About “Embeddings”

Deep Learning, Northwestern University spring 2023 Bryan Pardo

What's an "embedding"?

- A neural network embodies a function $f: X \rightarrow X'$
- X is the input to the net and X' is the set of activations of some layer of the net.
- Colloquially, we refer to X' as an "embedding" of the input to the net.
- Some even call any vector of real numbers an "embedding" (e.g. calling a hand-made one-hot vector an "embedding")
- Ideally, want an embeddings to have meaningful "groupings"

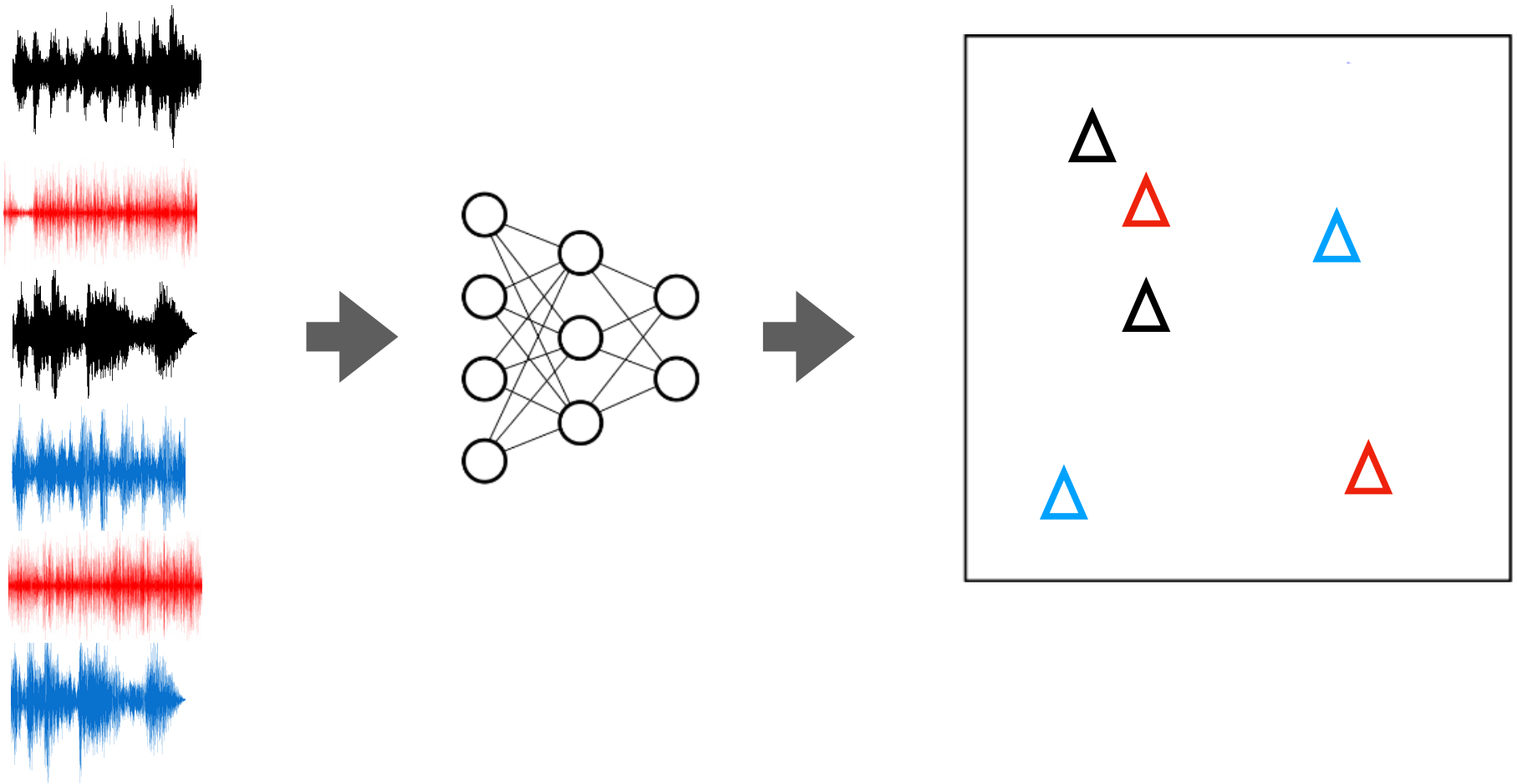
Common design goals

- We want to provide task-relevant similarity judgements
- We want the ability to compare things we've never trained on

