Multitask Learning for Semantic Acoustical Embedding

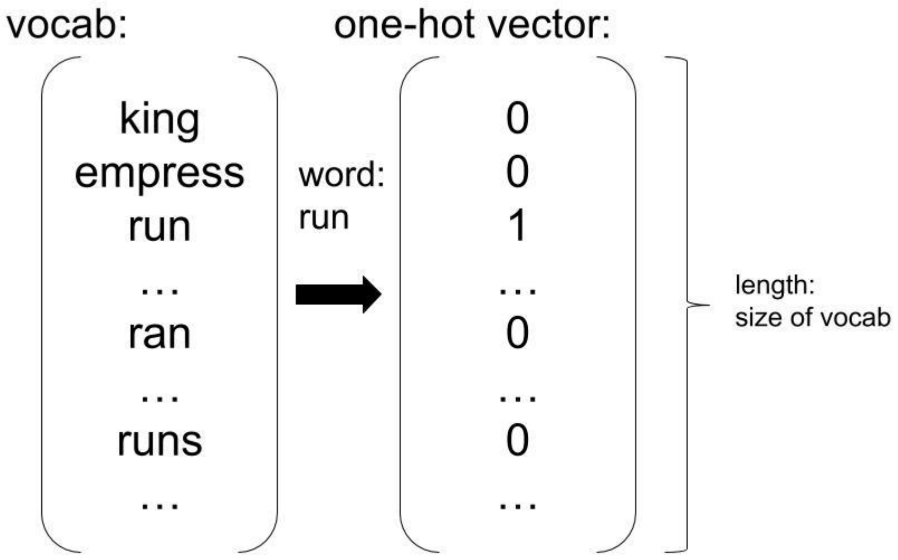
Daniel Clothiaux, Qinlan Shen, Wenbo Zhao

Background: Word Vectors

Question: How do we represent words when they are the input to a system?

Simplest possible answer: one-hot vectors

- Assign each word in vocab to an index
- Represent words with a vector the size of the vocab with a ‘1’ at that index



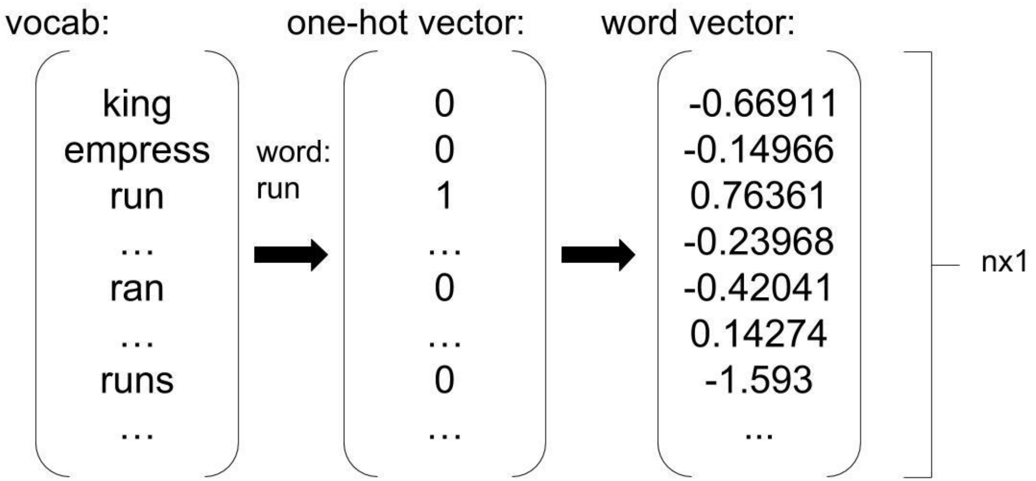
Problems with One-Hot Vectors

But…

- The vector dimension is very large
  - normally ***tens or hundreds of thousands !***
- No information is captured about syntactic or semantic properties
  - Examples: nouns vs. verbs, “run” vs. “runs”
- Out of vocabulary words are not handled
  - Can be dealt with by mapping them to an ‘UNK’ token

Solution

If we could map each word into a dense nx1 dimensional vector where words with similar properties are close together...



...we could then use the nx1 dimensional vectors as our word vector

Solution (continued)

How do we obtain these vectors?

