Unsupervised neural network based feature extraction using weak top-down constraints

Herman Kamper\textsuperscript{1,2}, Micha Elsner\textsuperscript{3}, Aren Jansen\textsuperscript{4}, Sharon Goldwater\textsuperscript{2}

\textsuperscript{1}CSTR and \textsuperscript{2}ILCC, School of Informatics, University of Edinburgh, UK
\textsuperscript{3}Department of Linguistics, The Ohio State University, USA
\textsuperscript{4}HLTCOE and CLSP, Johns Hopkins University, USA

ICASSP 2015
Introduction

- Huge amounts of speech audio data are becoming available online.
- Even for severely under-resourced and endangered languages (e.g. unwritten), data is being collected.
- Generally this data is unlabelled.
- We want to build speech technology on available unlabelled data.
Introduction

▶ Huge amounts of speech audio data are becoming available online.

▶ Even for severely under-resourced and endangered languages (e.g. unwritten), data is being collected.

▶ Generally this data is unlabelled.

▶ We want to build speech technology on available unlabelled data.

▶ Need unsupervised speech processing techniques.
Example application: query-by-example search
Example application: query-by-example search

Spoken query:
Example application: query-by-example search

Spoken query:
Example application: query-by-example search

Spoken query:
Example application: query-by-example search

Spoken query:
Example application: query-by-example search

Spoken query:
Example application: query-by-example search

Spoken query:
Example application: query-by-example search

Spoken query:

What features should we use to represent the speech for such unsupervised tasks?
Supervised neural network feature extraction

But what if we do not have phone class targets to train our network?
Supervised neural network feature extraction

Output: predict phone states

ay ey k v

Input: speech frame(s)
e.g. MFCCs, filterbanks

Feature extractor
(learned from data)

Phone classifier
(learned jointly)

But what if we do not have phone class targets to train our network?

Input: speech frame(s)
e.g. MFCCs, filterbanks
Supervised neural network feature extraction

Output: predict phone states

ay  ey  k  v

Feature extractor (learned from data)

Feature extractor (learned from data)

Phone classifier (learned jointly)

Input: speech frame(s)
e.g. MFCCs, filterbanks

But what if we do not have phone class targets to train our network?
Supervised neural network feature extraction

Output: predict phone states

ay ey k v

Phone classifier (learned jointly)

Feature extractor (learned from data)

Input: speech frame(s)
e.g. MFCCs, filterbanks

But what if we do not have phone class targets to train our network?
Supervised neural network feature extraction

Input: speech frame(s)
e.g. MFCCs, filterbanks
Output: predict phone states

Feature extractor (learned from data)
Phone classifier (learned jointly)

But what if we do not have phone class targets to train our network?
Weak supervision: unsupervised term discovery
Weak supervision: unsupervised term discovery

Can we use these discovered word pairs to provide us with weak supervision?
Weak supervision: unsupervised term discovery

Can we use these discovered word pairs to provide us with weak supervision?
Weak supervision: unsupervised term discovery

Can we use these discovered word pairs to provide us with weak supervision?
Weak supervision: unsupervised term discovery

Can we use these discovered word pairs to provide us with weak supervision?
Weak supervision: unsupervised term discovery

Can we use these discovered word pairs to provide us with weak supervision?
Weak supervision: unsupervised term discovery

Can we use these discovered word pairs to provide us with weak supervision?
Weak supervision: align the discovered word pairs

Use correspondence idea from [Jansen et al., 2013]
Weak supervision: align the discovered word pairs

Use correspondence idea from [Jansen et al., 2013]:

![Image 1]

![Image 2]
Weak supervision: align the discovered word pairs

Use correspondence idea from [Jansen et al., 2013]:
Weak supervision: align the discovered word pairs

Use correspondence idea from [Jansen et al., 2013]:
Autoencoder (AE) neural network

A normal autoencoder neural network is trained to reconstruct its input. Output is same as input.

This reconstruction criterion can be used to pretrain a deep neural network.
Autoencoder (AE) neural network

A normal autoencoder neural network is trained to reconstruct its input. Output is same as input.

This reconstruction criterion can be used to pretrain a deep neural network.