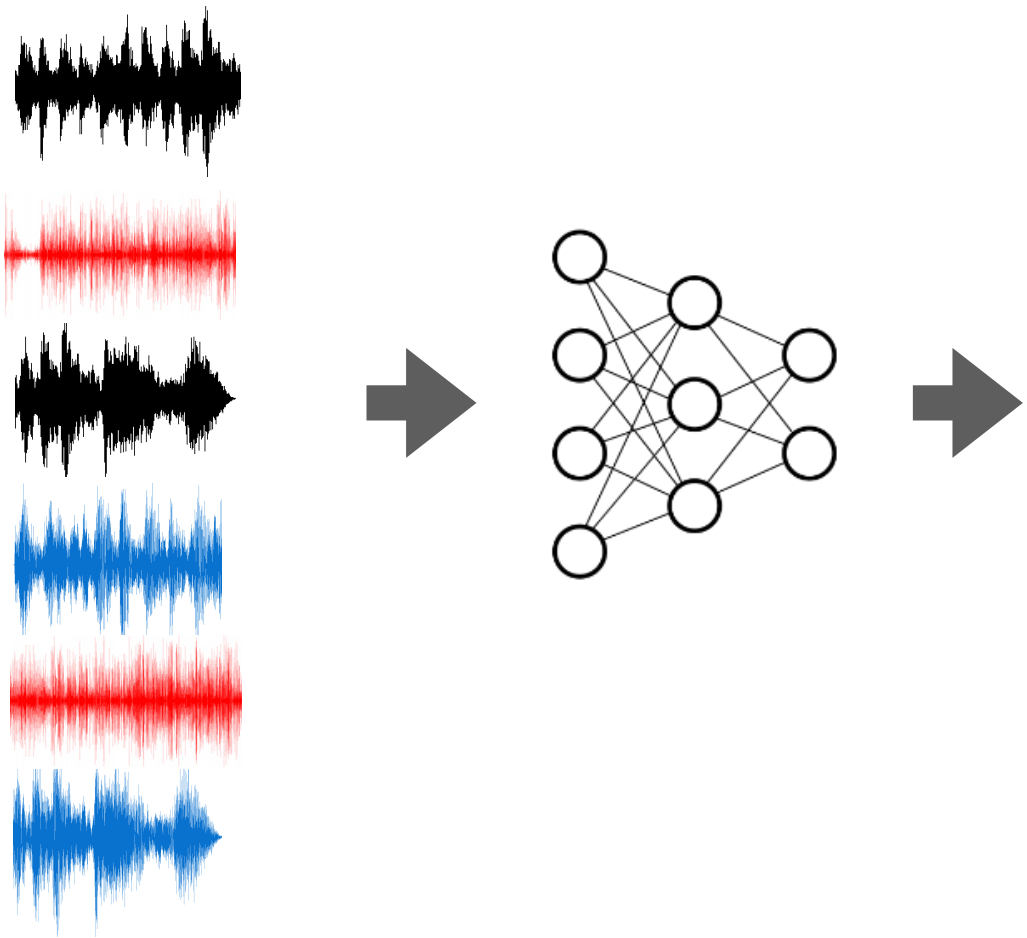
Training a Dedicated Embedding Network

Example: VoiceID

Train an embedding net

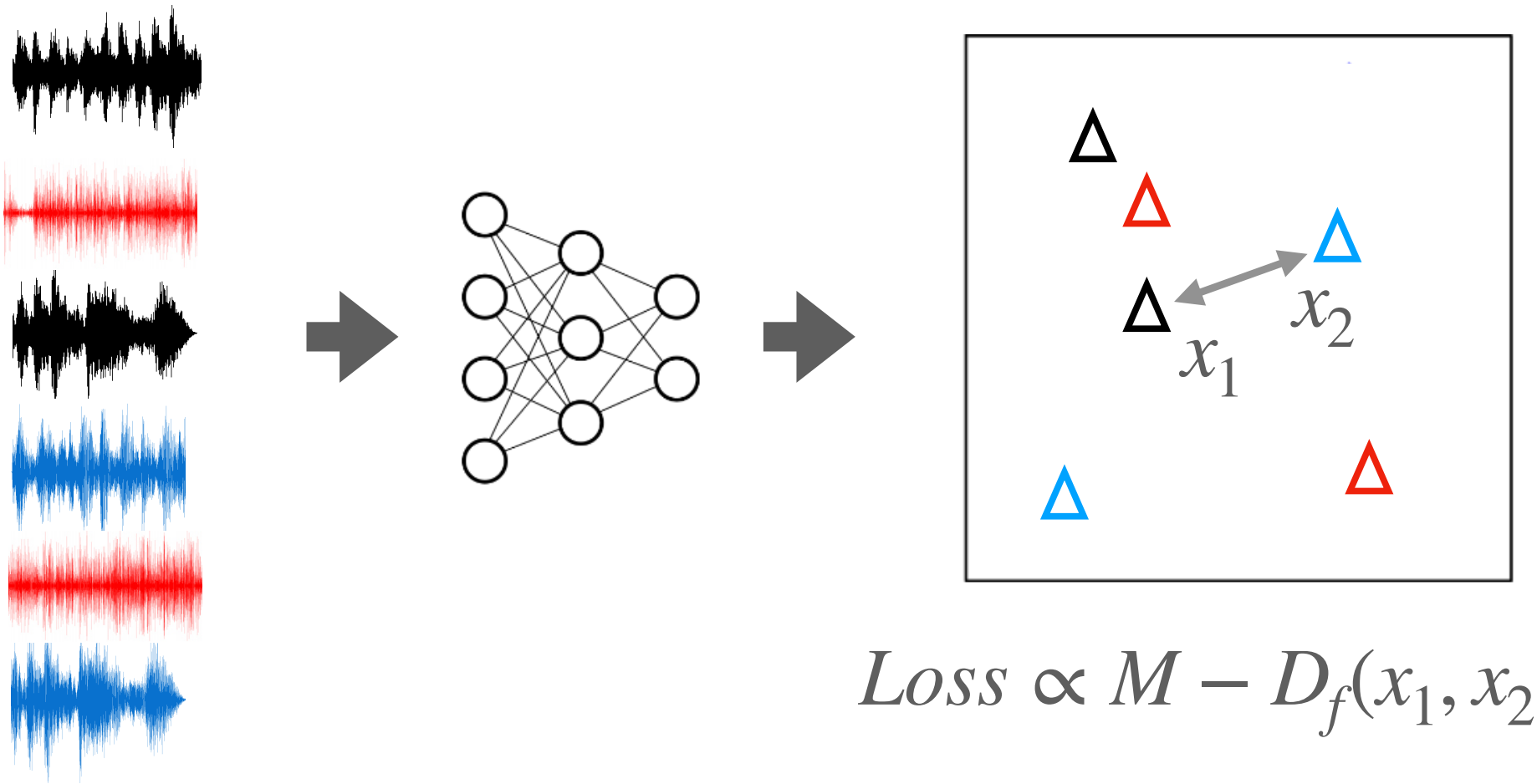


Move things from the same group closer



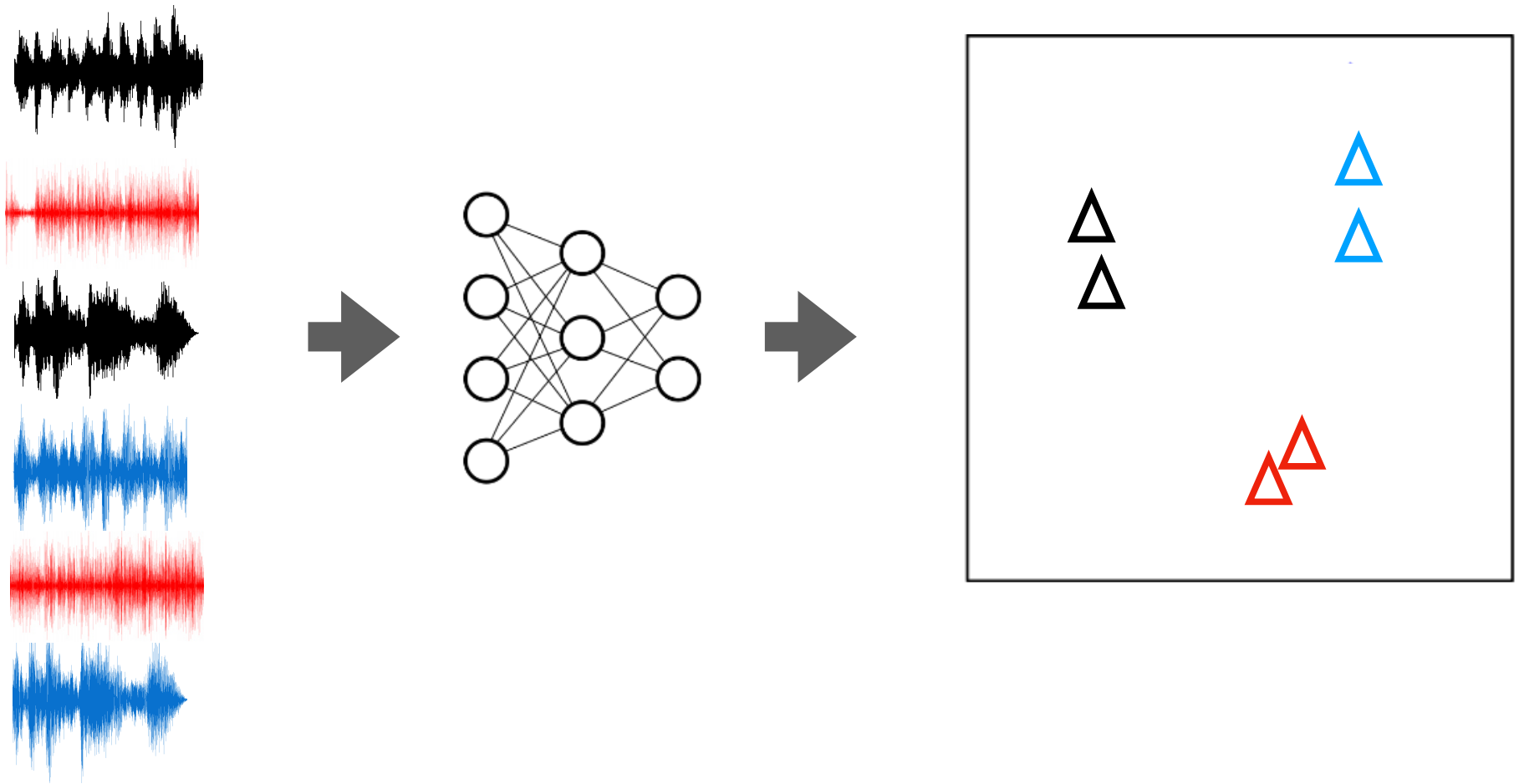
$$Loss \propto D_f(x_1, x_2)$$

Push things from different groups apart

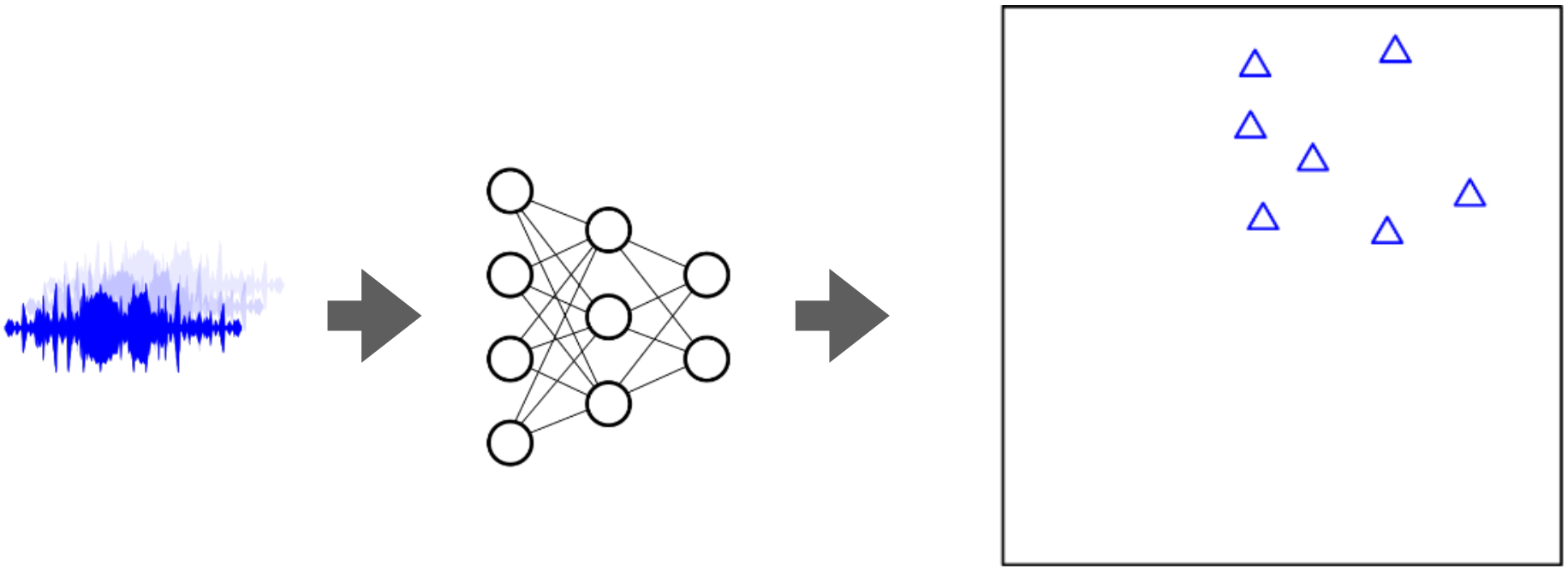


$$Loss \propto M - D_f(x_1, x_2)$$

Train an embedding net

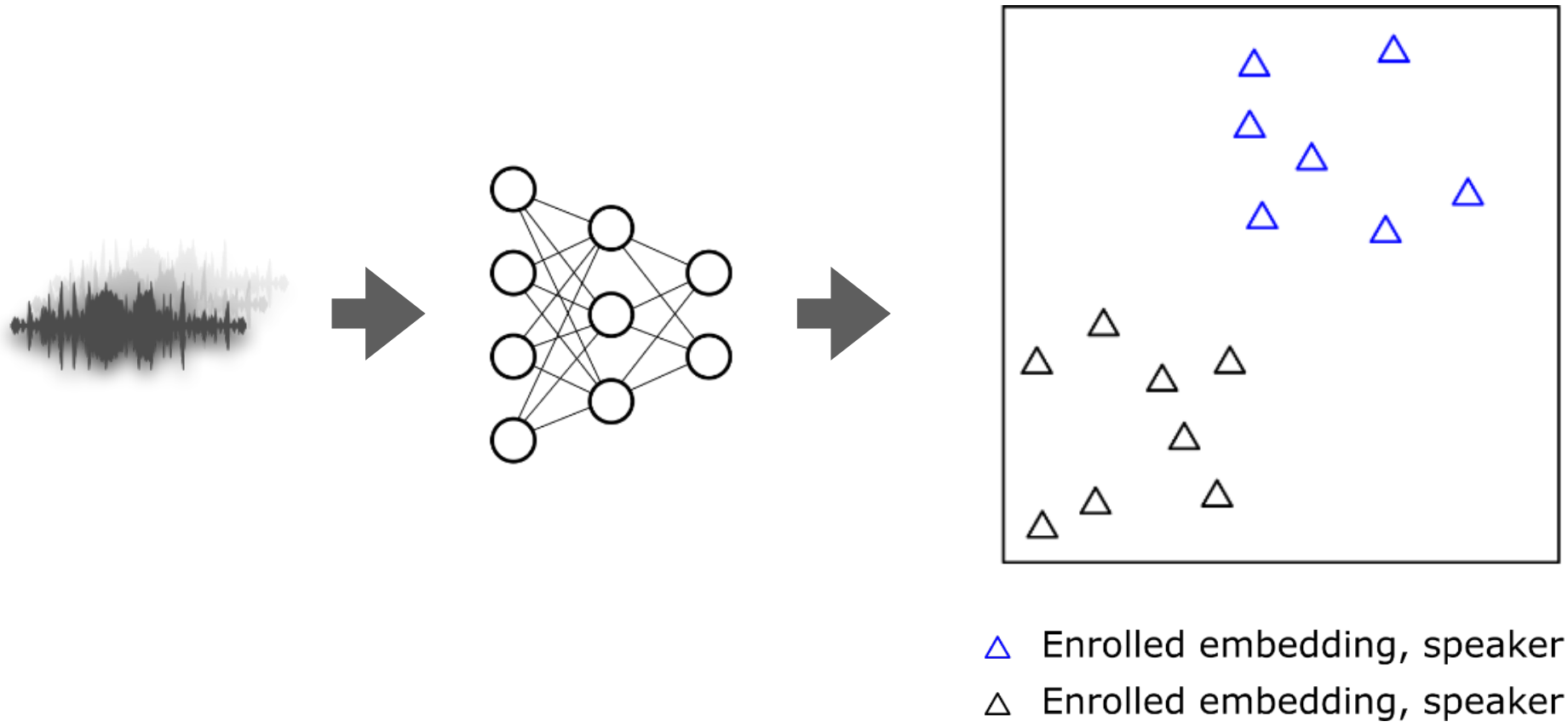


Enroll a voice



△ Enrolled embedding, speaker A

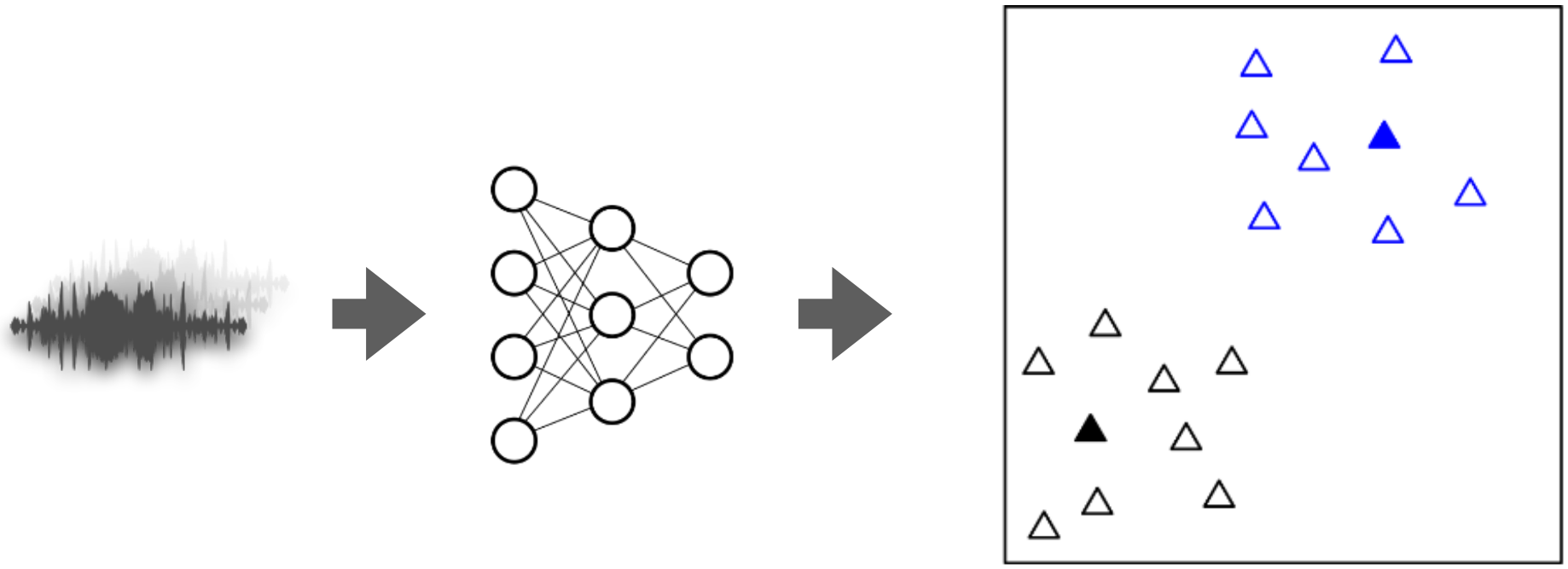
Enroll more voices



△ Enrolled embedding, speaker A

△ Enrolled embedding, speaker B

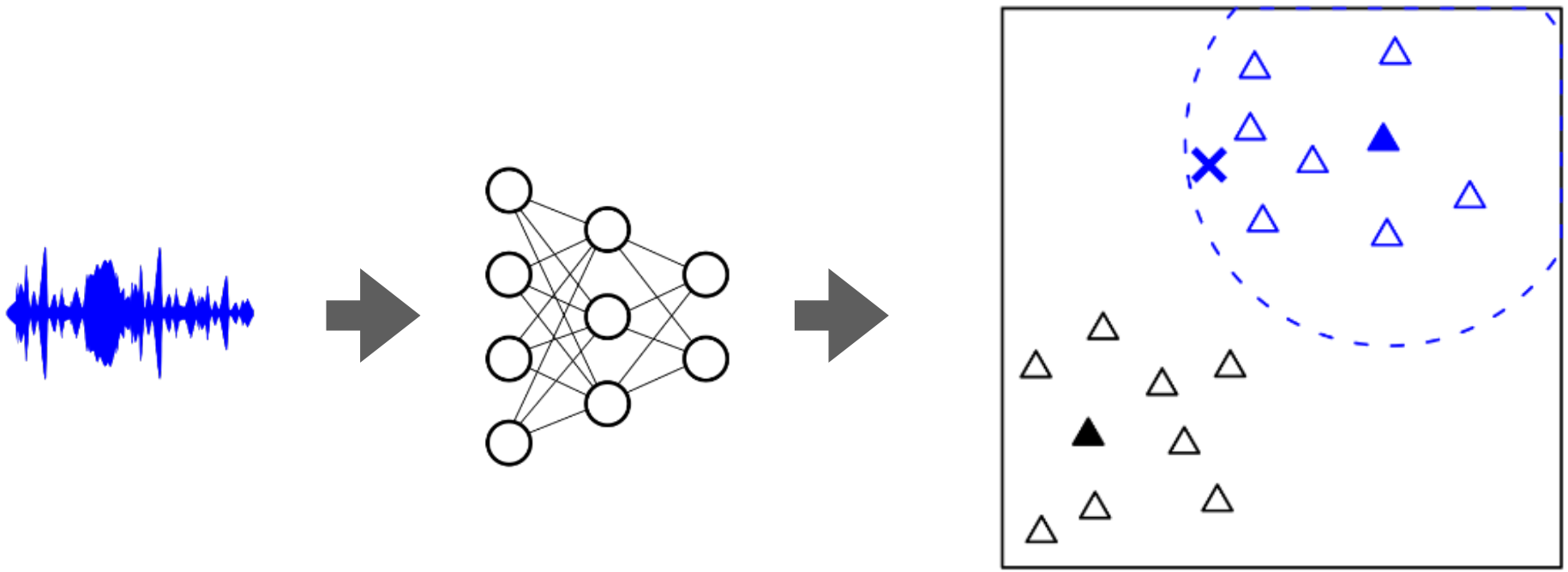
Find centroids



▲ Enrolled embedding centroid, speaker A

