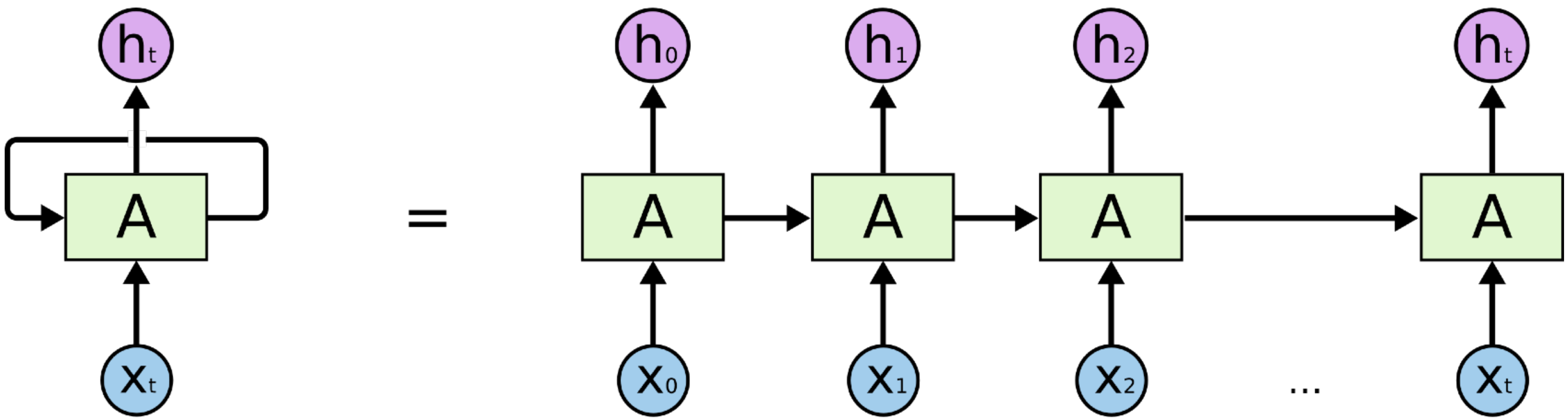
Machine learning for sequences
Text and time series

Romain Tavenard (Université de Rennes 2)

Machine learning for structured data (continued)

- Two options
  1. Cast the data to tabular
     - Representation based on global features (*eg.* bag of words for text or images)
  2. Use structural information in the model
     - images: 2d convolutions
     - sequences: recurrent models, 1d convolutions, temporal kernels

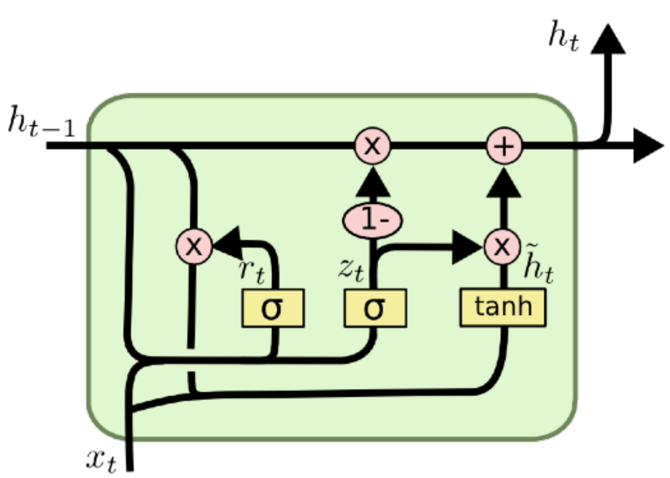
NN with recurrent units



Source: Christopher Olah's blog

NN with recurrent units

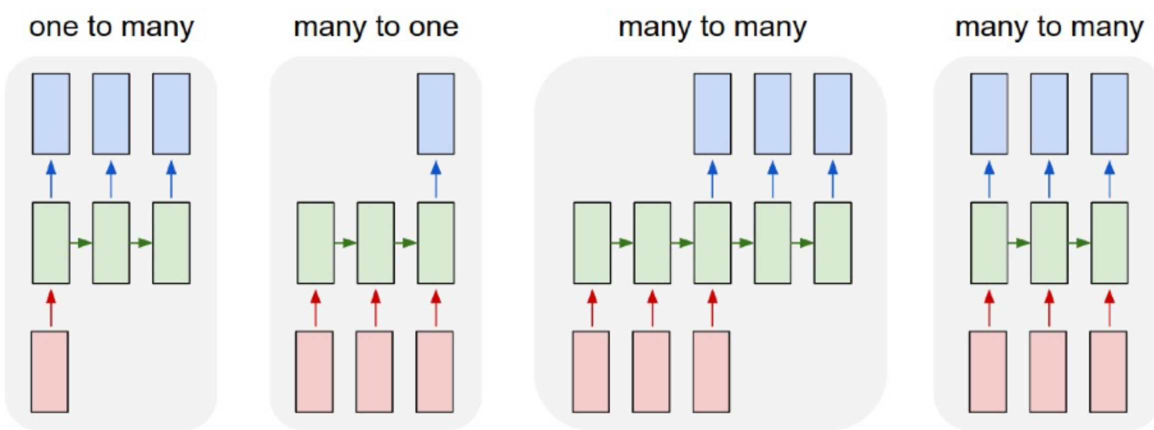
- Variants that work well in practice
  - Long Short Term Memory (LSTM)
  - Gated Recurrent Unit (GRU)
- Principle
  - At each time step, keep only part of the information



