Pulsar AI

How represent word as vector?

Pulsar AI

One-hot-encoding

“I love cake hate pizza”

I	1	0	0	0	0
love	0	1	0	0	0
cake	0	0	1	0	0
hate	0	0	0	1	0
pizza	0	0	0	0	1

Pulsar AI

Problems of one-hot-encoding

1. One-hot vectors are high-dimensional and sparse
2. Feature vector grows with the vocabulary size
3. Semantic and syntactic information are lost

Pulsar AI

WORD2VEC by Google

Efficient Estimation of Word Representations in Vector Space

by Mikolov, Corrado, Dean, Chen

2013 NAACL

https://arxiv.org/pdf/1301.3781.pdf

"You shall know a word
by the company it keeps"

One of the most successful ideas of modern statistical NLP

government debt problems turning into	banking	crises as has happened in
saying that Europe needs unified	banking	regulation to replace the hodgepodge

↖ These words will represent *banking* ↗

Harvard

Similar words

Word	Cosine distance
yale	0.638970
cambridge	0.612665
university	0.597709
faculty	0.588422
harvey_mudd	0.578338
johns_hopkins	0.575645
graduate	0.570294
undergraduate	0.565881
professor	0.563657
mcgill	0.562168
ph_d	0.558665
california_berkeley	0.555539
yale_university	0.550480
harvard_crimson	0.549848
princeton	0.544070

