Corpus Experiments for Word Embeddings

Benjamin Wilson (with Adriaan Schakel)

Berlin ML Learning Group, July 28 2015

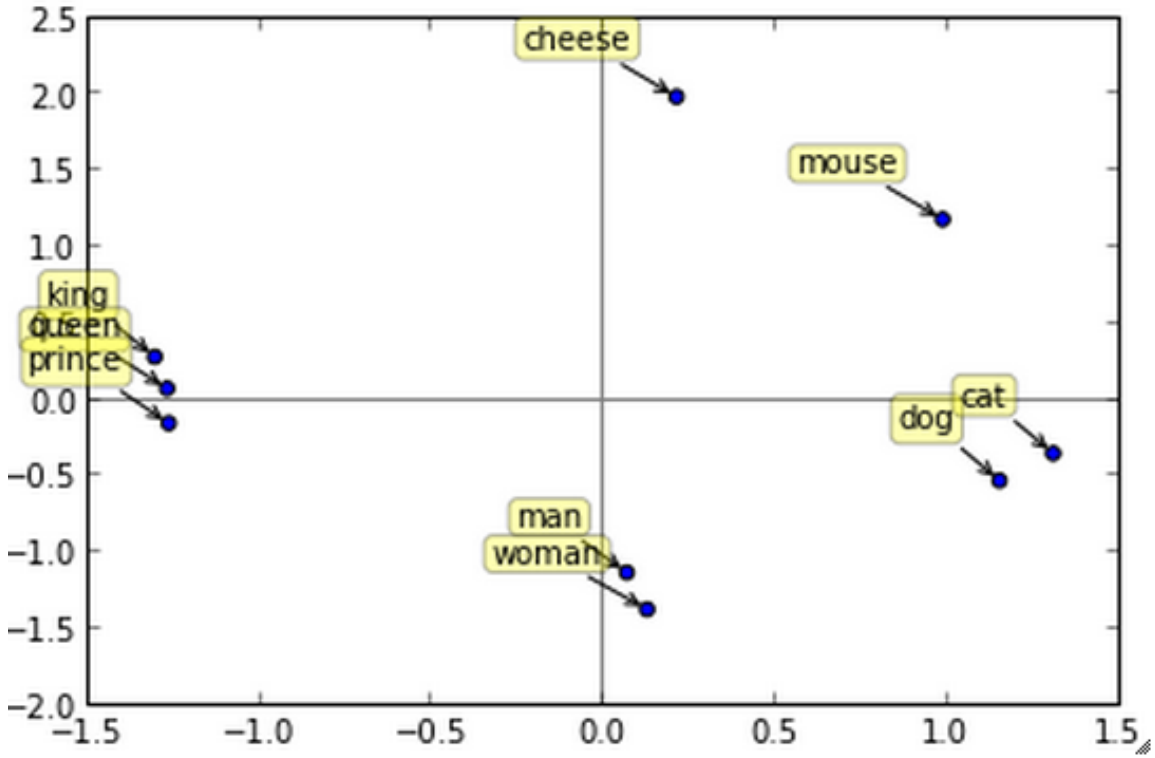
Outline

Word Embeddings

- associate words with points in space
- spatially encode word meaning and relationships between words
- learned from input texts
- e.g. word2vec, GloVe, ...

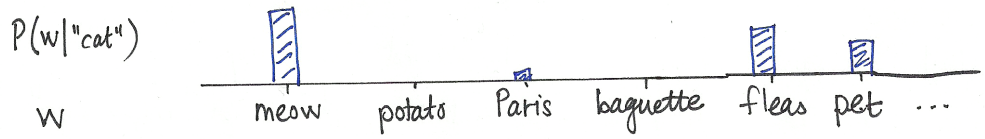
Pictorially

- spatial distance corresponds to word similarity
- words are close together $\Leftrightarrow$ their "meanings" are similar



Co-occurrence distributions

- Suppose we read the word `cat`. What is the probability $P(w|\texttt{cat})$ that we'll read the word w nearby?



- $P(\cdot|\texttt{cat})$ is the *co-occurrence distribution* of the word `cat`.
- Distributional Hypothesis: the meaning of `cat` is captured by the co-occurrence distribution.
- Word embeddings are trained by sampling from the co-occurrence distribution.