- General idea: Train a model under the distributional hypothesis
  - **GloVe (global vector) vectors**: Train a log-bilinear model that aims for the dot product of two vectors to be the probability of the word co-occurrences (Pennington et al. 2014)
  - **Skip-grams**: Train model to predict word based on the surrounding context, take hidden layer as a vector (Mikolov et al. 2013)

I ▪ to the store

P(ran| I ▪ to the store)

Speech Embeddings

Given a segment of acoustic data, map it into a dense nx1 dimensional vector where segments that sound similar are close together

- If results from a speech recognition system are used in downstream tasks, how should we represent the results?
  - Word embeddings!
- But the speech signal carries more information that just text
  - Dialect, gender, age of the speaker, and perhaps even education level
  - We might want to include this as well

Why Multitask Learning?

- We want our acoustic word embeddings to encode all of this extra information
- Potential solution: break each class of information into its own task, then train against each task

- How do we train against many tasks? Multitask learning!
    - Create classifiers/regressors for each task, then send the output of a hidden shared layer into each of these tasks
    - Train by selecting an instance and a random task, then training the network against that task
    - Has been used for related NLP tasks (Collobert and Weston 2008)(Liu et al. 2015)

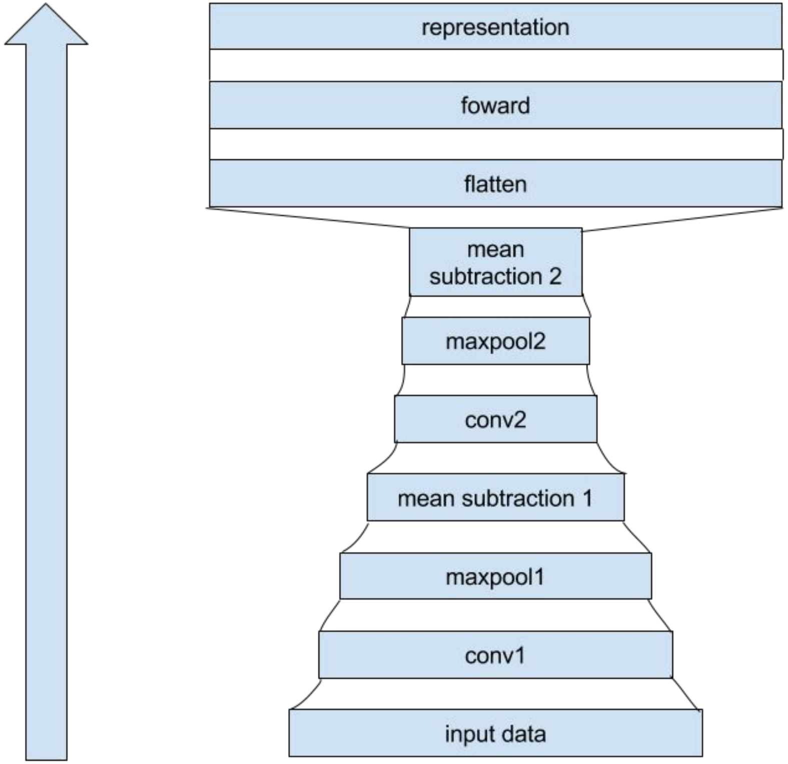
Setup

- Dataset: Switchboard-1 release 2 (LDC97S62) dataset
  - Telephone conversations among 543 speakers from all across the U.S.
- Tasks
  - Word recognition
  - Word semantic prediction
  - Gender prediction
  - Speaker identification
  - Age prediction
  - Education prediction
  - Dialect prediction

Preprocessing - Alignment

- Align filterbank features with word transcriptions
  - Generate filterbank features: segment → preemphasize → power spectrum → log Mel transform → <num_frames, 40>
  - **Hard align**: divide the features evenly for each word in the utterance
  - **Soft align**: train an acoustic model using HMM → Viterbi decoding to decide word boundary

Multitask Learning Architecture (Shared Component)



