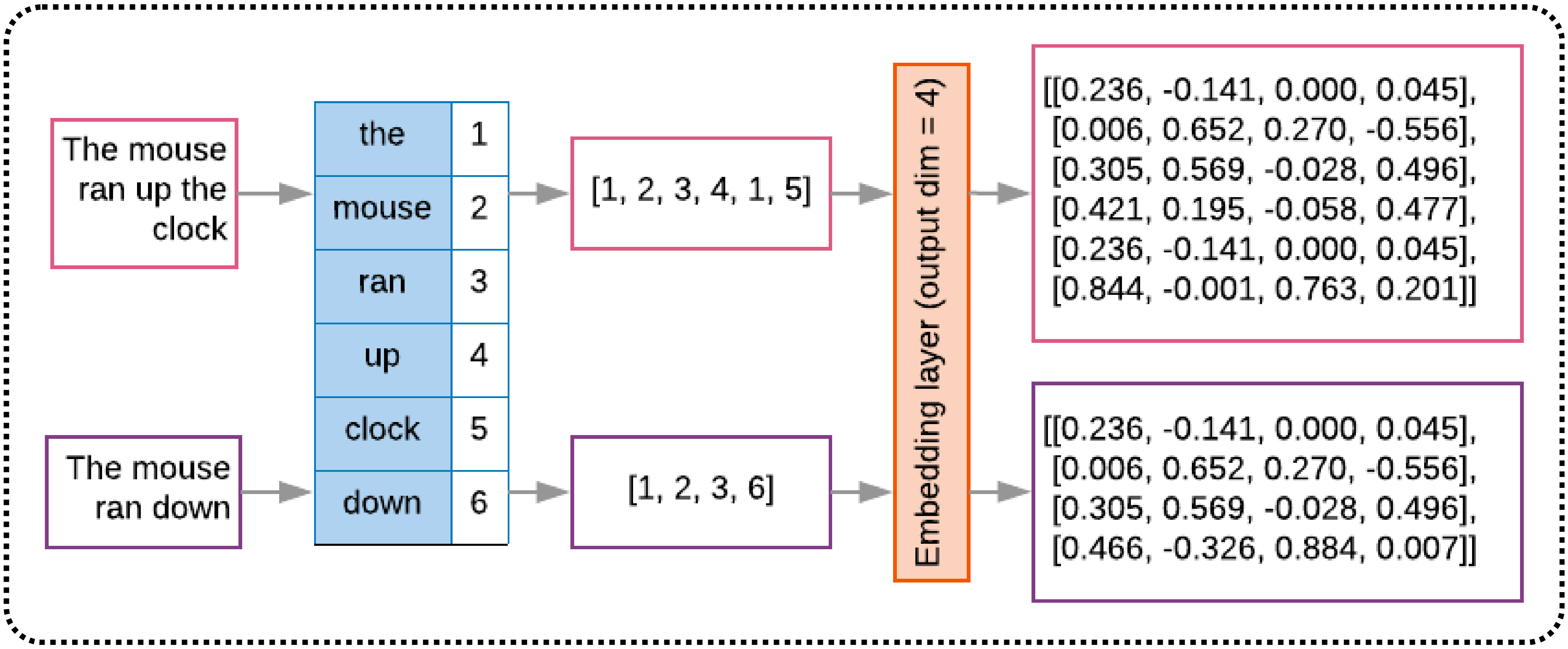
Analytics Vidhya

Mastering LLMs

Day 1: Embedding Era

Converting Text to Continuous Dense Vectors



The mouse ran up the clock

The mouse ran down

the	1
mouse	2
ran	3
up	4
clock	5
down	6

[1, 2, 3, 4, 1, 5]