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Illustration: GRU cell, source: Christopher Olah’s blog

NN with recurrent units




```
PANDARUS:
Alas, I think he shall be come approached and the day
When little srain would be attain'd into being never fed,
And who is but a chain and subjects of his death,
I should not sleep.

Second Senator:
They are away this miseries, produced upon my soul,
Breaking and strongly should be buried, when I perish
The earth and thoughts of many states.

DUKE VINCENTIO:
Well, your wit is in the care of side and that.

Second Lord:
They would be ruled after this chamber, and
my fair nues begun out of the fact, to be conveyed,
Whose noble souls I'll have the heart of the wars.
```

Sample text generated by a RNN trained on Shakespeare words

For $\bigoplus_{n=1,\ldots,m}$ where $\mathcal{L}_{m_\bullet} = 0$, hence we can find a closed subset $\mathcal{H}$ in $\mathcal{H}$ and any sets $\mathcal{F}$ on X, U is a closed immersion of S, then $U \to T$ is a separated algebraic space.

Proof. Proof of (1). It also start we get

$$S = \mathrm{Spec}(R) = U \times_X U \times_X U$$

and the comparicoly in the fibre product covering we have to prove the lemma generated by $\coprod Z \times_U U \to V$. Consider the maps M along the set of points Sch_{fppf} and $U \to U$ is the fibre category of S in U in Section, ?? and the fact that any U affine, see Morphisms, Lemma ??. Hence we obtain a scheme S and any open subset $W \subset U$ in $Sh(G)$ such that $\mathrm{Spec}(R') \to S$ is smooth or an

$$U = \bigcup U_i \times_{S_i} U_i$$

which has a nonzero morphism we may assume that f_i is of finite presentation over S. We claim that $\mathcal{O}_{X,x}$ is a scheme where $x, x', s'' \in S'$ such that $\mathcal{O}_{X,x'} \to \mathcal{O}'_{X',x'}$ is separated. By Algebra, Lemma ?? we can define a map of complexes $\mathrm{GL}_{S'}(x'/S'')$ and we win. □

To prove study we see that $\mathcal{F}|_U$ is a covering of $\mathcal{X}'$, and $\mathcal{T}_i$ is an object of $\mathcal{F}_{X/S}$ for $i > 0$ and $\mathcal{F}_p$ exists and let $\mathcal{F}_i$ be a presheaf of $\mathcal{O}_X$-modules on $\mathcal{C}$ as a $\mathcal{F}$-module. In particular $\mathcal{F} = U/\mathcal{F}$ we have to show that

$$\widetilde{M}^\bullet = \mathcal{I}^\bullet \otimes_{\mathrm{Spec}(k)} \mathcal{O}_{S,s} - i_X^{-1}\mathcal{F})$$

is a unique morphism of algebraic stacks. Note that

$$\mathrm{Arrows} = (Sch/S)^{opp}_{fppf}, (Sch/S)_{fppf}$$

and

$$V = \Gamma(S, \mathcal{O}) \longmapsto (U, \mathrm{Spec}(A))$$

is an open subset of X. Thus U is affine. This is a continuous map of X is the inverse, the groupoid scheme S.

Proof. See discussion of sheaves of sets. □

The result for prove any open covering follows from the less of Example ??. It may replace S by $X_{spaces,étale}$ which gives an open subspace of X and T equal to S_{Zar}, see Descent, Lemma ??. Namely, by Lemma ?? we see that R is geometrically regular over S.

