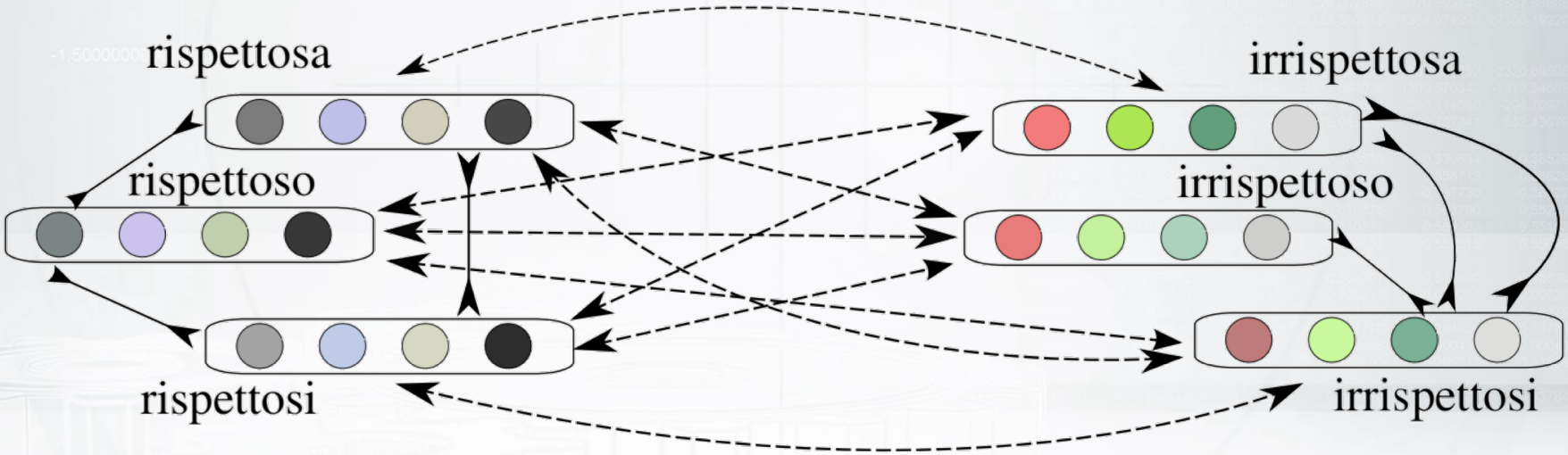
Why words? From character to sentence embeddings

Morphology can help

Problems:

- Languages with rich morphology
- Low frequency words, OOV words



Use morphology to improve word embeddings

I. Vulic et. al. Morph-fitting: Fine-Tuning Word Vector Spaces with Simple Language-Specific Rules, 2017.

FastText

Represent a word as a bag of character n-grams, e.g. for n = 3:

G_{where} : ***_wh, whe, her, ere, re_, _where_***

Model a word vector as a sum of sub-word vectors:

SGNS:

$$sim(u, v) = \langle \phi_u, \theta_v \rangle$$

FastText:

$$sim(u, v) = \sum_{g \in G_v} \langle \phi_u, \theta_g \rangle$$

Code and pre-trained embeddings: https://fasttext.cc/

P. Bojanowsky et al. Enriching Word Vectors with Subword Information, 2016.

Sent2vec

First ideas:

- Average pre-trained word vectors *(word2vec, GloVe, etc)*.
- Maybe use TF-IDF weights for averaging.

Sent2vec:

- Learn sentence embedding as a sum of sub-sentence units:

$$sim(u, s) = \frac{1}{|G_s|} \sum_{g \in G_s} \langle \phi_u, \theta_g \rangle$$

where G_s is a set of word n-grams for the sentence *s*.

Code and embeddings: https://github.com/epfml/sent2vec

Pagliardini et. al. Unsupervised Learning of Sentence Embeddings using Compositional n-Gram Features, 2017.

StarSpace

General framework:

entities (e.g. sentences) and *features* (e.g. words)

Lot's of applications:

- Text classification, e.g. sentiment analysis
- Ranking, e.g. ranking web documents given a query
- Collaborative filtering-based recommendation
- Content-based recommendation
- Embedding graphs, e.g. Freebase
- Learning word, sentence or document embeddings

Code and tutorials: github.com/facebookresearch/Starspace

Wu et. al. StarSpace: Embed All The Things! 2017

StarSpace

Mode 3 (sentence embeddings):

Use case: learn pairwise similarity from collections of similar objects, e.g. sentence similarity.