Some Questions about Word Embeddings

How would the word vector change if ...

1. ... the word were less frequent? more frequent?
2. ... the word were less informative? i.e. if $P(\cdot|\texttt{cat})$ were noisier? less noisy?

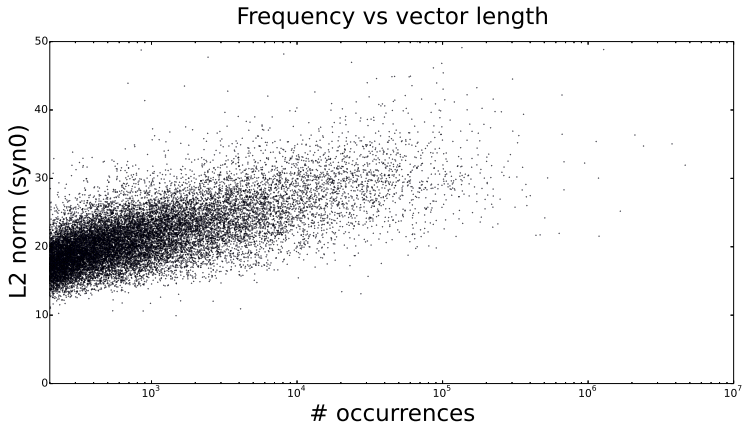
Different Approaches

We could attempt to answer such questions by either:

1. reasoning about the mechanism (**theoretical**).
2. studying the data (**observational**).
3. studying how the output changes when the input is varied in a controlled manner (**experimental**).

Observing word frequency

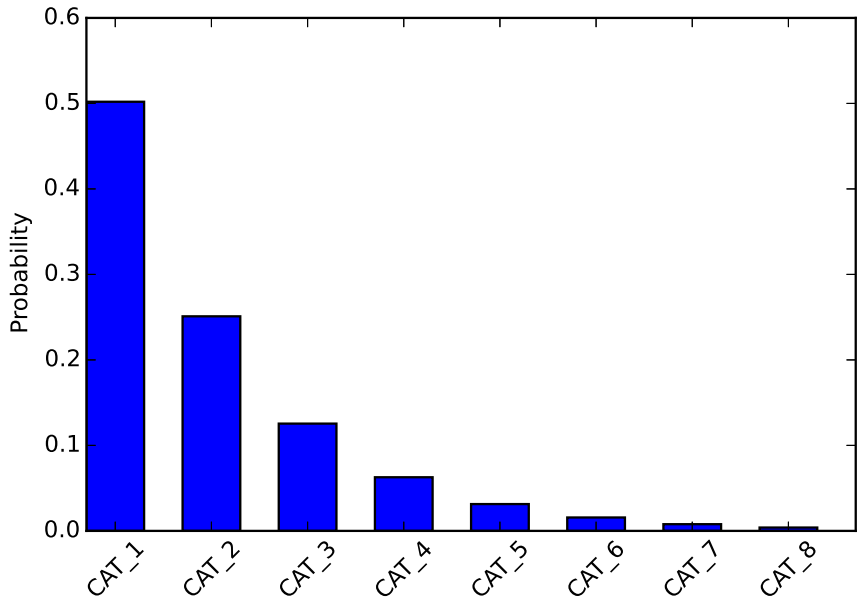
- word frequency is encoded in vector length



- but word frequency is related to many other things (e.g. significance)
- so difficult to determine **observationally** if vector length is related to word frequency **directly**
- for this, an experimental approach is required

Word Frequency Experiment

- Modify the training corpus, replacing all occurrences of the word `cat` with a token chosen from the distribution:

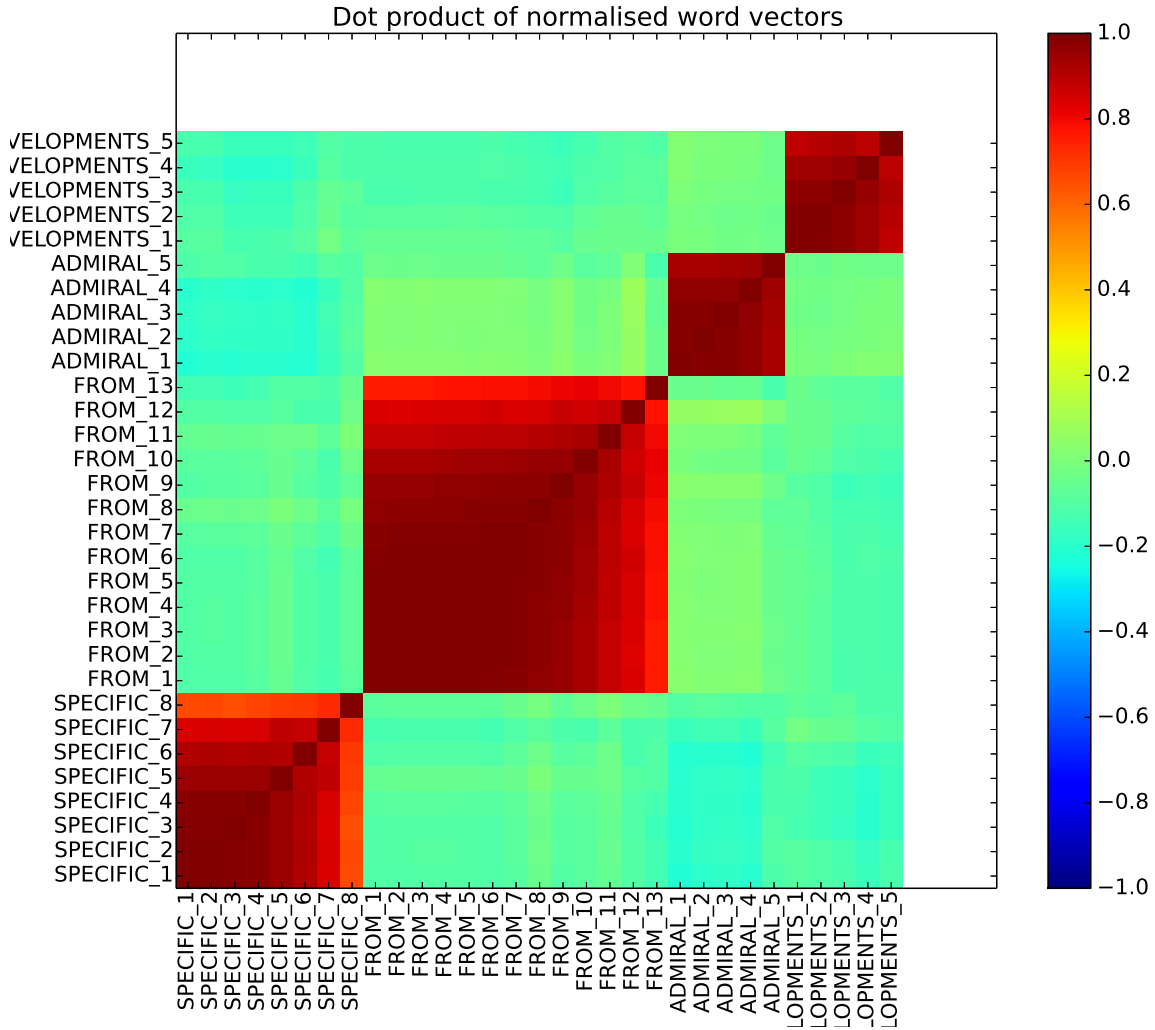


- Thus `cat` is replaced by a family of tokens with varying frequencies, but the same co-occurrence distribution as `cat`.
- Train the word embedding on the modified corpus
- Study the word vectors of `CAT_i` for $i = 1, 2, \ldots$.