Sample LaTeX generated by a RNN trained on a book of algebraic geometry

Source: Andrej Karpathy's blog, http://karpathy.github.io/2015/05/21/rnn-effectiveness/

NN with recurrent units

- Efficient at modelling sequential dependencies
- Long-term dependencies: use LSTM or GRU
- Recent alternative (not covered here):
  Transformer modules
  - Assign importance weights to all items in a sequence

Dealing with text in practice

- Raw text data is challenging to handle
  - typos
  - what is a term?
  - lots of variants for a term
    - verb conjugation
    - plural form
    - *etc.*
  - synonyms

Dealing with text in practice

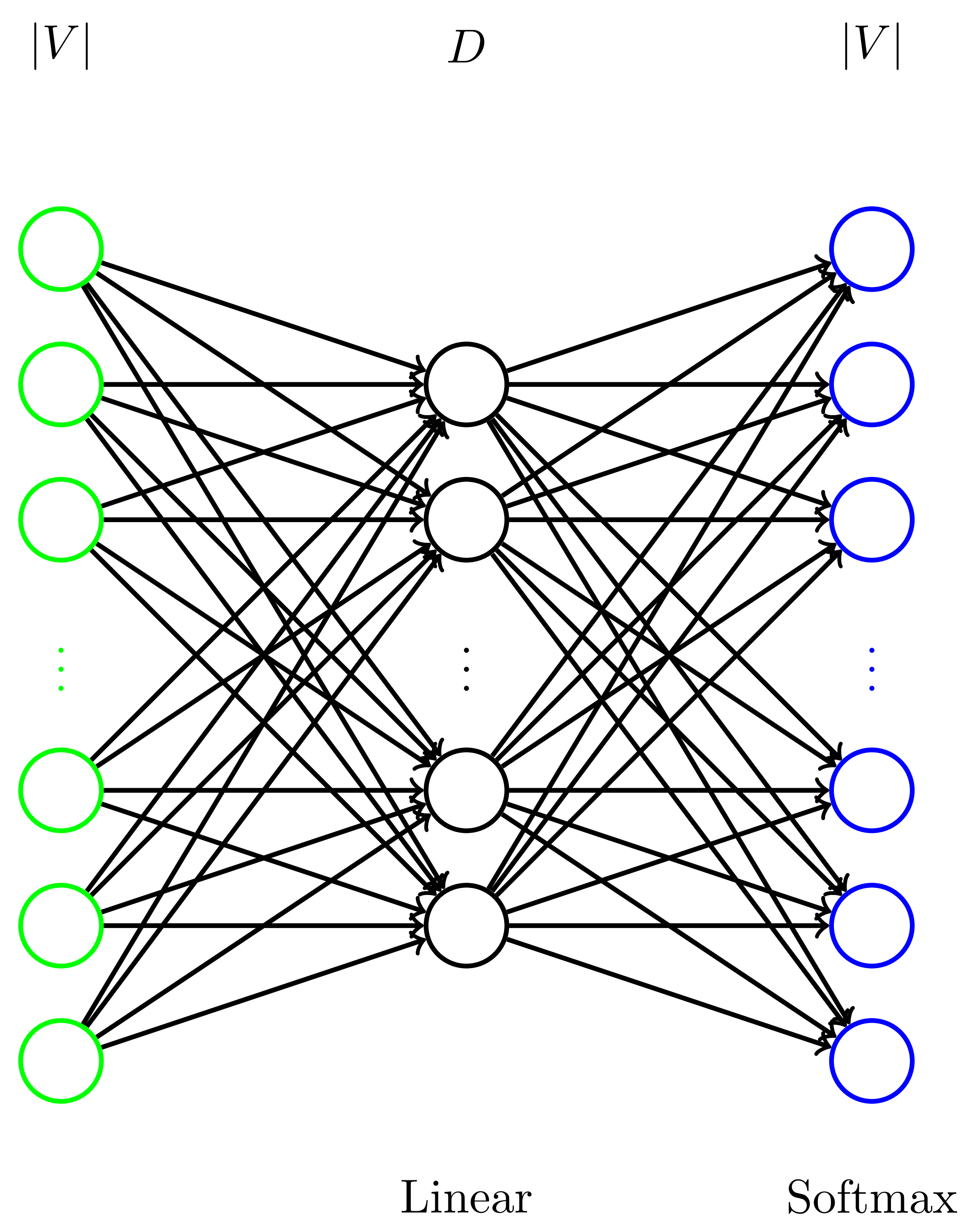
- Raw text data is challenging to handle
  - typos preprocessing
  - what is a term? tokenization
  - lots of variants for a term stemming
    - verb conjugation
    - plural form
    - *etc.*
  - synonyms word embeddings

Word embeddings

- Basic idea
  - 1 term = 1 point in multidimensional space
  - Goal: define a space such that similar terms are close

- Reference embedding
  - word2vec

word2vec



word2vec: Continuous Bag of Words (CBOW)

Le **chien mange un os dans sa gamelle**.