▲ Enrolled embedding centroid, speaker B

Use nearest-neighbor search to pick a group



× Query embedding, speaker A

Finding neighbors: Cosine similarity

- Often used with embeddings
- A kind of normalized dot product
- Higher values = more similar

$$S_C(A, B) := \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}},$$

What if we remove the normalization?

- Often used with embeddings
- A kind of ~~normalized~~ dot product
- Higher values = more similar
- How would scale affect things?

$$S(A, B) = \sum_i A_i B_i$$

Making an embedding for words

- With voices, we know which 2 things should go together.
- How do I decide this for words?
- Is there a way to make training go faster than to train an entire neural network?

Example: GloVE

Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. "Glove: Global vectors for word representation." *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014.

The big ideas: distance and co-occurence

- Words that are occur together in text are related (big assumption).

- Measuring co-occurrence of words tells us how related they are.
  - E.G. “Bryan Pardo” vs “Bryan Billionare”

- Words beyonds some cutoff distance aren’t co-occuring.
  - “Bryan is, in no way even remotely close to a billionaire”.

Cutoff

Table 1: Co-occurrence probabilities for target words *ice* and *steam* with selected context words from a 6 billion token corpus. Only in the ratio does noise from non-discriminative words like *water* and *fashion* cancel out, so that large values (much greater than 1) correlate well with properties specific to ice, and small values (much less than 1) correlate well with properties specific of steam.

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

$$P_{ik} = P(k \mid i) = X_{ik}/X_i$$

$$X_i = \sum_j X_{ij}$$

X_{ik} **is the co-occurence count between i and k**

Sum over all words in dictionary

Lets learn a function that…

- Captures the ratio of word probabilities:

$$F(w_i, w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}}$$

That ~ is meaningful, by the way. It means that the context word's vector is drawn from a different vector space (of the same dimensionality as the other two words vector)

- Encodes words as vectors in a vector space (or two):

$$e\,.\,g\,.\,w_x = [0{,}1, -3{,}4]$$

  - (The exact values in these vectors is part of what we'll learn)

- Makes addition and subtraction of word vectors meaningful

Defining a function to enforce additivity

- If we make F the exponential function (which is the inverse of taking the natural log), then we can do this…

$$F\left((w_i - w_j)^T \tilde{w}_k\right) = \frac{F(w_i^T \tilde{w}_k)}{F(w_j^T \tilde{w}_k)} = \frac{P_{ik}}{P_{jk}}$$

The difference between the word vectors relates to the ratio of their probabilities of co-occurence

Make inner products relate to probabilities

- Define making the inner product between a word and its context word relate to the LOG of the probability of seeing that word in the context.