Input speech frame

A normal autoencoder neural network is trained to reconstruct its input.
Autoencoder (AE) neural network

This reconstruction criterion can be used to pretrain a deep neural network.
The correspondence autoencoder (cAE) takes a frame from one word, and tries to reconstruct the corresponding frame from the other word in the pair.

In this way we learn an unsupervised feature extractor using the weak word-pair supervision.
The correspondence autoencoder (cAE)

Frame from one word

Frame from other word in pair

Unsupervised feature extractor

In this way we learn an unsupervised feature extractor using the weak word-pair supervision.
Complete unsupervised cAE training algorithm

1. Train stacked autoencoder (pretraining)
2. Unsupervised term discovery
3. Align word pair frames
4. Train correspondence autoencoder

Unsupervised feature extractor
Evaluation of features: the same-different task
Evaluation of features: the same-different task

“apple”
“pie”
“grape”
“apple”
“apple”
“like”

DTW distance:
d1
d2
d3
✓
×
✓

Treat as terms to search

Determine if di < threshold?
predict:
different
same
same

 ✓
×
✓


Evaluation of features: the same-different task

Treat as query

“apple”
“pie”
“grape”
“apple”
“apple”
“like”

DTW distance:

Treat as query

“apple”

“pie”
“grape”
“apple”
“apple”
“like”

DTW distance:

Treat as query

“apple”

“pie”
“grape”
“apple”
“apple”
“like”

DTW distance:

Treat as query

“apple”

“pie”
“grape”
“apple”
“apple”
“like”

DTW distance:

Treat as query

“apple”

“pie”
“grape”
“apple”
“apple”
“like”

DTW distance:

Treat as query

“apple”

“pie”
“grape”
“apple”
“apple”
“like”

DTW distance:

Treat as query

“apple”

“pie”
“grape”
“apple”
“apple”
“like”

DTW distance:
Evaluation of features: the same-different task
Evaluation of features: the same-different task

“apple”  “pie”  “grape”  “apple”  “apple”  “like”

Treat as query

“pie”  “pie”  “pie”  “pie”  “pie”  “pie”

“grape”  “grape”  “grape”  “grape”  “grape”  “grape”

“like”  “like”  “like”  “like”  “like”  “like”

DTW distance:

d1 ✓
d2
✓
×
✓

predict:
different
same
different
same
different

”apple”  “pie”  “grape”  “apple”  “apple”  “like”
Evaluation of features: the same-different task

“apple”
“pie”
“grape”
“apple”
“apple”
“like”

“pie”
“grape”
“apple”
“apple”
“like”

“apple”

“apple”

“pie”

“grape”

“apple”

“apple”

“like”

✓
×

DTW distance:

d1

d2

✓

×

✓

Treat as query

“pie”

“grape”

“apple”

“like”

Treat as terms to search

“apple”

“pie”

“grape”

“apple”

“apple”

“like”

“pie”

“grape”

“apple”

“apple”

“like”
Evaluation of features: the same-different task

DTW distance:

Treat as query

Treat as terms to search
Evaluation of features: the same-different task

<table>
<thead>
<tr>
<th>“apple”</th>
<th>“pie”</th>
<th>“grape”</th>
<th>“apple”</th>
<th>“apple”</th>
<th>“like”</th>
</tr>
</thead>
</table>

DTW distance: $d_i < \text{threshold?}$

predict:

$\text{different}$
Evaluation of features: the same-different task

"apple"  "pie"  "grape"  "apple"  "apple"  "like"

DTW distance: $d_i < \text{threshold?}$

predict:

different ✓

$\text{same}$
Evaluation of features: the same-different task

<table>
<thead>
<tr>
<th>“apple”</th>
<th>“pie”</th>
<th>“grape”</th>
<th>“apple”</th>
<th>“apple”</th>
<th>“like”</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

DTW distance:

<table>
<thead>
<tr>
<th>$d_i &lt; \text{threshold}$?</th>
<th>predict:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>different ✓</td>
</tr>
</tbody>
</table>
## Evaluation of features: the same-different task

<table>
<thead>
<tr>
<th>“apple”</th>
<th>“pie”</th>
<th>“grape”</th>
<th>“apple”</th>
<th>“apple”</th>
<th>“like”</th>
</tr>
</thead>
</table>