- **Input**
  - A bag-of-word representation (binary encoding) of the target word's neighborhood
- Classification task : predict the **target** (middle word)

- Generating a training sample
  1. Sample a word at random in a text
  2. Provide its fixed-length neighborhood

- Why CBOW ?
  - Hidden layer is a Continuous representation of the input Bag of Word

word2vec: skip-gram

Le **chien** mange un **os** dans sa gamelle.

- **Input**
  - A word
- Classification task: predict **a neighborhood word**

- Generating a training sample
  1. Sample a word at random in a text
  2. Sample a word from its neighbourhood at random

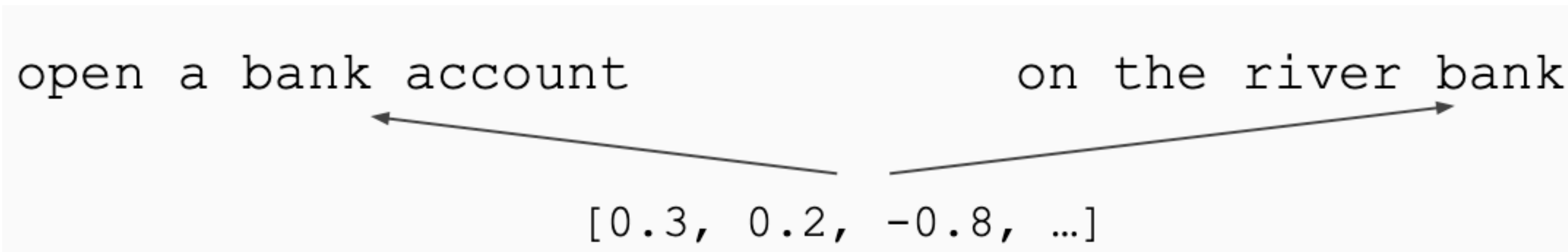
- Why skip-gram ?
  - Associate word pairs (like in bi-gram)
  - Allow skips

More about word2vec

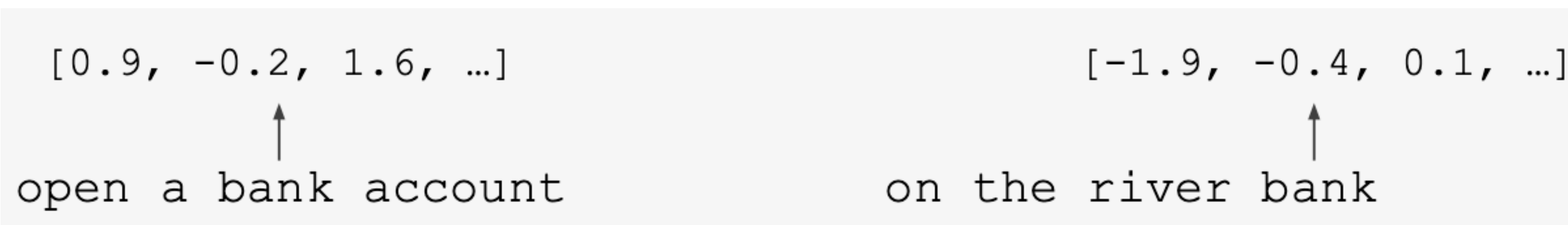
- According to authors
  - Skip-gram
    - works well with small amount of the training data
    - represents well even rare words or phrases
  - CBOW
    - several times faster to train than the skip-gram
    - slightly better accuracy for the frequent words