We'll learn these word vectors

$$w_i^T \tilde{w}_K = log(P_{ik}) = log(\frac{X_{ik}}{X_i}) = log(X_{ik}) - log(X_i)$$

This is in the data

-

So what do we need to learn, exactly?

- Stated again…

$$w_i^T \tilde{w}_K = log(X_{ik}) - log(X_i)$$

- If we create a bias term related to each word i and k, they can stand in for log(X_i), resulting in this…

$$w_i^T \tilde{w}_k + b_i + \tilde{b}_k = \log(X_{ik}) \; . \qquad (7)$$

- We're going to learn everything to the left of the = in equation 7.

So what do we use as our word vectors?

- We end up learning 2 vectors for every word.
- Just sum them up and use those as the embeddings

$$E = W + \tilde{W}$$

The word analogy task

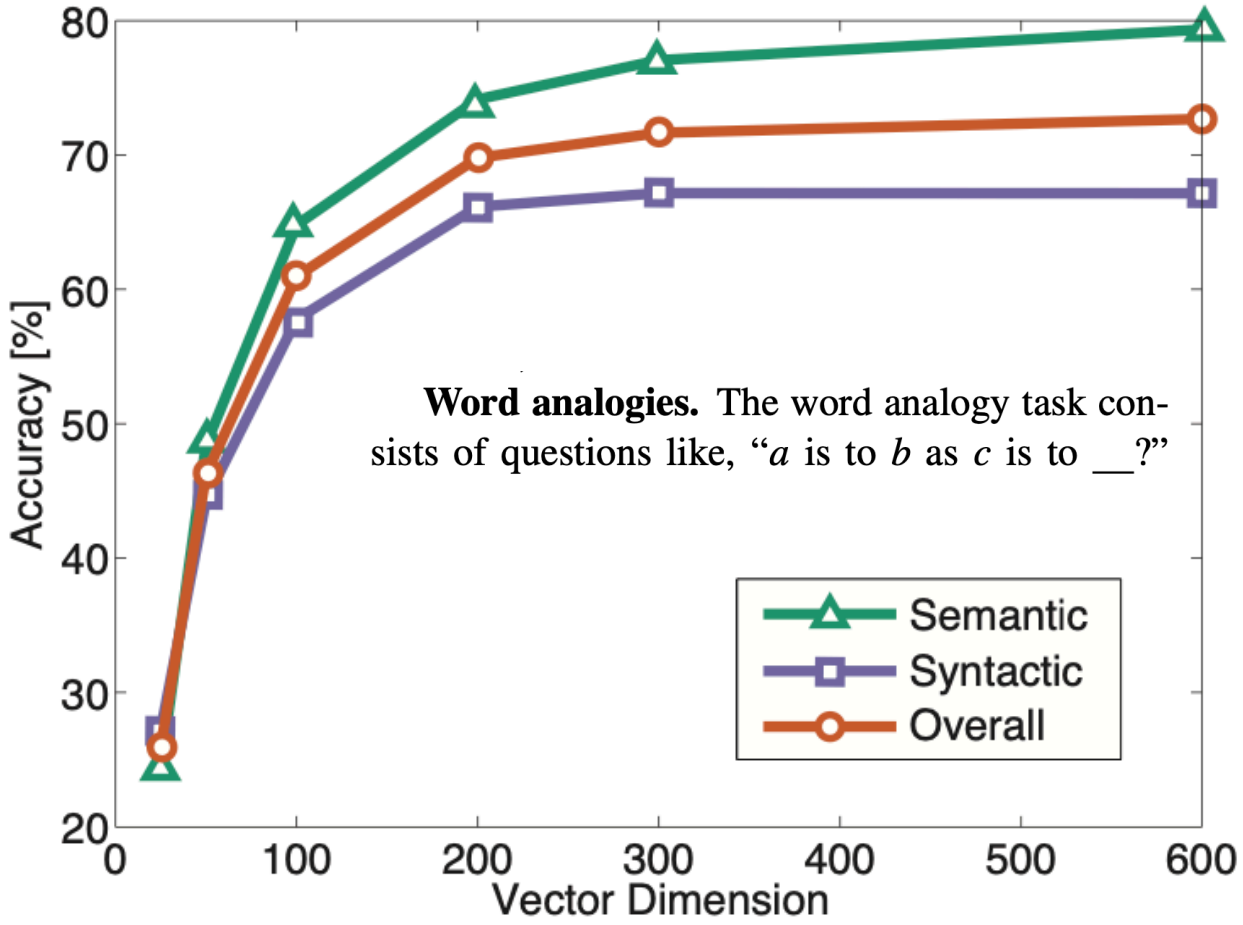
Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. "Efficient estimation of word representations in vector space." *arXiv preprint arXiv:1301.3781* (2013).