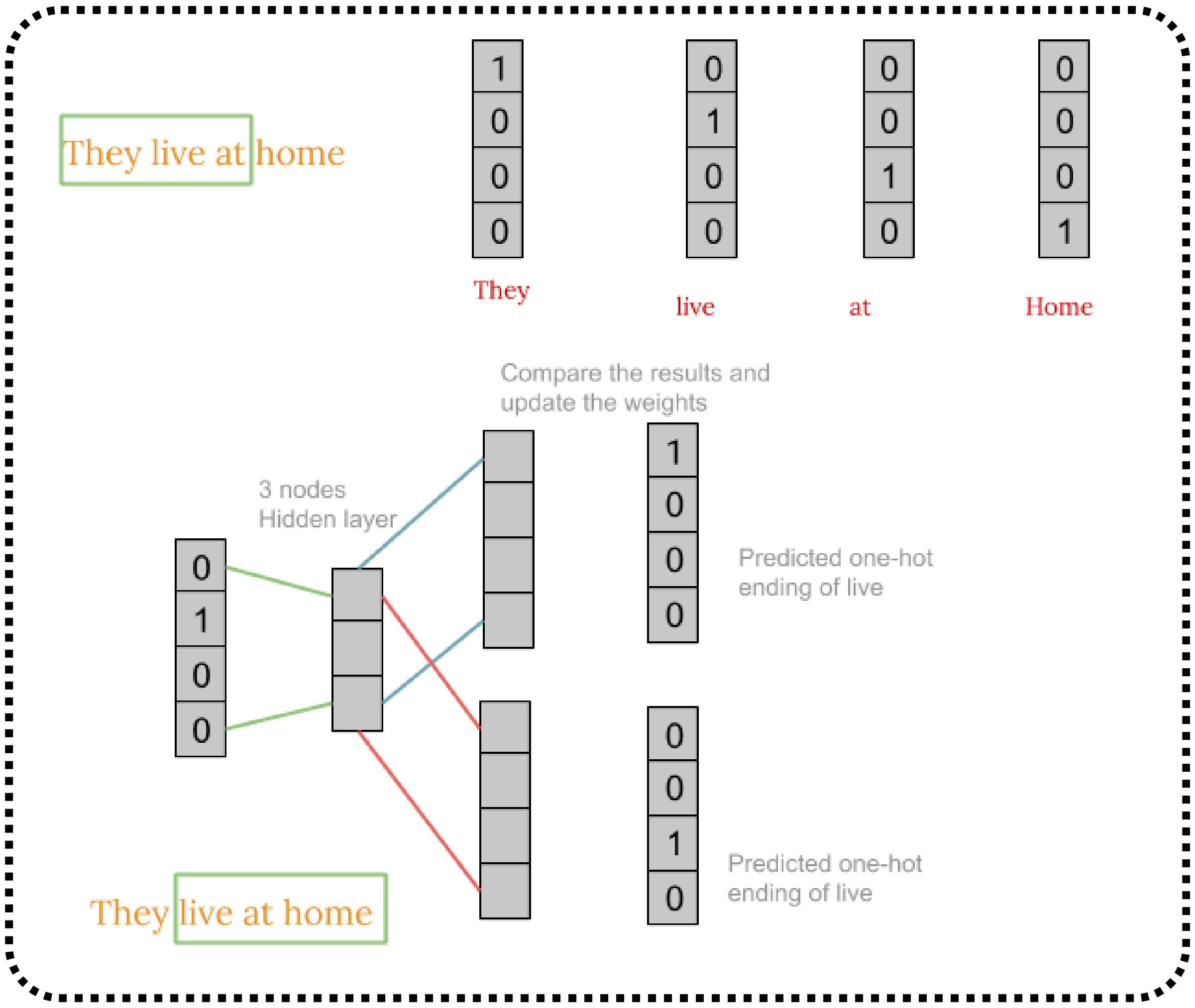
[1, 2, 3, 6]

Embedding layer (output dim = 4)