(WATER - WET) + FIRE = FLAMES

(PARIS - FRANCE) + ITALY = ROME

(WINTER - COLD) + SUMMER = WARM

(MINOTAUR - MAZE) + DRAGON = SIMCITY

Pulsar AI

word2vec. Visualization

Machine Translation

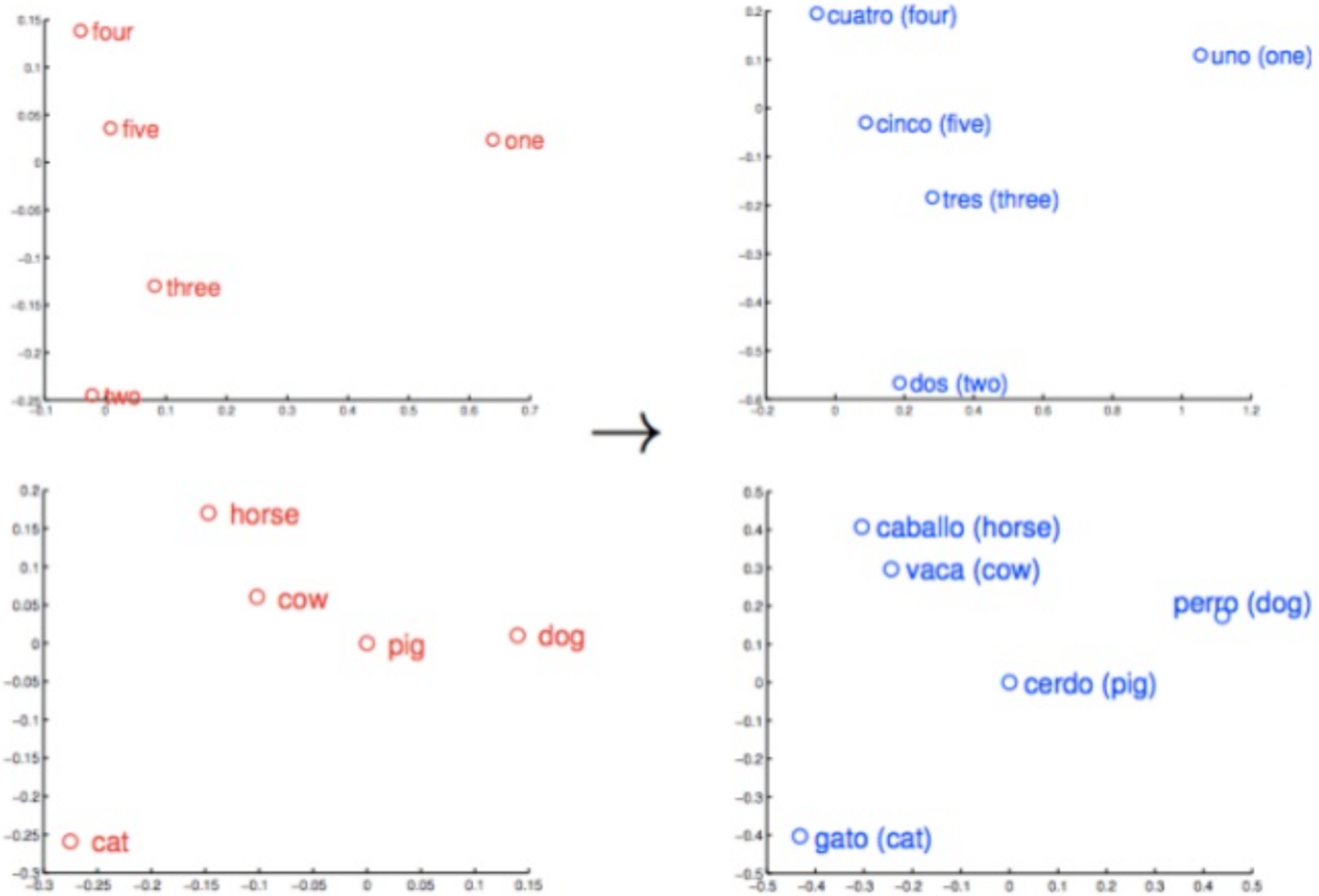
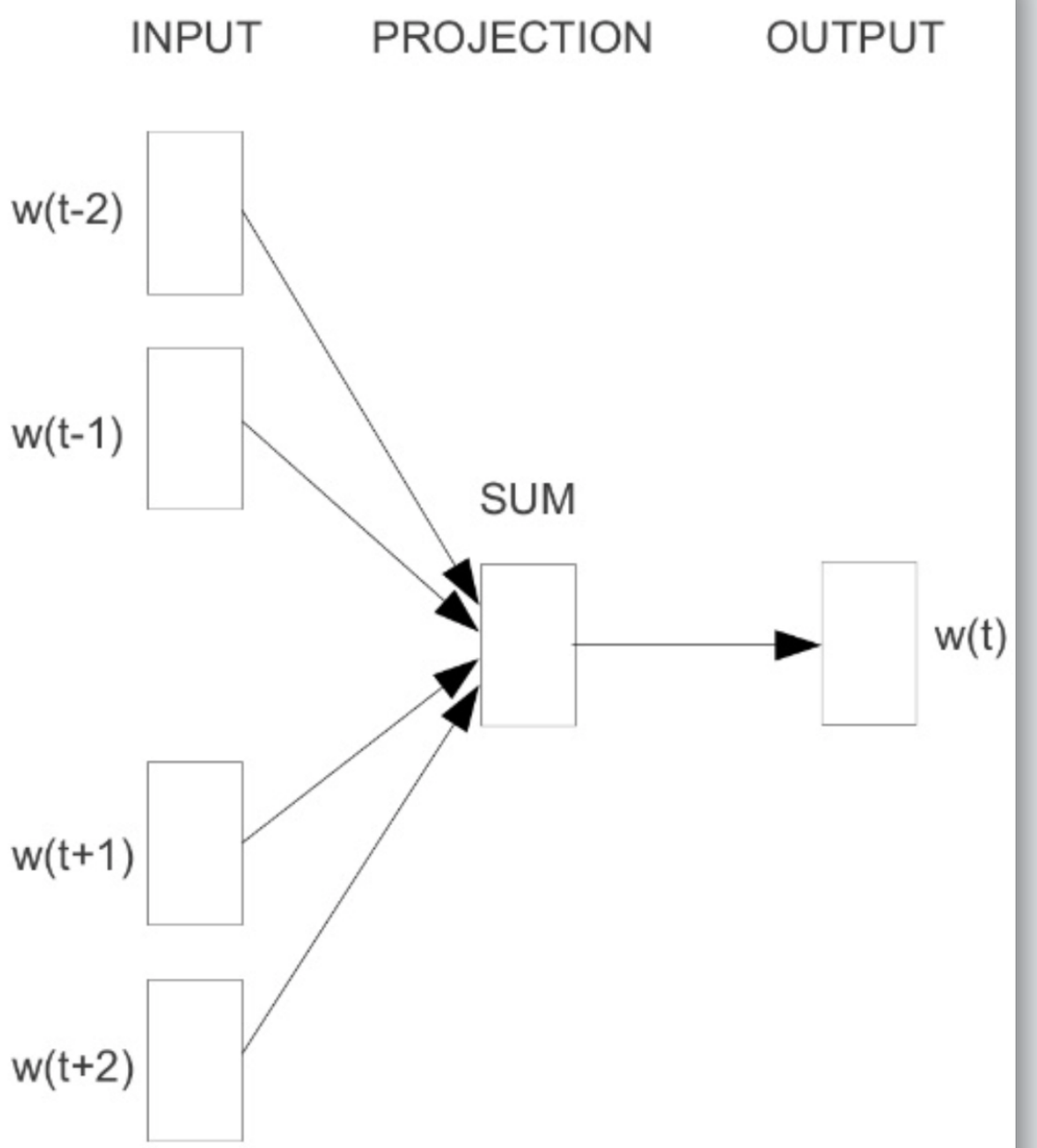


Figure 1: Distributed word vector representations of numbers and animals in English (left) and Spanish (right). The five vectors in each language were projected down to two dimensions using PCA, and then manually rotated to accentuate their similarity. It can be seen that these concepts have similar geometric arrangements in both spaces, suggesting that it is possible to learn an accurate linear mapping from one space to another. This is the key idea behind our method of translation.

