possible boundary location
- We apply a smoothed z-score algorithm to identify peaks
  - The score is anomalous at k if it exceeds n standard deviations over the running mean of the previous m bytes
- We execute the z-score in forward and reverse to identify regions of local anomalous scores (shaded regions), then take the maximum



Algorithm 2 Initialize the LZ dictionary.

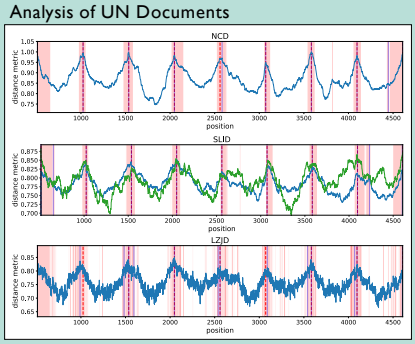
function makeLZdict(sequence b)
  $\ell \leftarrow [\,]$, start $\leftarrow 0$, end $\leftarrow 0$
  while end $< |b|$ **do**
    item $\leftarrow b$[start : end]
    if item $\notin \ell$ **then**
      start $\leftarrow$ end
    end if
    ℓ.append(item)
    end $\leftarrow$ end $+ 1$
  end while
  return ℓ, set(ℓ)
end function

Algorithm 3 Update the LZ dictionary.

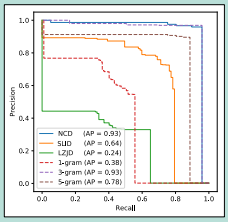
function updateLZdict(token t, list ℓ)
  if $\ell[-1] \notin \ell[0:-2]$ **then**
    item $\leftarrow t$
  else
    item $\leftarrow \ell[-1] + t$
  end if
  $\ell \leftarrow \ell[1:]$
  ℓ.append(item)
  return ℓ, set(ℓ)
end function

RESULTS

Analysis of UN Documents

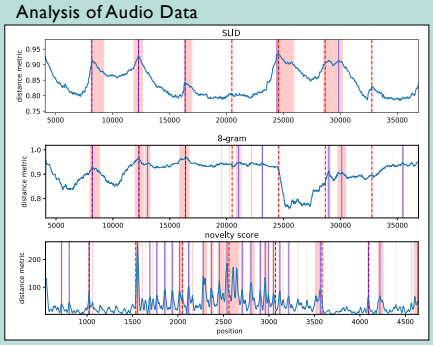


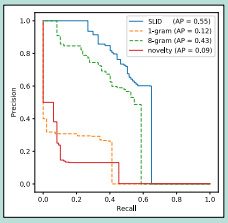
	Time (seconds)	σ	NCD Correlation
NCD	1100	0.0019	1.00
SLID	0.2	0.0016	0.78
LZJD	3.6	0.0123	0.77



- Compare SLID to Normalized Compression Distance (NCD), Lempel-Ziv Jaccard Distance (LZJD), n-Gram, and a Novelty Score (developed for audio signals)
- SLID is both accurate (see Precision-Recall curves) and efficient, while being robust to small changes in information distance
  - A noisy signal is more likely to produce false positives

Analysis of Audio Data





- If data is well-described by an n-gram, as is the case with text, a 3-gram performs best
  - Knowledge of underlying data needed to select optimal n
  - Compression is a good choice when there is no such knowledge
- SLID performs consistently well for both applications
  - Even outperforming the specialized novelty score for audio.

CONCLUSIONS

We have presented a fast, efficient, and robust compression-based method for detecting boundaries in arbitrary sequences of data, including data streams. We have demonstrated the versatility of our approach through several experiments on multiple data sources. As our method is computationally efficient, it is possible to apply different window sizes to obtain a multi-scale structural representation of the underlying data source for boundary detection. This could lead to further improved detection performance or provide the ability to identify coarse and fine-grained boundaries. We defer investigating this problem to our future work.

Supported by the Laboratory Directed Research and Development program at Sandia National Laboratories, a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. SAND2019-3781 C