DTW distance: $d_1$

- $d_i < \text{threshold?}$
  - predict:
    - different ✓

![Waveforms and DTW calculations](image)
Evaluation of features: the same-different task

<table>
<thead>
<tr>
<th>“apple”</th>
<th>“pie”</th>
<th>“grape”</th>
<th>“apple”</th>
<th>“apple”</th>
<th>“like”</th>
</tr>
</thead>
</table>

DTW distance:

<table>
<thead>
<tr>
<th>$d_1$</th>
<th>$d_2$</th>
</tr>
</thead>
</table>

$d_i <$ threshold?
predict:

<table>
<thead>
<tr>
<th>different ✓</th>
<th>same</th>
</tr>
</thead>
</table>
Evaluation of features: the same-different task

<table>
<thead>
<tr>
<th>DTW distance:</th>
<th>$d_i &lt; \text{threshold?}$</th>
<th>predict:</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_1$</td>
<td>different</td>
<td>✓</td>
</tr>
<tr>
<td>$d_2$</td>
<td>same</td>
<td>✗</td>
</tr>
</tbody>
</table>

Treat as query: “apple” “pie” “grape” “apple” “apple” “like”

Treat as terms to search: “pie” “grape” “apple” “apple” “like”

“apple” “pie” “grape” “apple” “apple” “like”

“apple” “pie” “grape” “apple” “apple” “like”
Evaluation of features: the same-different task

<table>
<thead>
<tr>
<th>“apple”</th>
<th>“pie”</th>
<th>“grape”</th>
<th>“apple”</th>
<th>“apple”</th>
<th>“like”</th>
</tr>
</thead>
</table>

DTW distance:

d_1 ✓

di < threshold?
predict:
different ✓

“apple”

Treat as query

“pie”

“grape”

“apple”

“apple”

“like”

Treat as terms to search

DTW distance:

d_2 ×

d_1 < threshold?
predict:
different ✓

different ✓
Evaluation of features: the same-different task

<table>
<thead>
<tr>
<th>DTW distance:</th>
<th>$d_i &lt; \text{threshold?}$</th>
<th>$\text{predict:}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_1$</td>
<td>different $\checkmark$</td>
<td></td>
</tr>
<tr>
<td>$d_2$</td>
<td>same $\times$</td>
<td></td>
</tr>
<tr>
<td>$d_3$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Evaluation of features: the same-different task

<table>
<thead>
<tr>
<th>DTW distance:</th>
<th>$d_i &lt;$ threshold?</th>
<th>predict:</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_1$</td>
<td></td>
<td>different ✓</td>
</tr>
<tr>
<td>$d_2$</td>
<td></td>
<td>same ×</td>
</tr>
<tr>
<td>$d_3$</td>
<td></td>
<td>same</td>
</tr>
</tbody>
</table>
Evaluation of features: the same-different task

<table>
<thead>
<tr>
<th>“apple”</th>
<th>“pie”</th>
<th>“grape”</th>
<th>“apple”</th>
<th>“apple”</th>
<th>“like”</th>
</tr>
</thead>
</table>

DTW distance:

<table>
<thead>
<tr>
<th>$d_1$</th>
<th>$d_2$</th>
<th>$d_3$</th>
<th>$d_4$</th>
</tr>
</thead>
</table>

$d_i < \text{threshold?}$

<table>
<thead>
<tr>
<th>predict:</th>
</tr>
</thead>
<tbody>
<tr>
<td>different ✓</td>
</tr>
<tr>
<td>same ×</td>
</tr>
<tr>
<td>same ✓</td>
</tr>
</tbody>
</table>
**Evaluation of features: the same-different task**

<table>
<thead>
<tr>
<th>Treatment</th>
<th>DTW distance:</th>
<th>$d_i &lt; \text{threshold}$?</th>
<th>predict:</th>
</tr>
</thead>
<tbody>
<tr>
<td>“apple”</td>
<td>$d_1$</td>
<td>different ✓</td>
<td></td>
</tr>
<tr>
<td>“pie”</td>
<td>$d_2$</td>
<td>same ×</td>
<td></td>
</tr>
<tr>
<td>“grape”</td>
<td>$d_3$</td>
<td>same ✓</td>
<td></td>
</tr>
<tr>
<td>“apple”</td>
<td>$d_4$</td>
<td>different ×</td>
<td></td>
</tr>
<tr>
<td>“like”</td>
<td>$d_N$</td>
<td>different ✓</td>
<td></td>
</tr>
</tbody>
</table>