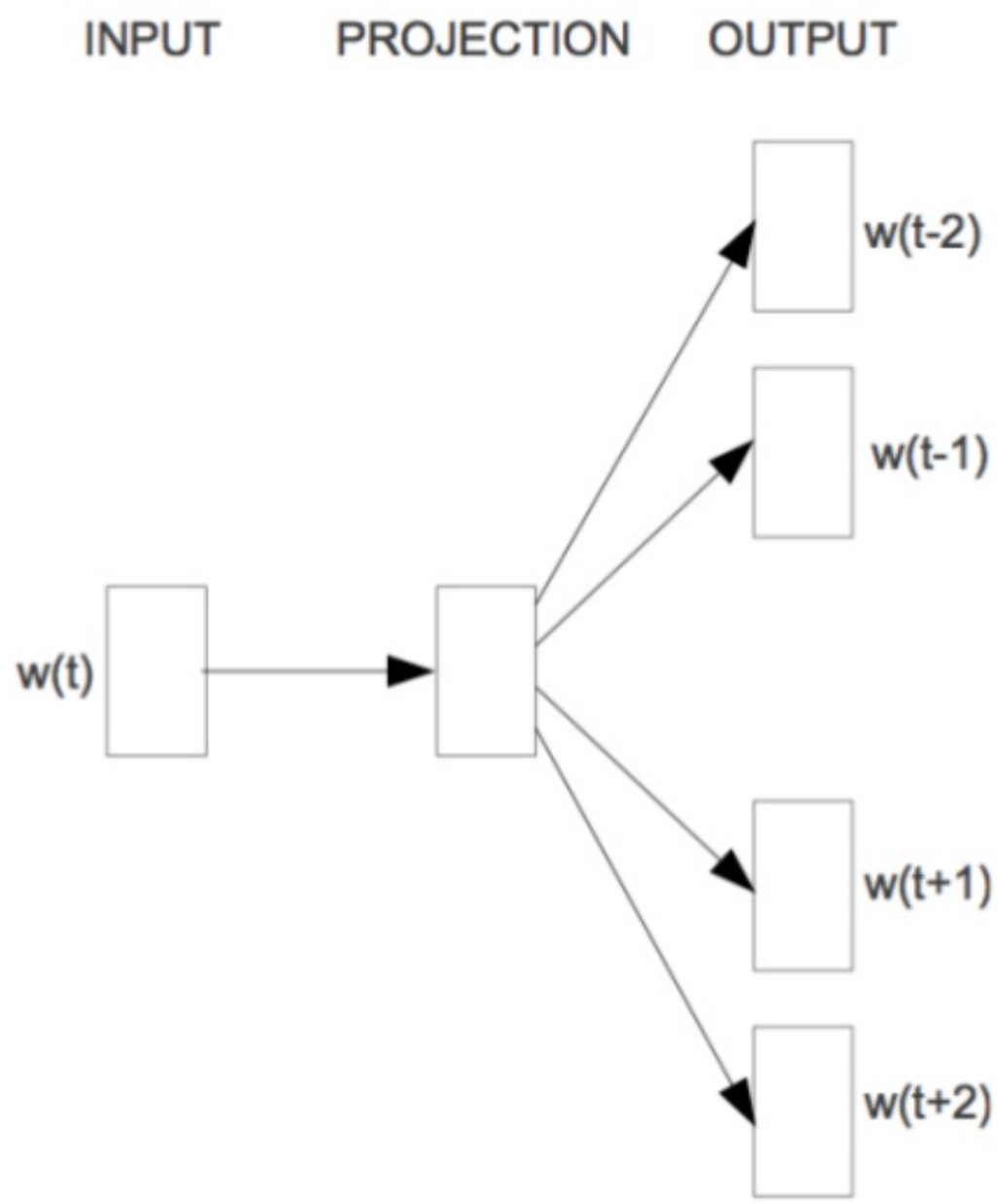
continuous bag-of-words

- predicting the current word **based on the context**
- **order** of words in the history **does not influence the projection**
- faster & **more appropriate for larger corpora**



continuous skip-gram

- maximize classification of a word **based on another word in the same sentence**
- better word vectors for **frequent words**, but slower to train