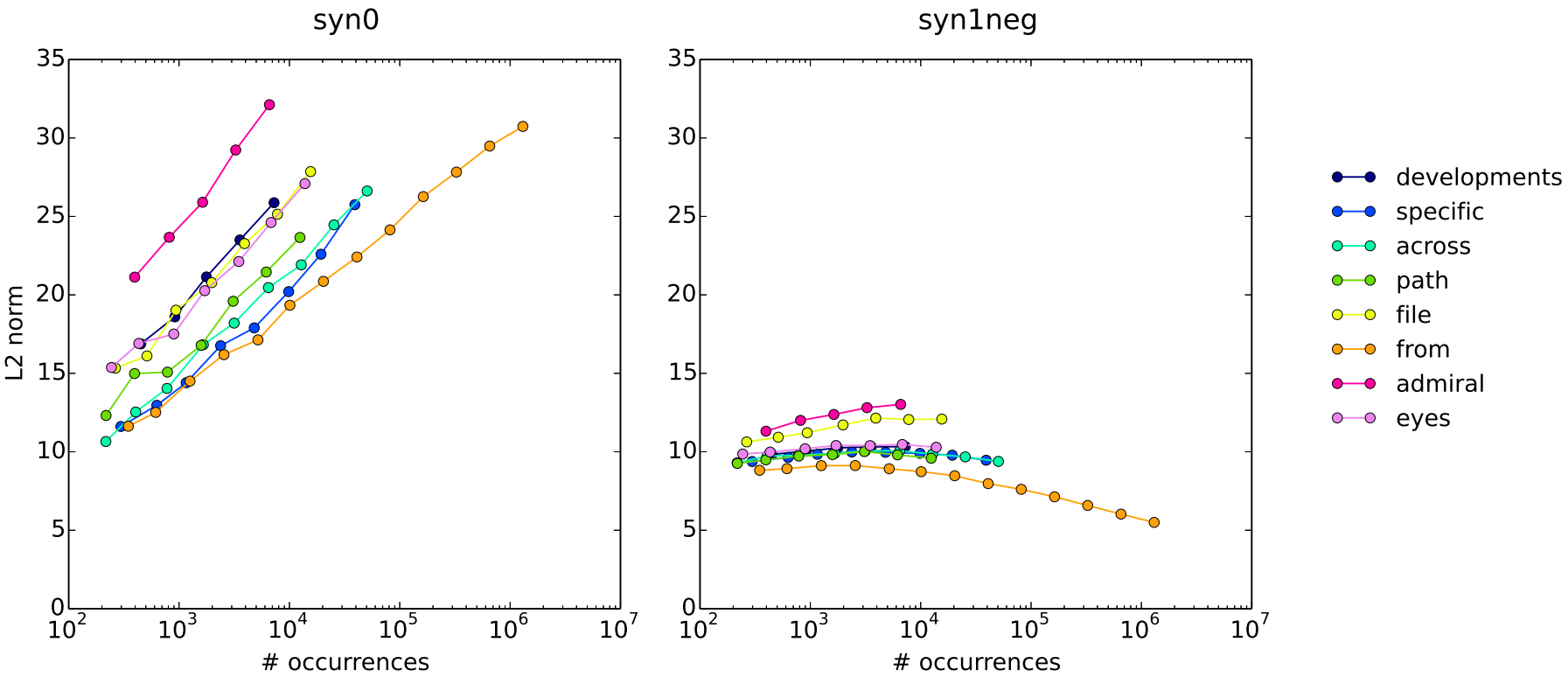
Word Frequency Experiment Example

a pedigreed CAT_2 is one whose ancestry is recorded by a CAT_5 fancier organization a purebred CAT_2 is one whose ancestry contains only individuals of the same breed the semiferal CAT_1 a mostly outdoor CAT_1 is not owned by any one individual the domestic CAT_1 was first classified as felis catus the domesticated CAT_1 and its closest wild ancestor are both diploid organisms that possess chromosomes and roughly genes ... dogs perform many roles for people such as hunting herding pulling loads protection assisting police and military companionship and more recently aiding handicapped individuals in carl linnaeus listed among the types of quadrupeds familiar to him the latin word for dog canis among the species within this genus linnaeus listed the red fox as canis vulpes wolves canis lupus and the domestic dog canis canis in later editions linnaeus dropped canis canis and greatly expanded his list of the canis genus of quadrupeds and by included alongside the

Word Frequency Experiment Results (word2vec)



Word Frequency Experiment Results (word2vec)



Word Frequency Experiment Results (word2vec)

- Vector direction is independent of frequency.
- Vector length depends directly and linearly on word frequency (for the word vectors, “syn0”).