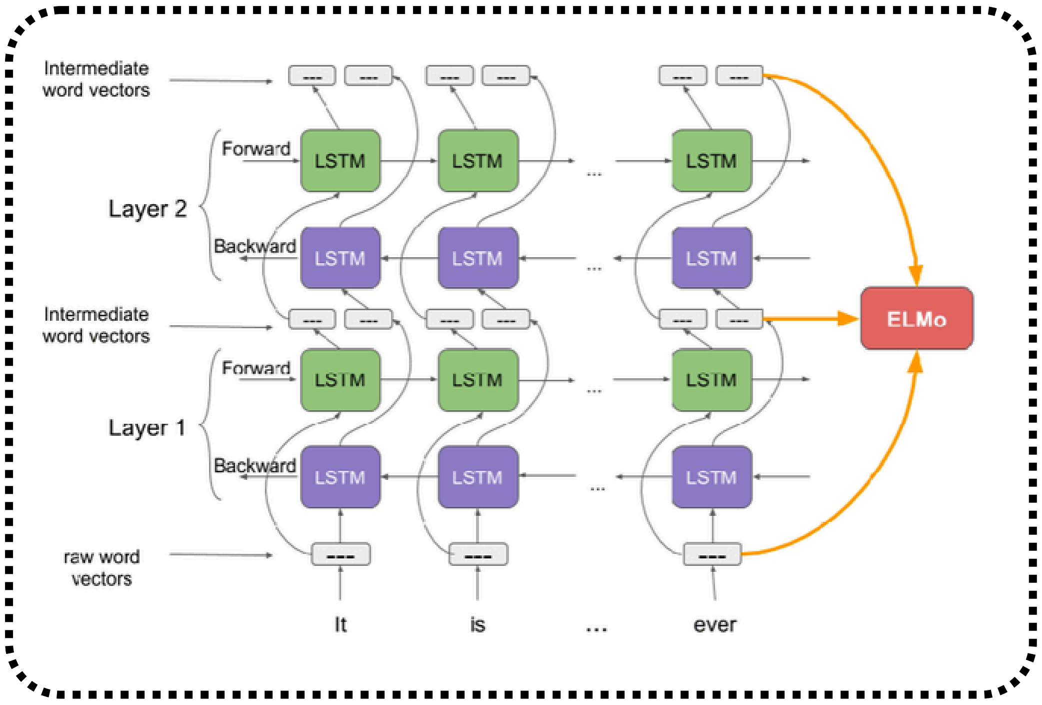
[[0.236, -0.141, 0.000, 0.045],
[0.006, 0.652, 0.270, -0.556],
[0.305, 0.569, -0.028, 0.496],
[0.421, 0.195, -0.058, 0.477],
[0.236, -0.141, 0.000, 0.045],
[0.844, -0.001, 0.763, 0.201]]

[[0.236, -0.141, 0.000, 0.045],
[0.006, 0.652, 0.270, -0.556],
[0.305, 0.569, -0.028, 0.496],
[0.466, -0.326, 0.884, 0.007]]