Data format: each line is a collection of similar sentences:

```
sent1_word1 sent1_word2 … <tab> sent2_word1 sent2_word2 …
```

Training:

- Each sentence is represented as a bag of features (words or n-grams). They are embedded to predict sentence similarity
- Similar sentence pairs are taken from the collections
- Dissimilar sentence pairs are sampled at random

Deep learning?

The most popular options:

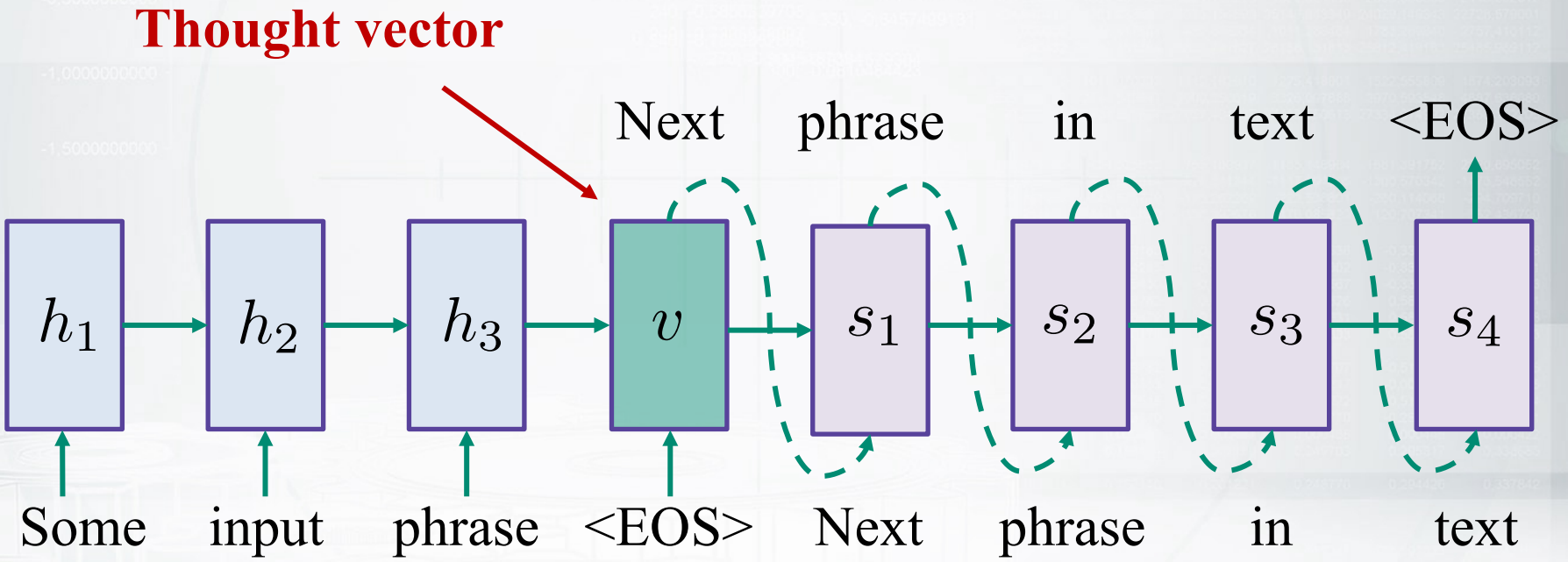
- Recurrent Neural Networks (sequence modelling)
- Convolutional Neural Networks (much faster)
- Recursive Neural Networks (use hierarchical structure)

Linguistic structure is back:

- Morphology can help to build word embeddings
- Recursive Neural Networks, Tree-LSTMs, DAG-LSTMs, etc. use syntax, span annotations, co-reference links…

Skip-thought vectors

- Predict next and previous sentences in text
- RNN encoder-decoder architecture



Kiros et. al. Skip-Thought Vectors, 2015, https://github.com/ryankiros/skip-thoughts.