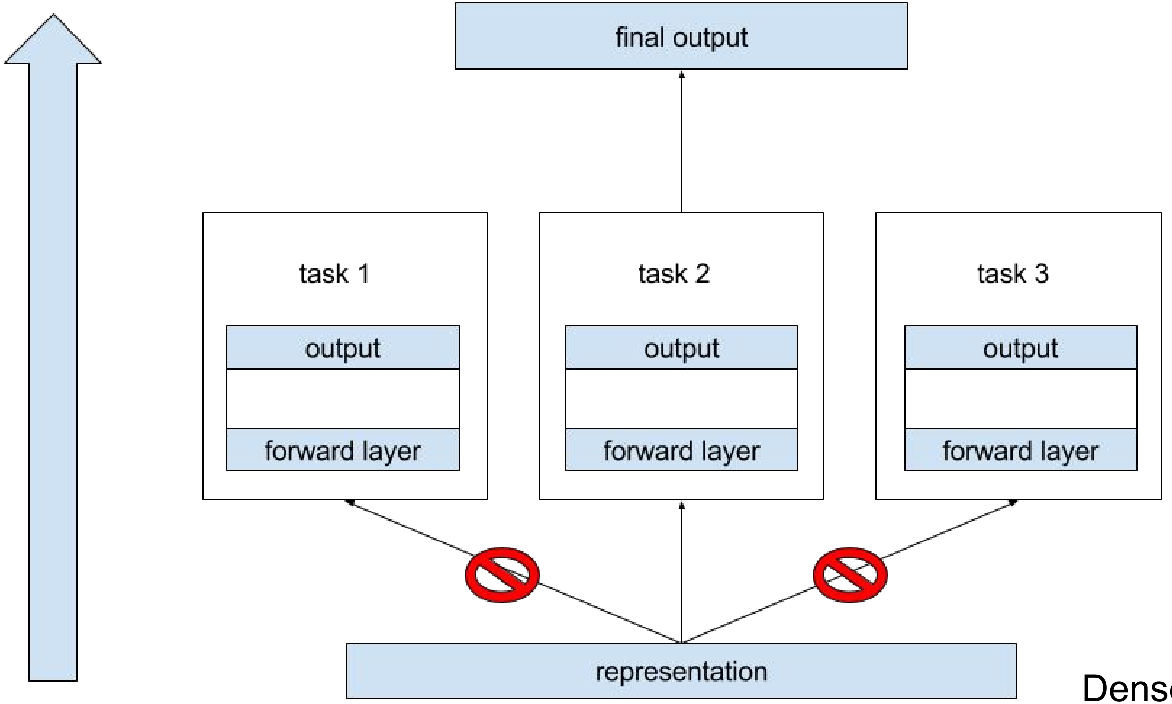
Dense 1024

Filter (5,20,5)