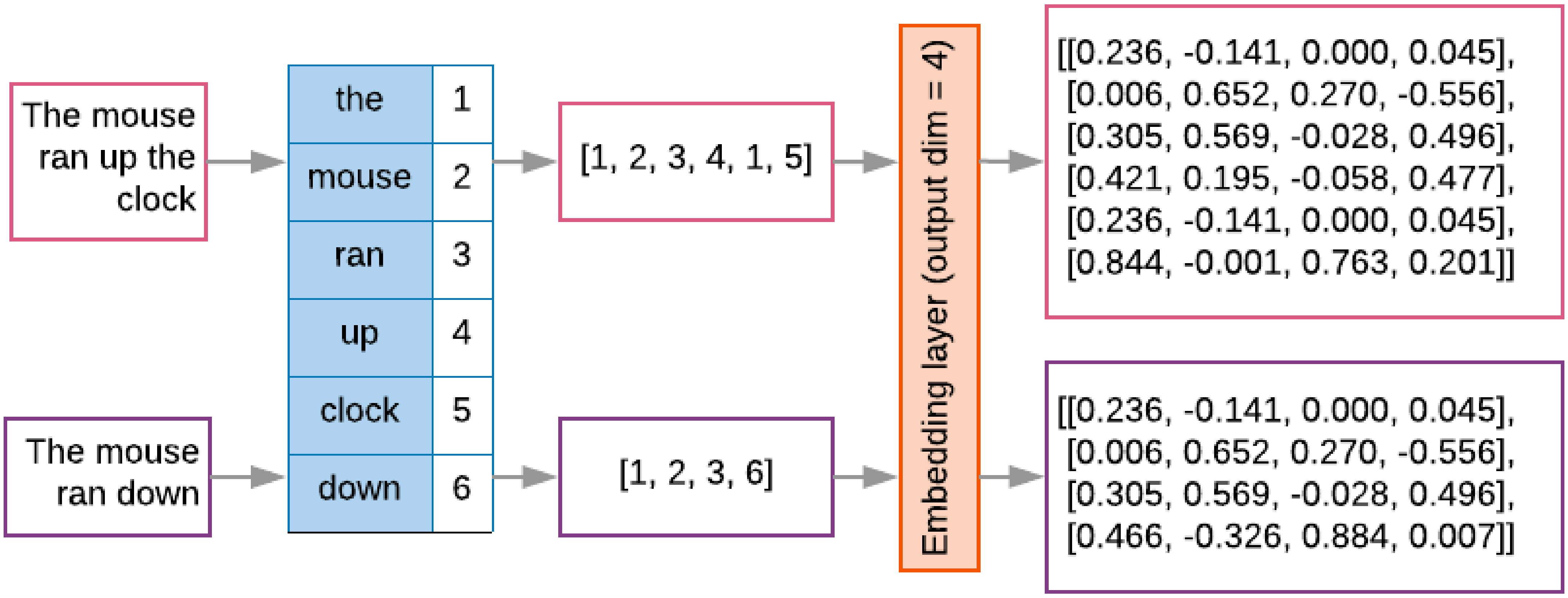
GloVe



ELMo



After the Machine Learning and Classical NLP era, researchers took different perspectives for **representing Text**



The mouse ran up the clock

The mouse ran down

the	1
mouse	2
ran	3
up	4
clock	5
down	6

```
[1, 2, 3, 4, 1, 5]
```

```
[1, 2, 3, 6]
```

Embedding layer (output dim = 4)

```
[[0.236, -0.141, 0.000, 0.045],
 [0.006, 0.652, 0.270, -0.556],
 [0.305, 0.569, -0.028, 0.496],
 [0.421, 0.195, -0.058, 0.477],
 [0.236, -0.141, 0.000, 0.045],
 [0.844, -0.001, 0.763, 0.201]]
```