Co-occurrence Noise Experiment

- Objective: vary the level of noise in the co-occurrence distribution, without varying the frequency.
- Our noise distribution is the global word frequency distibution.
- Thus we can introduce noise into the co-occurrence distribution of a word by randomly displacing some of its original occurrences throughout the corpus.

Co-occurrence Noise Experiment Recipe

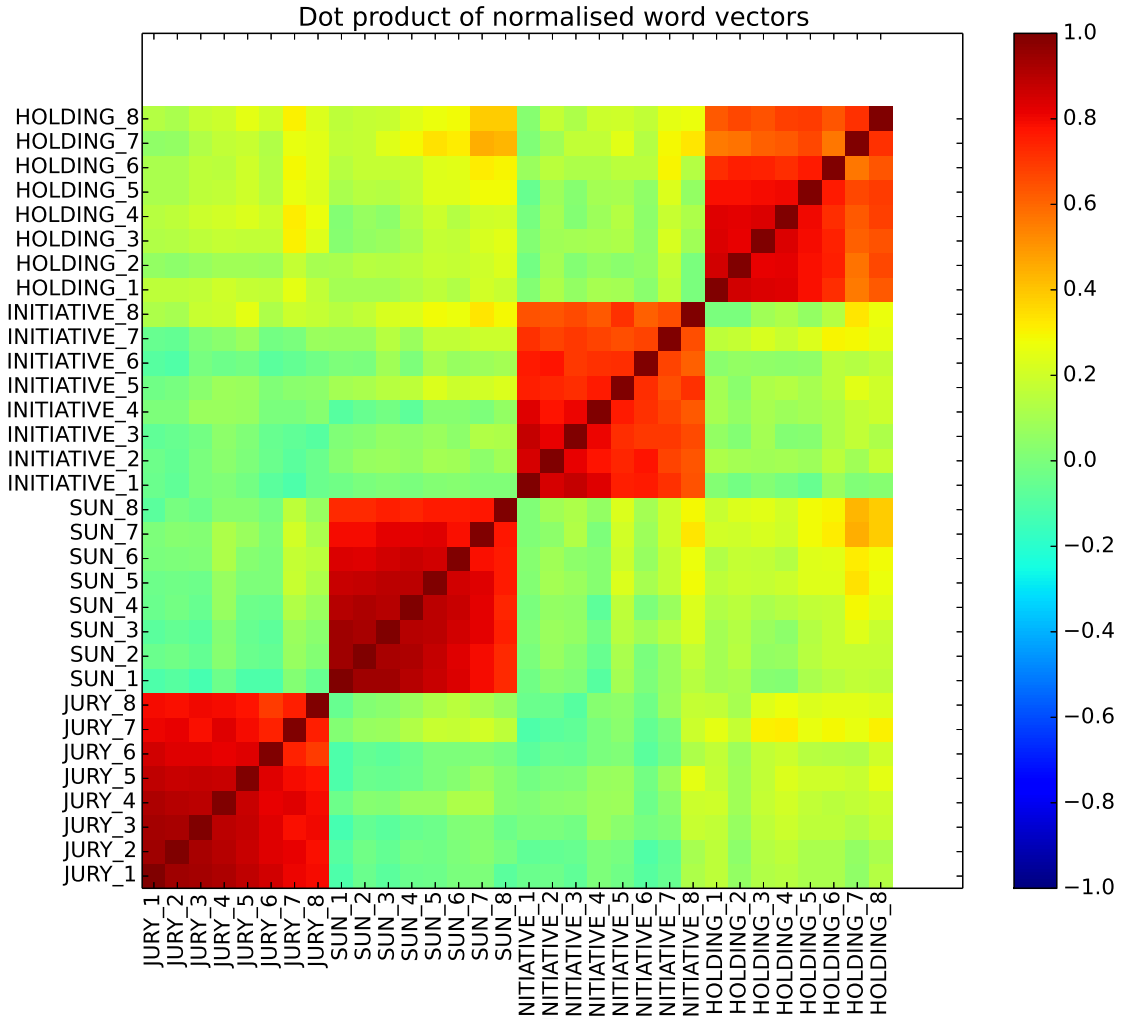
How, exactly?