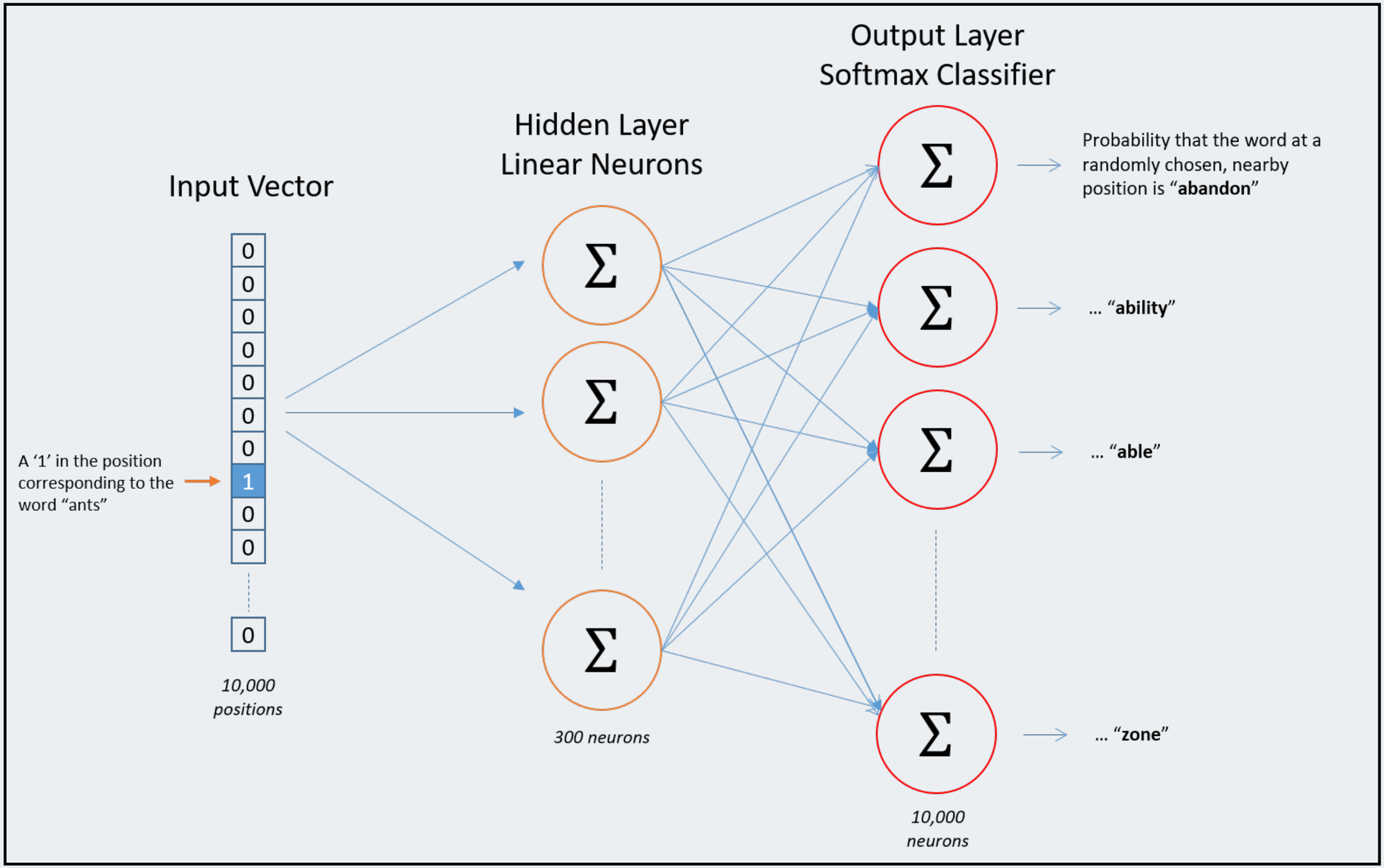
Skip-gram

Pulsar AI

Word2Vec Skip-gram model

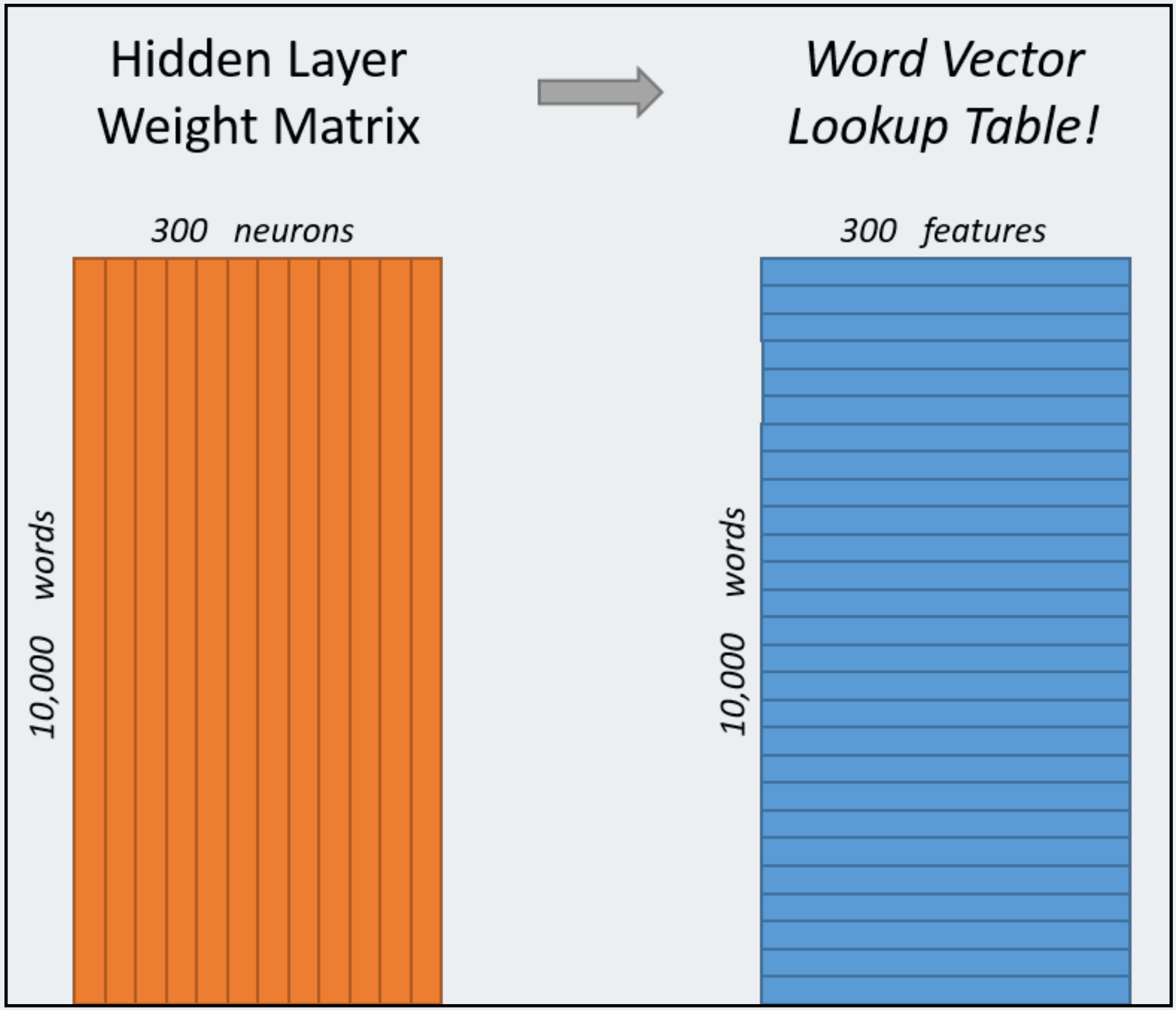
Pulsar AI

word2vec. skip-gram model



Pulsar AI

word2vec. skip-gram model