Table 1: *Examples of five types of semantic and nine types of syntactic questions in the Semantic-Syntactic Word Relationship test set.*

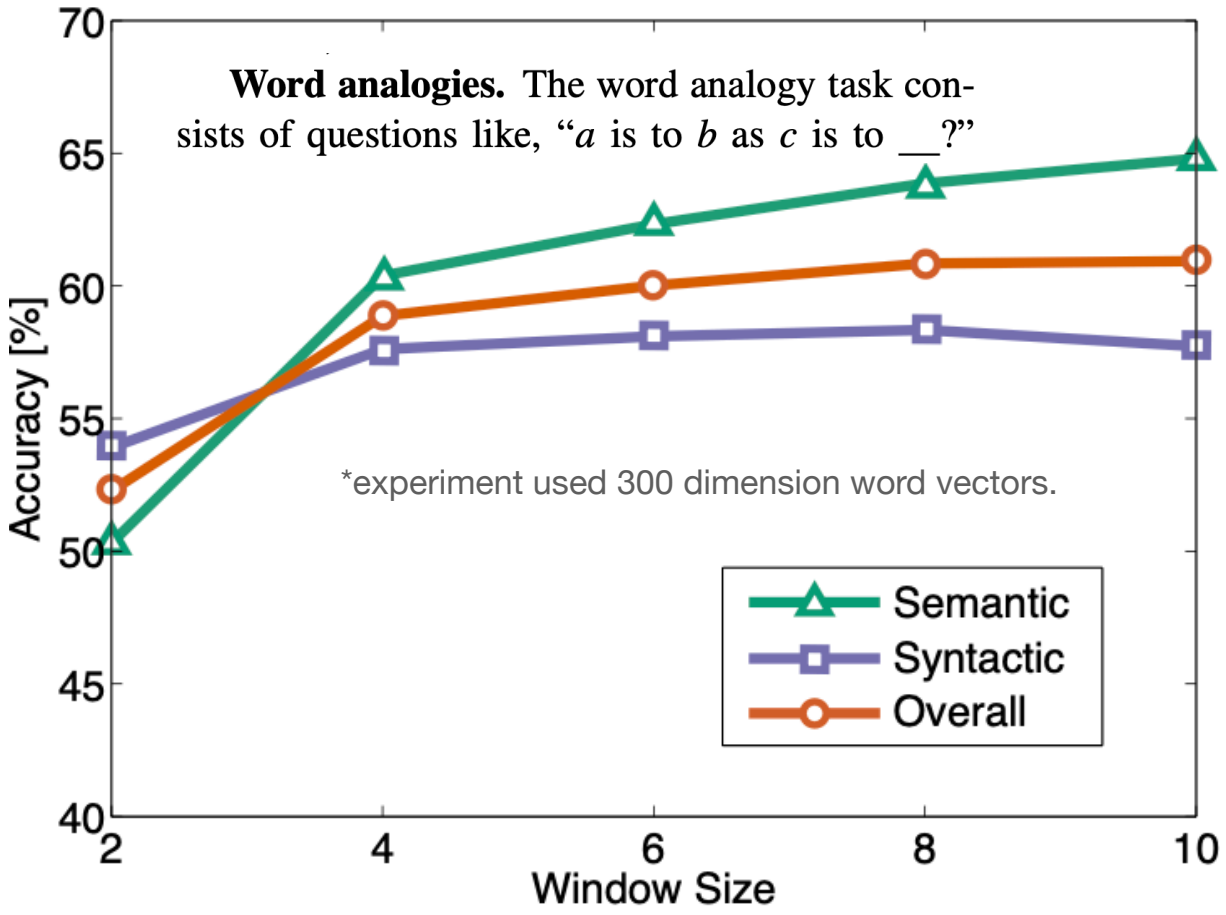
Type of relationship	Word Pair 1		Word Pair 2	
Common capital city	Athens	Greece	Oslo	Norway
All capital cities	Astana	Kazakhstan	Harare	Zimbabwe
Currency	Angola	kwanza	Iran	rial
City-in-state	Chicago	Illinois	Stockton	California
Man-Woman	brother	sister	grandson	granddaughter
Adjective to adverb	apparent	apparently	rapid	rapidly
Opposite	possibly	impossibly	ethical	unethical
Comparative	great	greater	tough	tougher
Superlative	easy	easiest	lucky	luckiest
Present Participle	think	thinking	read	reading
Nationality adjective	Switzerland	Swiss	Cambodia	Cambodian
Past tense	walking	walked	swimming	swam
Plural nouns	mouse	mice	dollar	dollars
Plural verbs	work	works	speak	speaks

We evaluate the overall accuracy for all question types, and for each question type separately (semantic, syntactic). Question is assumed to be correctly answered only if the closest word to the vector computed using the above method is exactly the same as the correct word in the question; synonyms are thus counted as mistakes. This also means that reaching 100% accuracy is likely to be impossible, as the current models do not have any input information about word morphology.

How big should we make embeddings?



What is the right context window size?



How does Glove compare to Word2Vec?

Model	Dim.	Size	Sem.	Syn.	Tot.
SG	300	1B	61	61	61
CBOW	300	1.6B	16.1	52.6	36.1
vLBL	300	1.5B	54.2	64.8	60.0
ivLBL	300	1.5B	65.2	63.0	64.0
GloVe	300	1.6B	80.8	61.5	70.3

Word analogies. The word analogy task consists of questions like, "*a* is to *b* as *c* is to __?" The dataset contains 19,544 such questions, di-

What do Transformers use?

NOT GLOVE!

NOT WORD2VEC!

They learn their own!