1. Modify corpus as before, replacing the word `cat` by tokens `CAT_i` for $i = 1, 2, \ldots$ with varying frequencies.
2. Introduce random occurrences of `CAT_i` into the corpus, such that its frequency is raised to that of `cat`.

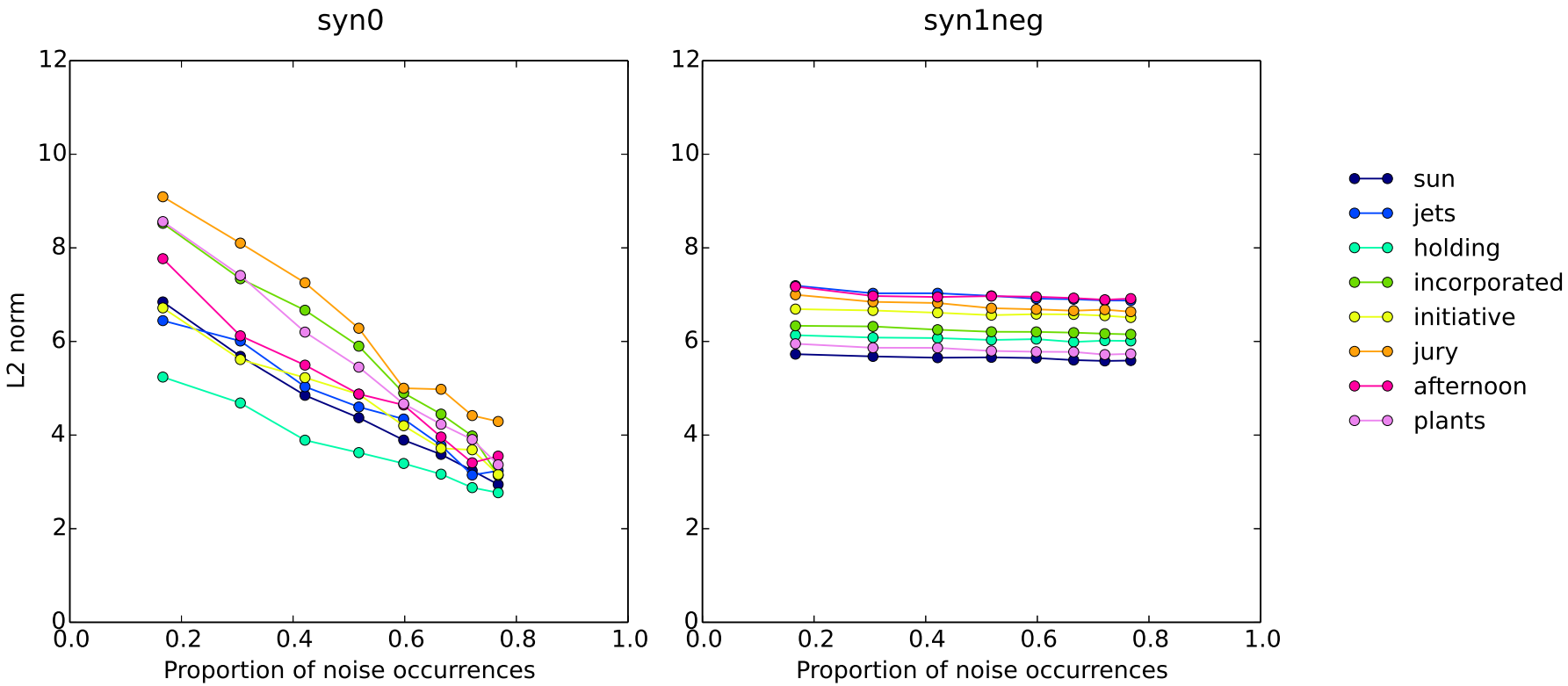
Co-occurrence Noise Experiment Example

a pedigreed CAT_1 is *CAT_3* one whose ancestry is recorded by a CAT_1 fancier organization a purebred CAT_3 is one whose ancestry contains only individuals of the same breed the semiferal CAT_2 a mostly outdoor CAT_2 is not owned by any one individual *CAT_2* the domestic CAT_3 was first classified as felis catus the domesticated CAT_1 and its closest wild ancestor are both diploid organisms *CAT_3* that possess chromosomes and roughly genes ... the domestic dog canis lupus familiaris or canis familiaris is a domesticated canid which has been selectively bred for millennia for various behaviors sensory capabilities and physical attributes although initially thought to have originated as a manmade variant of an extant canid species variously supposed as being the *CAT_1* dhole golden jackal or gray wolf *CAT_3* extensive genetic studies undertaken during the s indicate that dogs *CAT_2* diverged from a now-extinct canid in eurasia

Co-occurrence Noise Experiment Results (word2vec)



Co-occurrence Noise Experiment Results (word2vec)

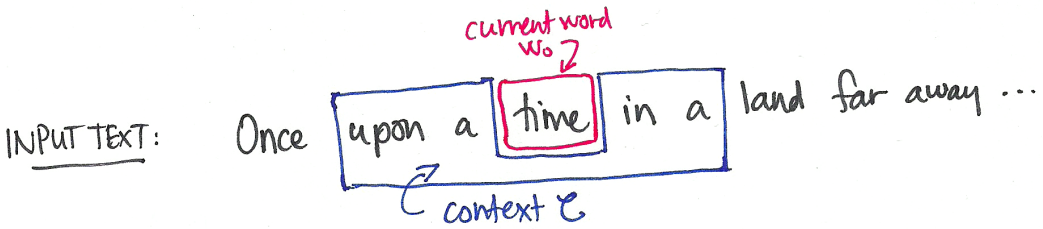


Co-occurrence Noise Experiment Results (word2vec)