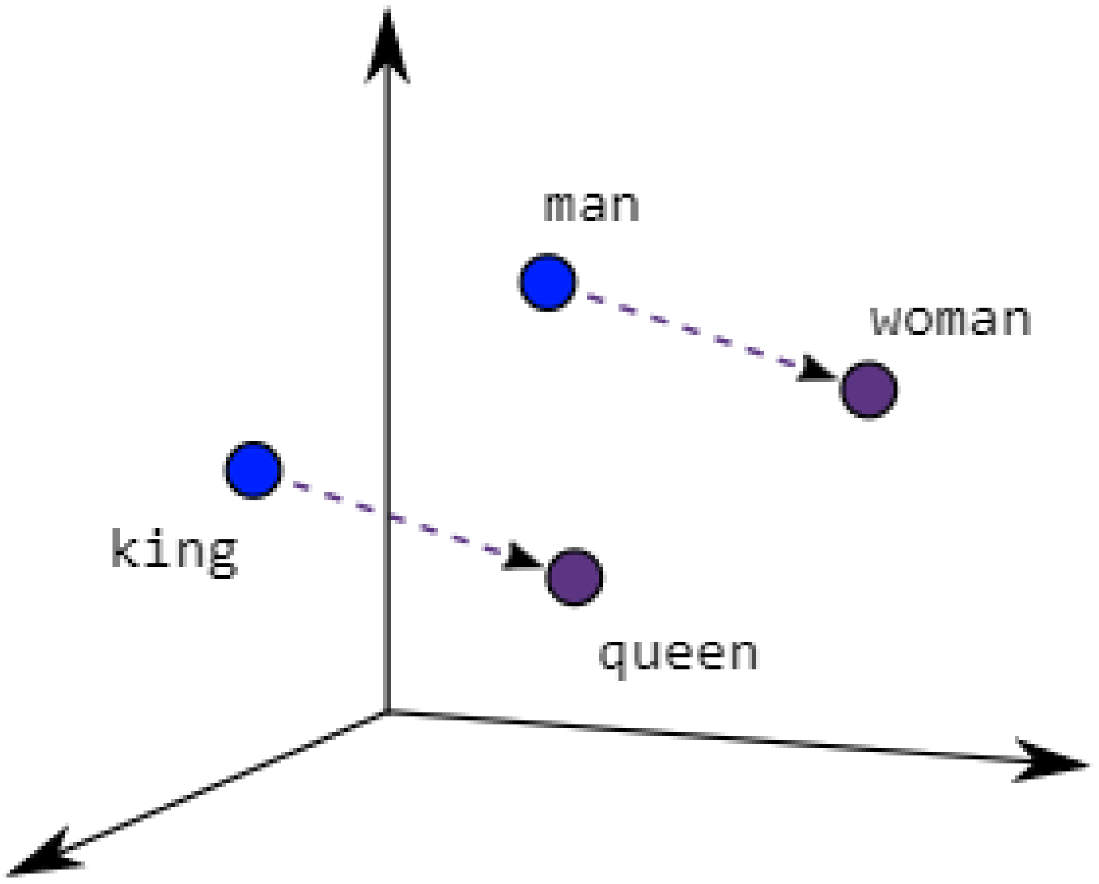
```
[[0.236, -0.141, 0.000, 0.045],
 [0.006, 0.652, 0.270, -0.556],
 [0.305, 0.569, -0.028, 0.496],
 [0.466, -0.326, 0.884, 0.007]]
```

They represented text as a **continuous dense vector** with a lower dimensional space as opposed to sparse high dimensional vectors, because this allowed models to better capture **semantic** relationships between words.

During the embedding era, some popular word embedding techniques came up:

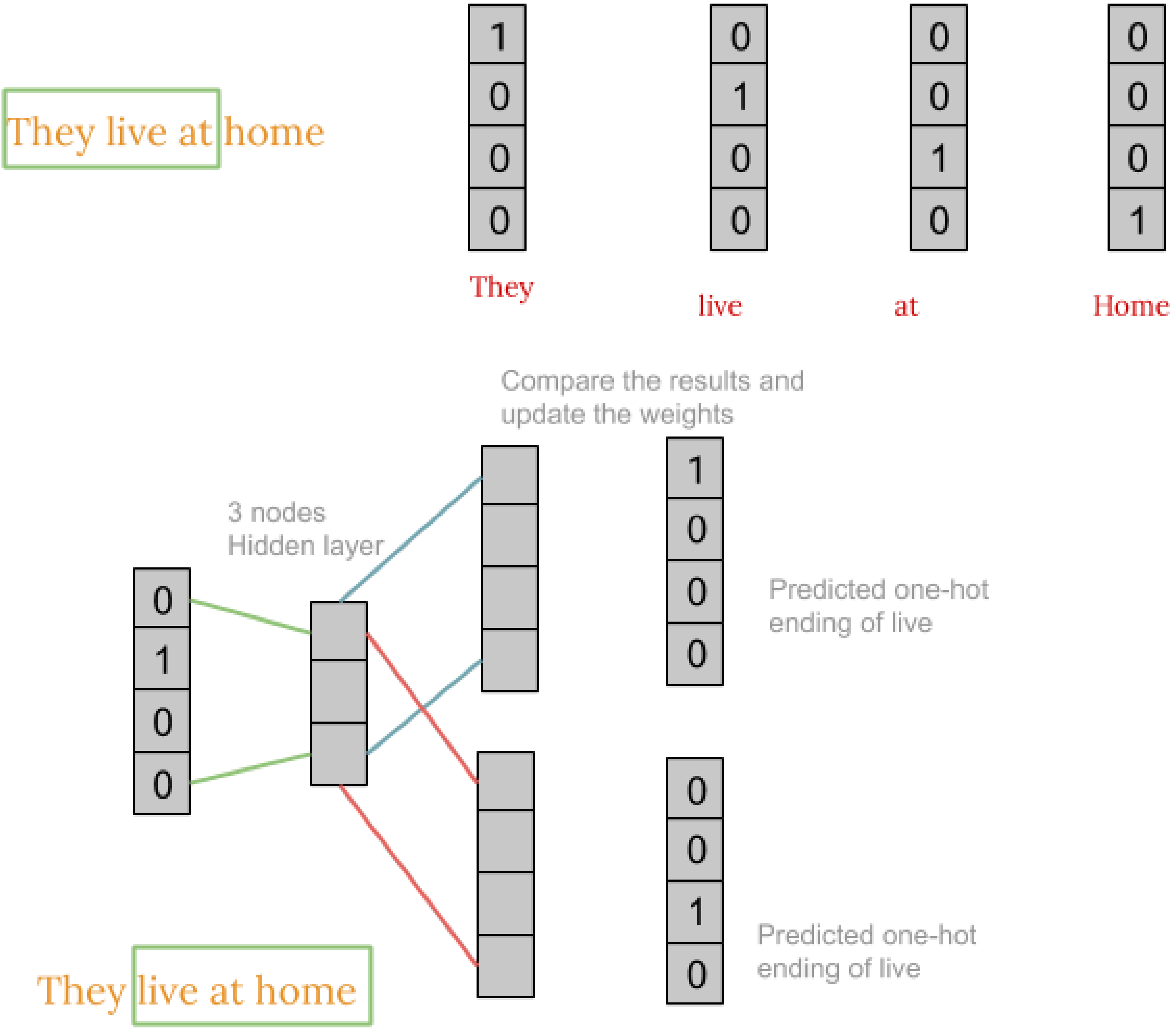
Word2Vec

The smart folks at Google created this model which uses shallow **neural networks** to generate word embeddings, capturing **semantic relationships** and **contextual similarities** between words.



GloVe

Brains at Stanford University, developed GloVe, which looked at how often words appear together or their concurrence.