Filter (10,40,9), Stride 2

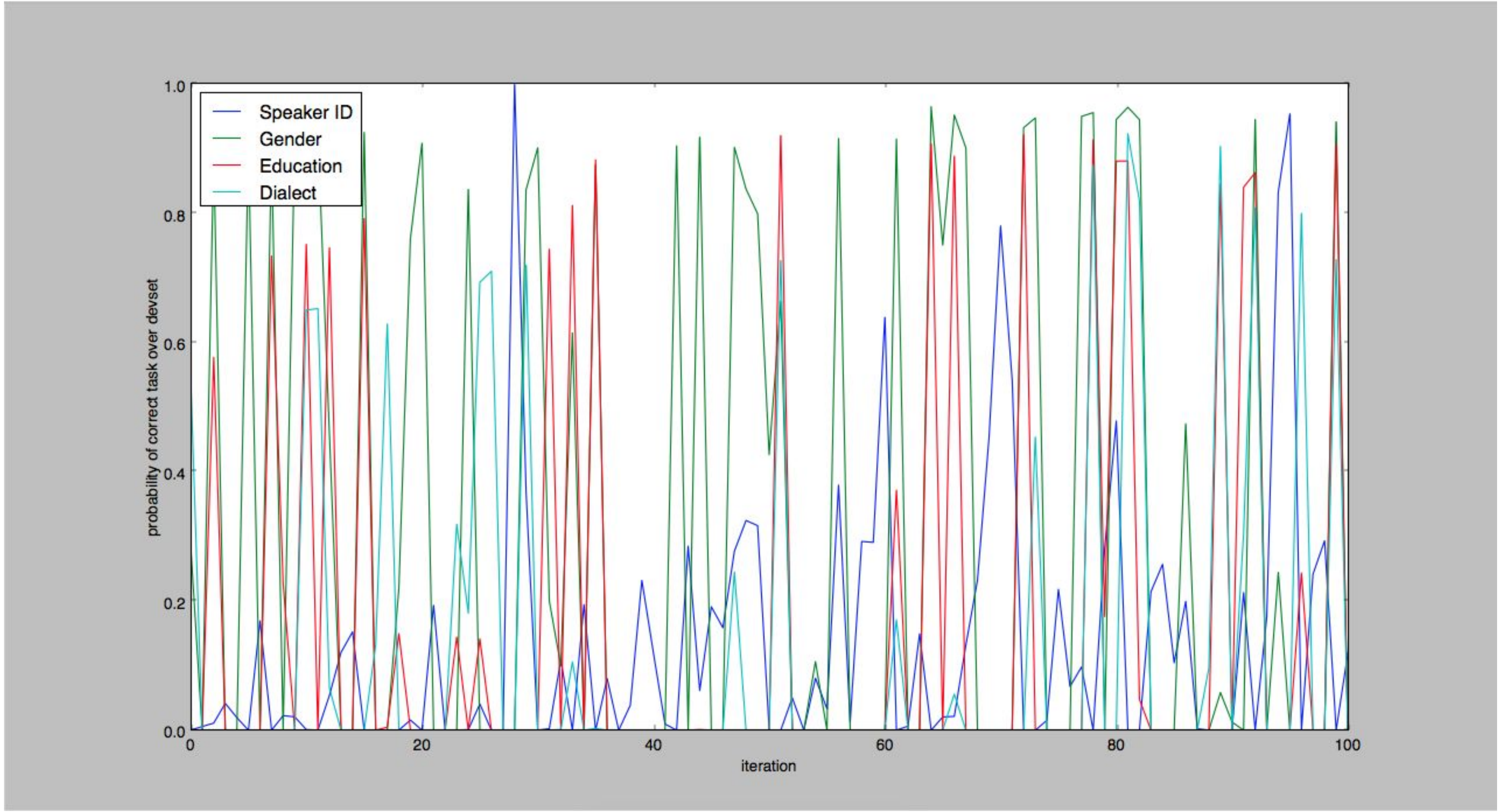
(N,100,40)

Multitask Learning Architecture (Task-Specific)