One-hot vector is almost all zeros… what's the effect of that?

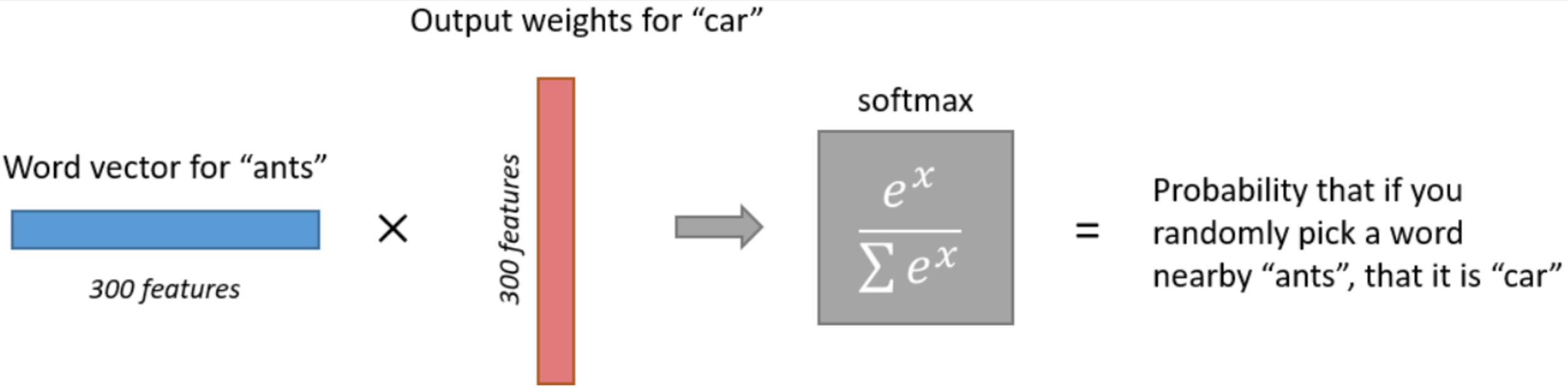
word2vec. skip-gram model

One-hot vector is almost all zeros…
what's the effect of that?