Limitations

- Word embeddings are applied in a context free manner

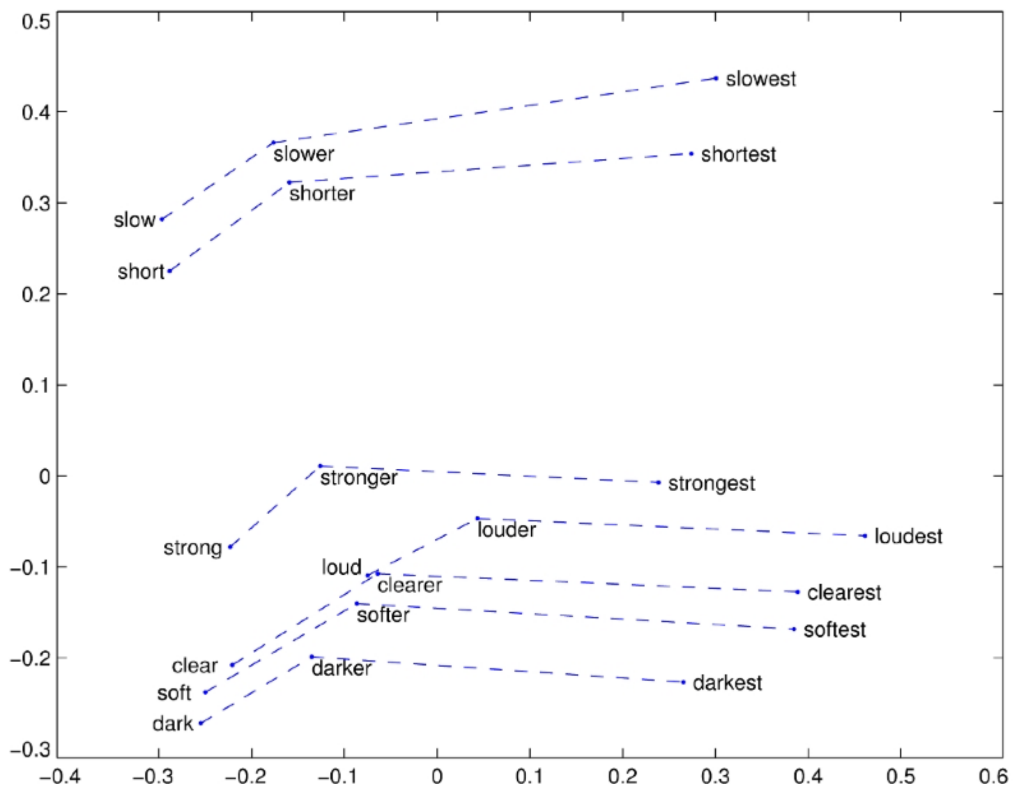


- Solution: Train contextual representations on text corpus

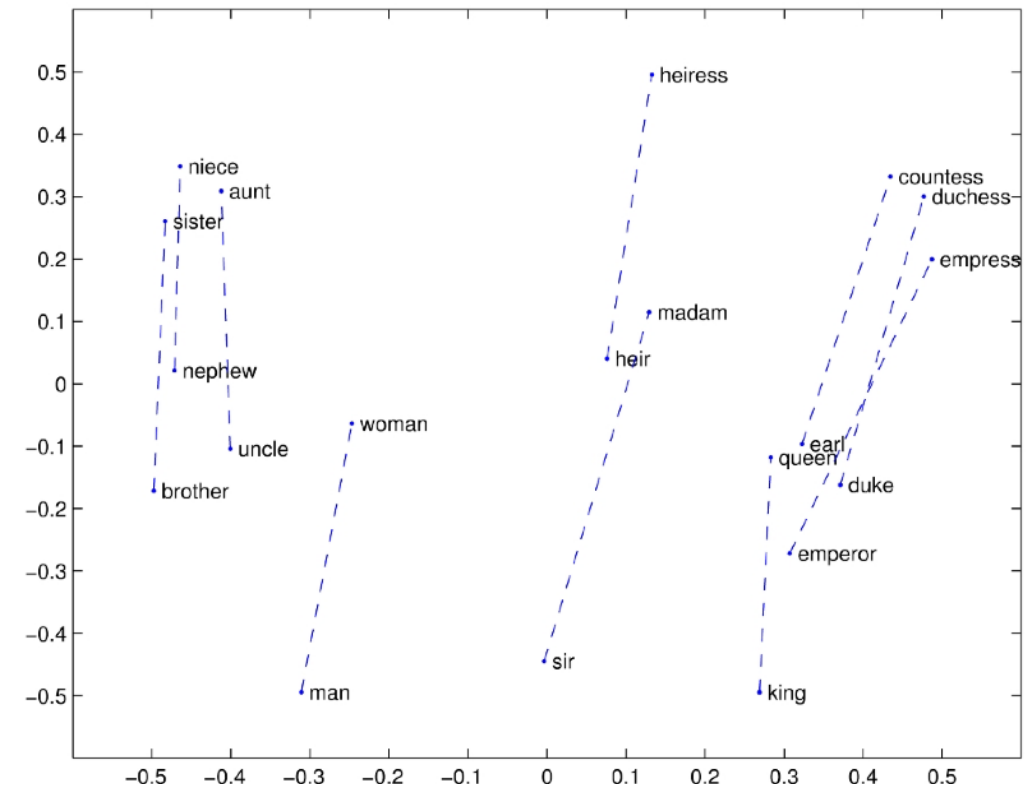


  - Example: BERT embeddings (based on Transformers)

Embedding visualisation



Source: Stanford NLP



- 2d-3d projections (PCA)
  - https://projector.tensorflow.org