Word2Vec & GloVe

Global Vectors for Word Representation

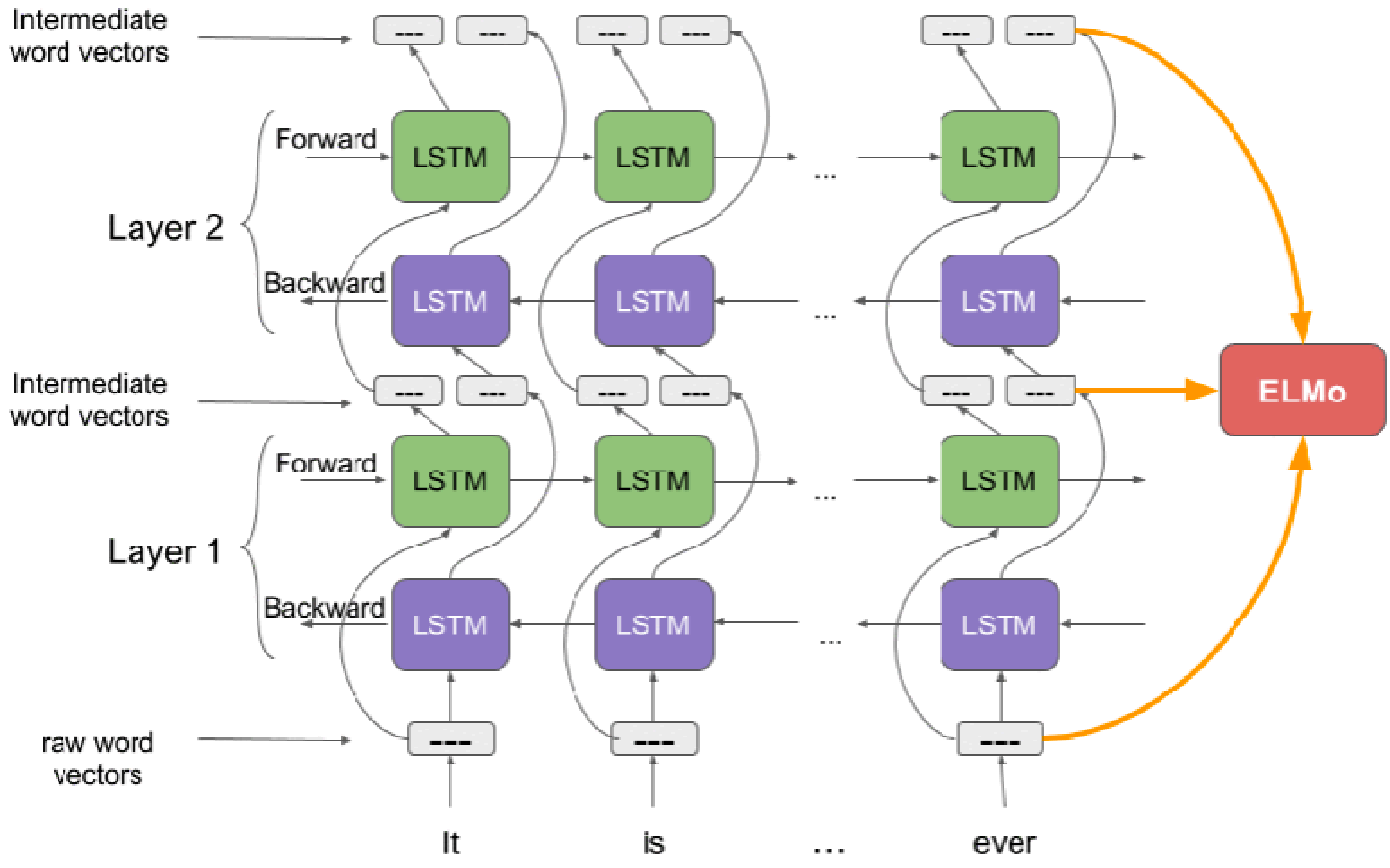
These models leverage **unsupervised** or **self supervised** techniques

These models were a game changer in NLP around 2013-2014, because they made it easier for **algorithms to understand language** in a **better way**.

Making Word Embeddings Better

To make Word Embeddings even better, researchers came out with Contextualised Word Embeddings Techniques:

ELMo → **E**mbeddings from **L**anguage **Mo**dels



- **ELMo** generates contextual embeddings that **capture the meaning** of a word based on its **usage** in a specific sentence or context.

- ELMo embeddings are derived from a deep bidirectional language model (**biLM**), which captures information from both the left and right contexts of a word.

- ELMo generates embeddings for words that vary depending on the context in which they appear.

- For example, the word "bank" in the sentence "I deposited money in the bank" will have a different embedding than in "The river overflowed near the bank."

Limitation of Embedding Era

Static Embeddings

- **Issue**: Models like Word2Vec and GloVe generate a single, fixed vector for each word, regardless of its context in a sentence.
- **Impact**: Words with multiple meanings (polysemy) are poorly represented. For example, "bank" (a financial institution) and "bank" (a riverbank) have the same vector, causing ambiguity.