$$[0 \quad 0 \quad 0 \quad \boxed{1} \quad 0] \times \begin{bmatrix} 17 & 24 & 1 \\ 23 & 5 & 7 \\ 4 & 6 & 13 \\ \boxed{10} & \boxed{12} & \boxed{19} \\ 11 & 18 & 25 \end{bmatrix} = [10 \quad 12 \quad 19]$$

Pulsar AI

word2vec. skip-gram model



Pulsar AI

WORD2VEC LIMITATIONS???

Pulsar AI

WORD2VEC LIMITATIONS???

1. Doesn’t take into account the internal structure of words
(bad for morphologically rich languages)

2.Out-of-Vocabulary cases for unseen words

Pulsar AI

fastText by Facebook

Enriching Word Vectors with Subword Information

by Mikolov , Bojanowski, Grave, A. Joulin

2016

https://arxiv.org/abs/1607.04606

Pulsar AI

fastText

The main goal of the Fast Text embeddings is to take into account the internal structure of words while learning word representations – this is especially useful for morphologically rich languages, where otherwise the representations for different morphological forms of words would be learnt independently. The limitation becomes even more important when these words occur rarely.

N = 2 student => ['st', 'tu', 'ud', 'de', 'en', 'nt']

N = 4 student => ['stud', 'tude', 'uden', 'dent']

Pulsar AI

fastText

fastText differs from word2vec only in that it uses char n-gram embeddings as well as the actual word embedding in the scoring function to calculate scores and then likelihoods for each word, given a context word. In case char n-gram embeddings are not present, this reduces (at least theoretically) to the original word2vec model.

This can be implemented by setting 0 for the max length of char n-grams for fastText.

Pulsar AI

WORD2VEC vs FastText Comparision

Pulsar AI

fastText vs word2vec comparision

Syntactic test set

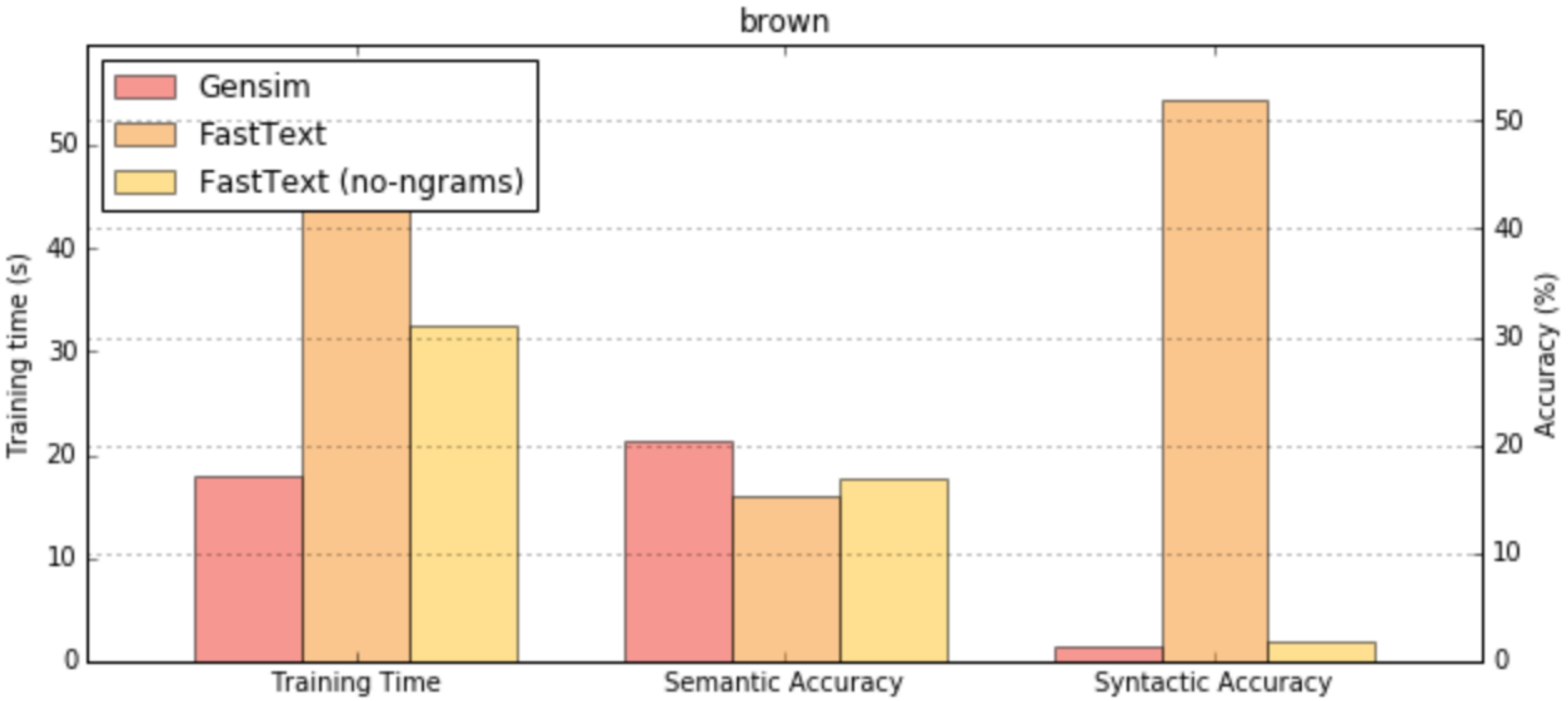
```
work works estimate estimates
work works find finds
work works generate generates
work works go goes
work works implement implements
work works increase increases
work works listen listens
work works play plays
work works predict predicts
work works provide provides
work works say says
```