Treat as query

Treat as terms to search
Evaluation of features: the same-different task

- Each term is treated in turn as the query.
- The threshold is varied to obtain a precision-recall curve.
- The area under the precision-recall curve is used as the final evaluation metric, referred to as average precision (AP).
- AP is higher for feature representations which are better able to associate words of the same type and discriminate between words of different types.
- AP has been shown to correlate well with phone recognition error rates [Carlin et al., 2011] and has been used in several other unsupervised studies.
Baseline: partitioned universal background model

Use posteriorogram features from the partitioned universal background model (UBM) as baseline [Jansen et al., 2013].
Evaluation

- Speech from Switchboard is used for evaluation.

- Pretraining data: 23 hours of untranscribed speech.

- We consider two sets of word pairs for training the cAE:
  1. 100k gold standard word pairs.
  2. 80k word pairs discovered using unsupervised term discovery (UTD).

- Test set for same-different evaluation: 11k word tokens, 60.7M pairs, 3% produced by same speaker.
Evaluation

- Speech from Switchboard is used for evaluation.
- Pretraining data: 23 hours of untranscribed speech.
- We consider two sets of word pairs for training the cAE:
  1. 100k gold standard word pairs.
  2. 80k word pairs discovered using unsupervised term discovery (UTD).
- Test set for same-different evaluation: 11k word tokens, 60.7M pairs, 3% produced by same speaker.
- Neural network architecture (optimized on development set):
  39-dimensional single-frame MFCC input features, 13 layers, 100 hidden units per layer, take features from the fourth-last encoding layer.
## Comparison with baseline: gold standard word pairs

<table>
<thead>
<tr>
<th>Features</th>
<th>Average precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFCCs with CMVN</td>
<td>0.214</td>
</tr>
<tr>
<td>UBM with 1024 components [Jansen et al., 2013]</td>
<td>0.222</td>
</tr>
<tr>
<td>1024-UBM partitioned 100 components [Jansen et al., 2013]</td>
<td>0.286</td>
</tr>
<tr>
<td>100-unit, 13-layer stacked autoencoder</td>
<td>0.215</td>
</tr>
<tr>
<td>100-unit, 13-layer correspondence autoencoder</td>
<td><strong>0.469</strong></td>
</tr>
<tr>
<td>Supervised NN, 10 hours [Carlin et al., 2011]</td>
<td>0.439</td>
</tr>
<tr>
<td>Supervised NN, 100 hours [Carlin et al., 2011]</td>
<td>0.516</td>
</tr>
</tbody>
</table>
## Evaluation using terms from unsupervised term discovery

<table>
<thead>
<tr>
<th>Features</th>
<th>Average precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFCCs with CMVN</td>
<td>0.214</td>
</tr>
<tr>
<td>Best of [Jansen et al., 2013] using gold standard word pairs</td>
<td>0.286</td>
</tr>
<tr>
<td>Correspondence autoencoder trained on gold standard word pairs</td>
<td>0.469</td>
</tr>
<tr>
<td>Correspondence autoencoder trained on UTD pairs</td>
<td><strong>0.341</strong></td>
</tr>
<tr>
<td>Supervised NN, 10 hours [Carlin et al., 2011]</td>
<td>0.439</td>
</tr>
<tr>
<td>Supervised NN, 100 hours [Carlin et al., 2011]</td>
<td>0.516</td>
</tr>
</tbody>
</table>
Summary and conclusion

▶ Introduced the correspondence autoencoder (cAE), a novel neural network which can be trained unsupervised on unlabelled speech data.

▶ Evaluated the network in a word discrimination task.

▶ Showed 64% relative improvement over a previous state-of-the-art GMM system.

▶ Come to within 23% of supervised baseline.

▶ Future work: apply in further unsupervised speech processing tasks; how can the correspondence idea be used in other neural network structures?
https://github.com/kamperh/speech_correspondence/
Choosing the network architecture

Development set cAE performance using gold standard word pairs. Features were taken from the fourth-last to second-last encoding layers.