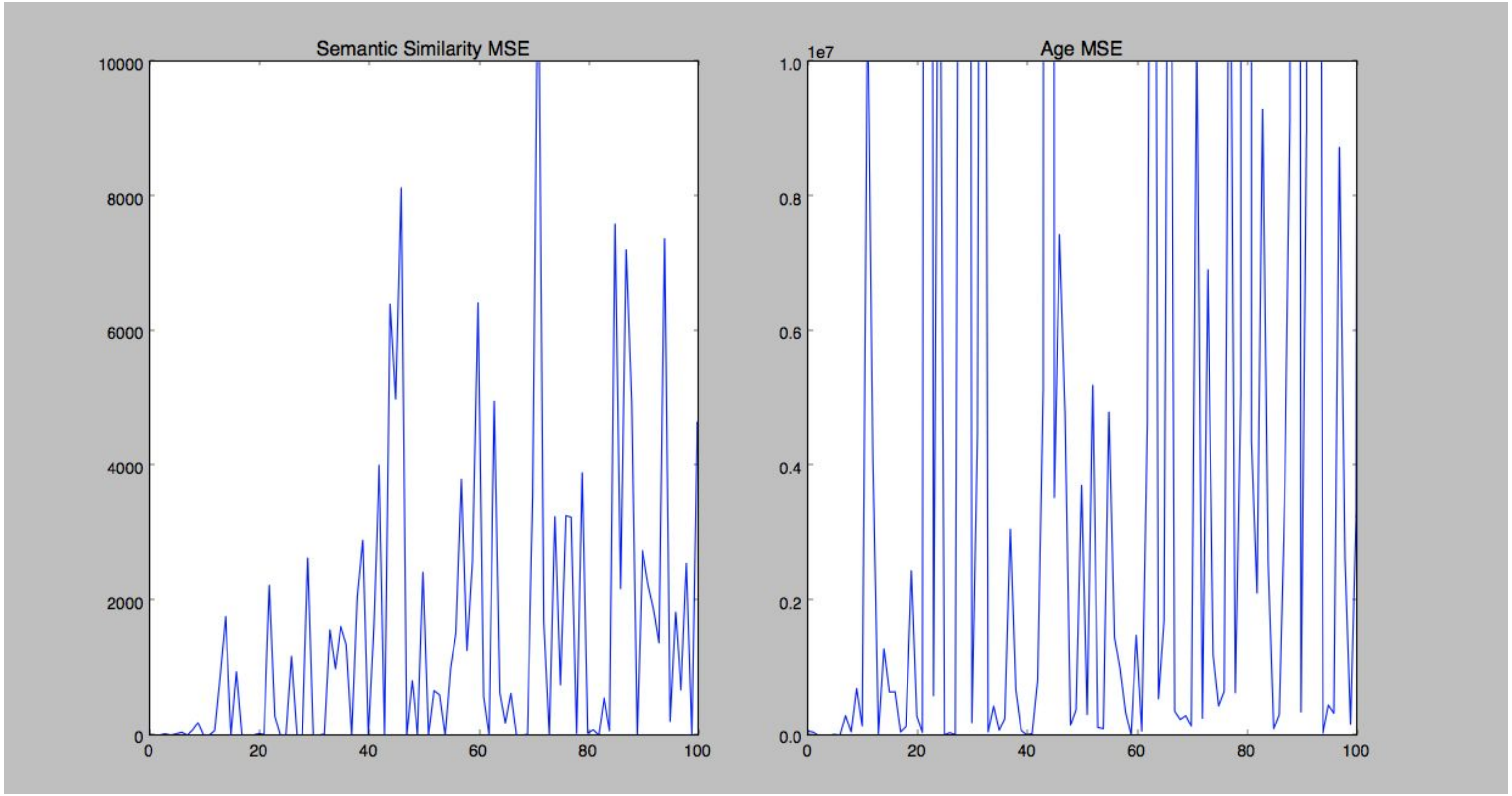


Dense 1024

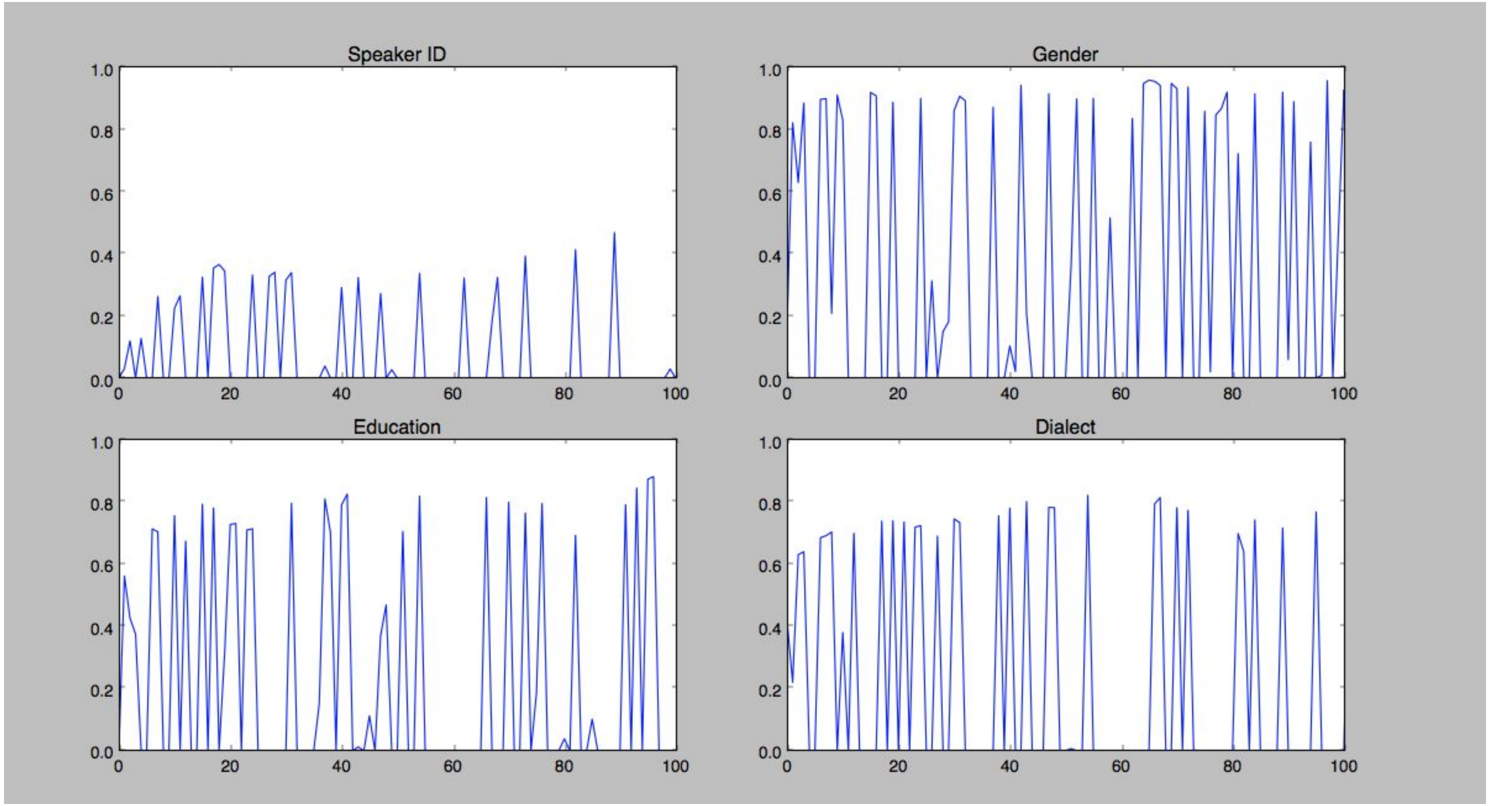
Multitask Results



Task-Specific Results



Task Specific Results (continued)



Single-Task Results

Tab.1 Single task results

(Batch SGD, batch size 20, lr: 0.1, 1000 epoch, on GPU)

Task	Age	Gender	Word	Education	Dialect	Speaker
Error	0.01510	0.02093	0.1592	0.1662	0.2093	0.2869

Discussion

- The general convolutional architecture is capable of learning each of the tasks in isolation
- Learning tasks pulls other tasks away from their correct solution
  - This results in spikes where the model will learn one task and forget the others
  - Spikes ‘larger’ in MSE than in probability as MSE isn’t bounded-being pulled away hurts more
- Possible explanations
  - The tasks we are learning are less related to each other than in other NLP multitask architectures. For example, dialect and age are less related than POS tag and semantics
  - Small data size
  - False alignment → bad word recognition, dialect prediction

Future Steps

- Continue training the model
  - Increase data size
  - Better alignment
  - Parameter tuning: try different filter/layer sizes
- Try adding more “insulation” feedforward layers between the shared layer and task-specific layers
- Separate speaker-dependent and -independent info for better speaker identification
- Simultaneously train on all tasks for an instance