- Vector direction is remarkably insensitive to co-occurrence noise.
- Vector length depends directly, linearly, on the level of co-occurrence noise.
- This suggests that, within any frequency band, vector length can be taken as a measure of signal strength.

Learning from text

- word2vec learns from input text
- considers each word w_0 in turn, along with its context $\mathcal{C}$
- context = neighbouring words (here, for simplicity, 2 words forward and back)



sample #	w_0	context $\mathcal{C}$
1	once	{upon, a}
	...	
4	time	{upon, a, in, a}
	...	

Two approaches: CBOW and Skip-gram

word2vec can learn the word vectors via two distinct learning tasks, CBOW and Skip-gram.

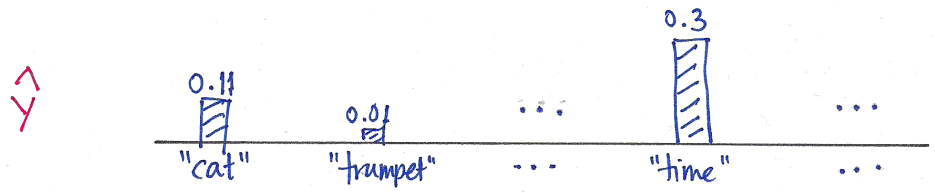
- CBOW: predict the current word w_0 given only $\mathcal{C}$
- Skip-gram: predict words from $\mathcal{C}$ given w_0
- Skip-gram produces better word vectors for infrequent words
- CBOW is faster by a factor of window size – more appropriate for larger corpora
- We will speak only of CBOW (life is short).

CBOW learning task

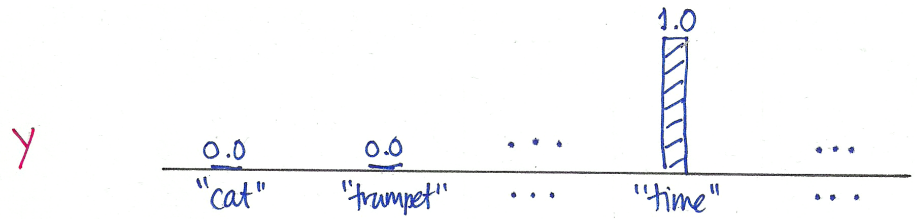
- Given only the current context $\mathcal{C}$, e.g.

$$\mathcal{C} = \{\texttt{upon}, \texttt{a}, \texttt{in}, \texttt{a}\}$$

  predict which of all possible words is the current word w_0, e.g. $w_0 = \texttt{time}$.
- multiclass classification on the vocabulary W
- output is $\hat{y} = \hat{y}(\mathcal{C}) = P(\cdot|\mathcal{C})$ is a probability distribution on W, e.g.