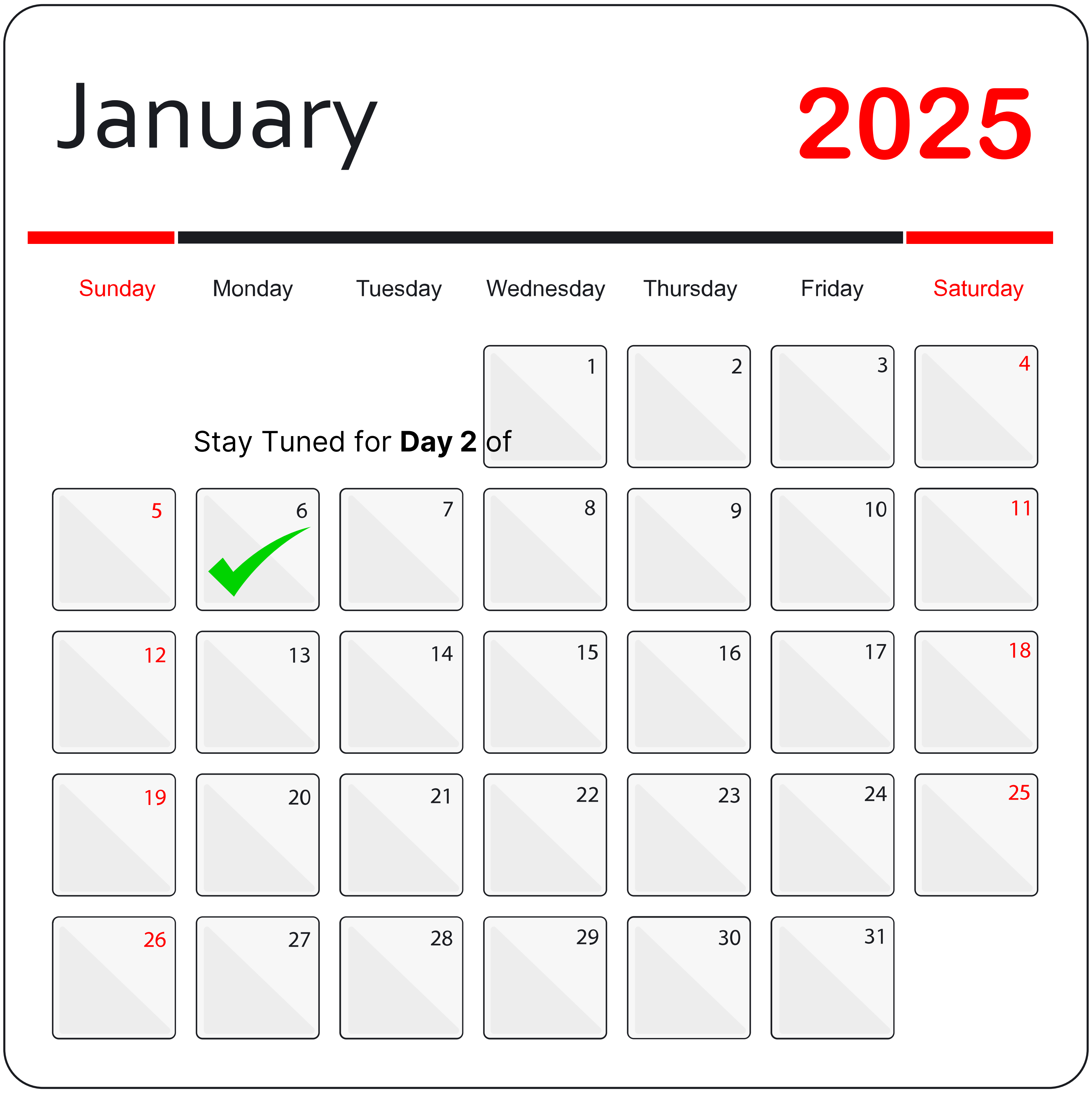
Linear Relationships Only

- **Issue**: Word embeddings rely on linear algebraic properties (e.g., king - man + woman≈queen).

- **Impact**: This oversimplifies complex word relationships, which are often nonlinear in nature.

The **Embedding Era** brought a lot of advancement in the NLP with powerful vector embedding representations but it was the emergence of the **Transformer models** that really took things to the next level.

Analytics Vidhya



January 2025

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
			1	2	3	4
5	6 ✓	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

Stay Tuned for **Day 2** of

Stay Tuned for **Day 2** of

Mastering LLMs