  - By training tasks together, force the network to minimize error of all of them together
  - Potential challenge:
    - Word recognition, age, and semantic similarity are trained against MSE, while the others use cross entropy

Sources

Word Embeddings:

Skip Grams: Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." *arXiv preprint arXiv:1301.3781* (2013).

Glove Vectors: Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. "Glove: Global Vectors for Word Representation." *EMNLP*. 2014. http://nlp.stanford.edu/projects/glove/

Sources (continued)

Speech Recognition:

Preeti Saini and Parneet Kaur. Automatic speech recognition: A review. International journal of Engineering Trends & Technology, pages 132–136, 2013.

Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal Processing Magazine, 29(6):82–97, 2012.

Sources (continued)

Multitask learning:

Collobert, Ronan, Jason Weston. “A unified architecture for natural language processing: Deep neural networks with multitask learning.” *ICML*. 2008.

Liu, Xiaodong, et al. “Representation learning using multi-task deep neural networks for semantic classification and information retrieval.” *NAACL*. 2015.

Sources (continued)

Acoustic Embeddings:

Bengio, Samy, and Georg Heigold. "Word embeddings for speech recognition." (2014).

Ghannay, Sahar, Yannick Esteve, and Nathalie Camelin. "Evaluation of acoustic word embeddings." *ACL.* 2016.