- train so that $\hat{y}$ approximates target distribution y – "one-hot" on the current word, e.g.

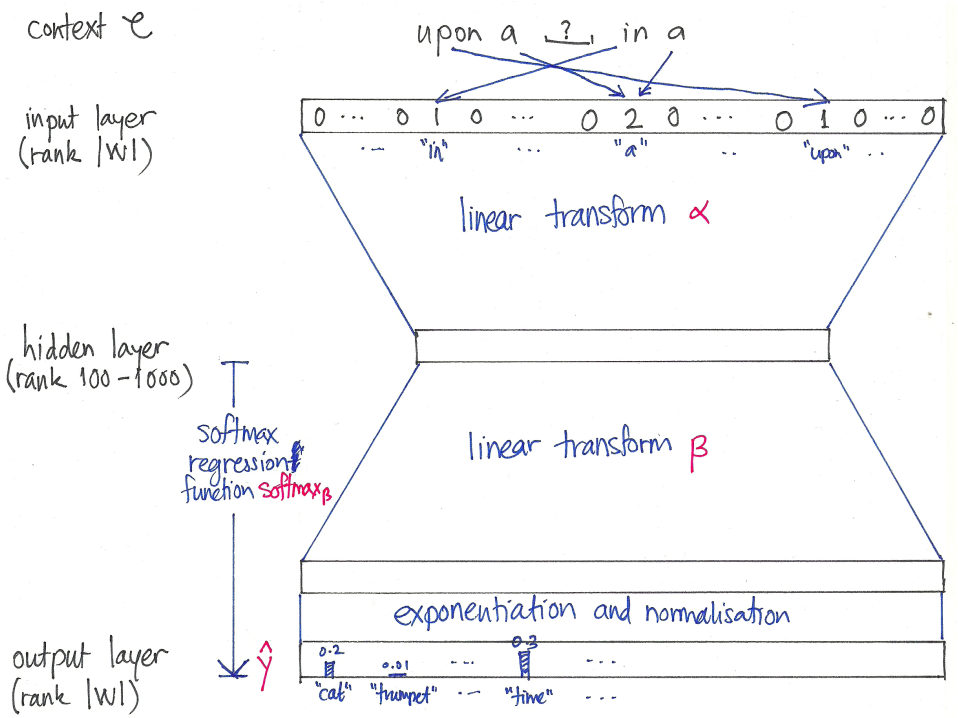


Training CBOW with softmax regression

Model:

$$\hat{y} = P(\cdot|\mathcal{C}; \alpha, \beta) = \text{softmax}_\beta(\sum_{w \in \mathcal{C}} \alpha_w),$$

where α, β are families of parameter vectors. Pictorially:



In practice, hierarchical softmax or negative sampling is used in place of softmax.

Stochastic Gradient Descent

- learn the model parameters (here, the linear transforms)
- minimize the difference between output distribution $\hat{y}$ and target distribution y, measured using the cross-entropy H:

$$H(y, \hat{y}) = -\sum_{w \in W} y_w \log \hat{y}_w$$

- given y is one-hot, same as maximizing the probability of the correct outcome

$$\hat{y}_{w_0} = P(w_0 | \mathcal{C}; \alpha, \beta).$$

- use stochastic gradient descent: for each (current word, context) pair, update all the parameters once.

Word Representation

Post-training, associate every word $w \in W$ with a vector $\mathbf{v}[w]$:

- $\mathbf{v}[w]$ is the vector of synaptic strengthes connecting the input layer unit w to the hidden layer
- more meaningfully, $\mathbf{v}[w]$ is the hidden-layer representation of the single-word context $\mathcal{C} = \{w\}$.
- vectors are (artifically) normed to unit length (Euclidean norm), post-training.