Semantic test set

```
Ljubljana Slovenia Tbilisi Georgia
London England Luanda Angola
London England Lusaka Zambia
London England Madrid Spain
London England Managua Nicaragua
London England Manama Bahrain
London England Manila Philippines
London England Maputo Mozambique
London England Minsk Belarus
London England Mogadishu Somalia
London England Monrovia Liberia
London England Montevideo Uruguay
London England Moscow Russia
```

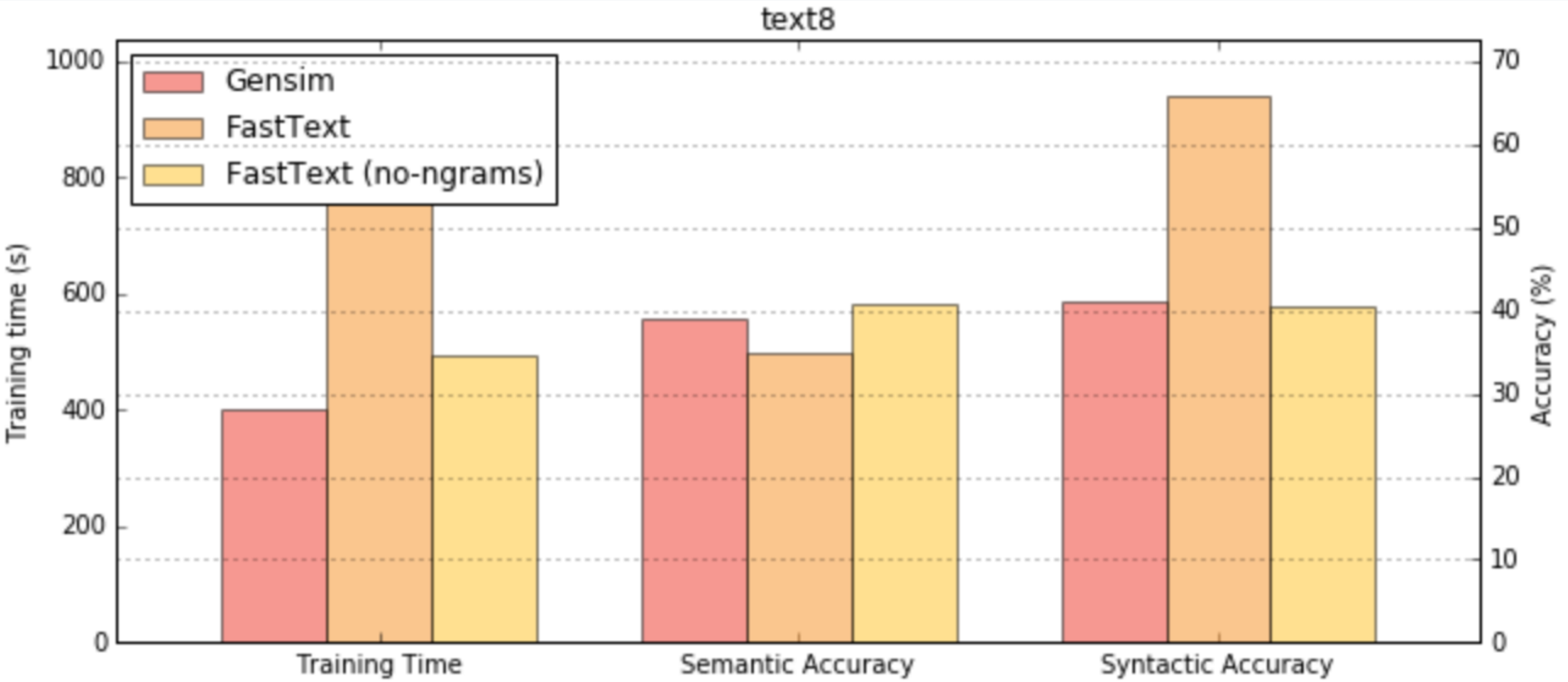
Pulsar AI

fastText vs word2vec comparision



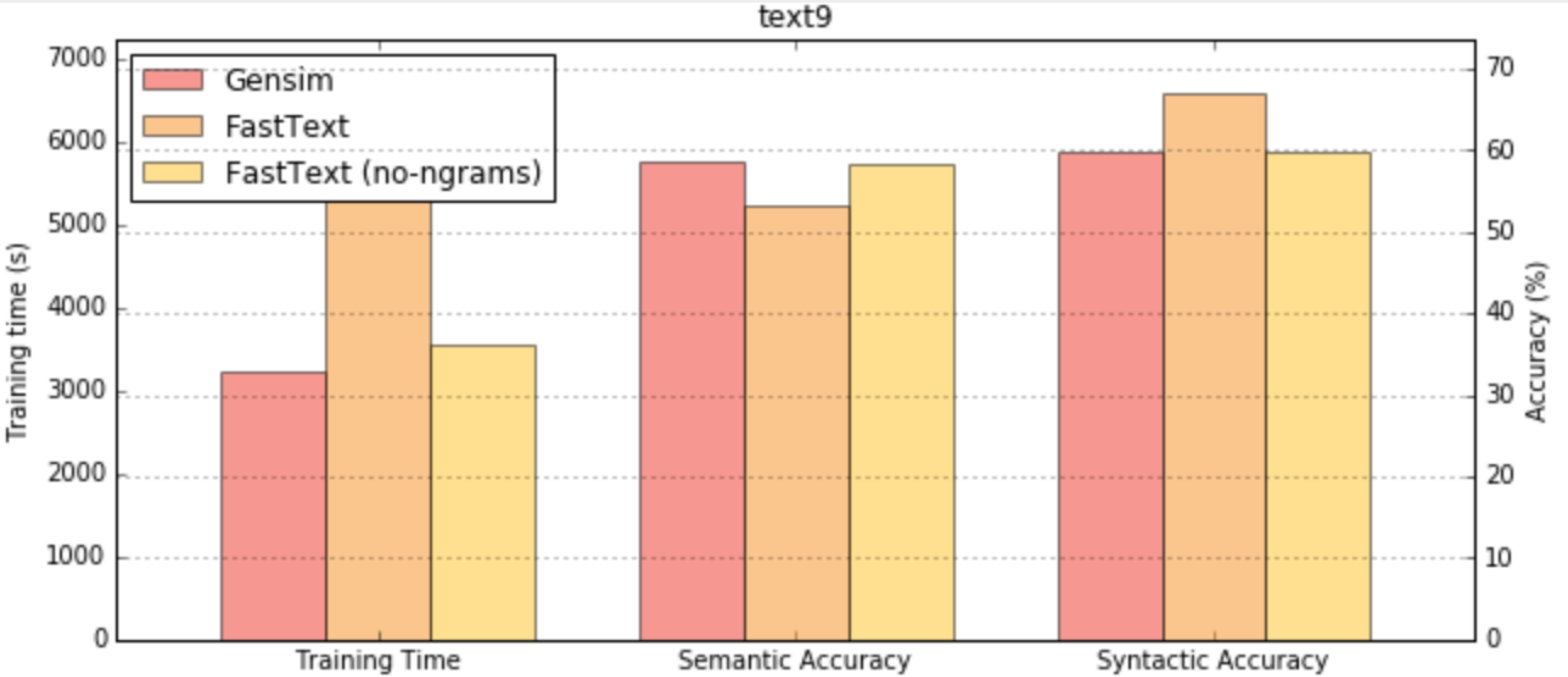
Pulsar AI

fastText vs word2vec comparision



Pulsar AI

fastText vs word2vec comparision



Pulsar AI

TOOLS

gensim

https://radimrehurek.com/gensim/

fastText

https://fasttext.cc

Pulsar AI

LINKS

- https://www.datascience.com/resources/notebooks/word-embeddings-in-python
- http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/
- https://rare-technologies.com/fasttext-and-gensim-word-embeddings/
- https://www.gavagai.se/blog/2015/09/30/a-brief-history-of-word-embeddings/

QUESTIONS?