# Big Data Analytics

## Word2Vec

Randy ANDRIAMANORO
Maxence RENIER

Carolyne VERET
Victoria STROPPIANA

Aya BOUMEDIENE
Dylan ESTEVES

# Summary

1- What is Word2Vec ?

2- How Word2Vec works ?

3- Application

4- Benefits & Limits of the Word2Vec

# What is Word2Vec ?

## Introduction

- A word embedding algorithm developed by a Google research team led by Tomas Mikolov
- Uses two-layser neural networks to learn vector representations of words
- Represents words with similar contexts as close numerical vectors

# What is Word2Vec ?

## Neural Architectures

- CBOW : Continuous Bag of Words
- Skip - Gram
- Training Process
- Key Parameters : The dimensionality of the vector space to be constructed and the size of the context of a word

# How Word2Vec Works

2 architectures :

a. Continuous Bag of Words (CBOW):

- Objective: Predict a single target word given the words surrounding it.
    - For the sentence "The quick brown fox jumps," if the target word is "brown," the context words are "The," "quick," "fox," and "jumps."
    - The input is a one-hot encoded representation of these context words.
    - The network outputs the most likely word for the given context, which in this case should be "brown."

# How Word2Vec Works

b. Skip-gram:

- Objective: Predict the surrounding context words given a single target word.
    - The model takes one word (e.g., "brown") as input and tries to predict its context words ("The," "quick," "fox," "jumps").
    - It works well for small datasets and is good at capturing semantic relationships between rare words.

# Application

We did it in **five steps** :

1- Define a text

2- Delete all the words that was useless

3- Apply Word2Vec Model

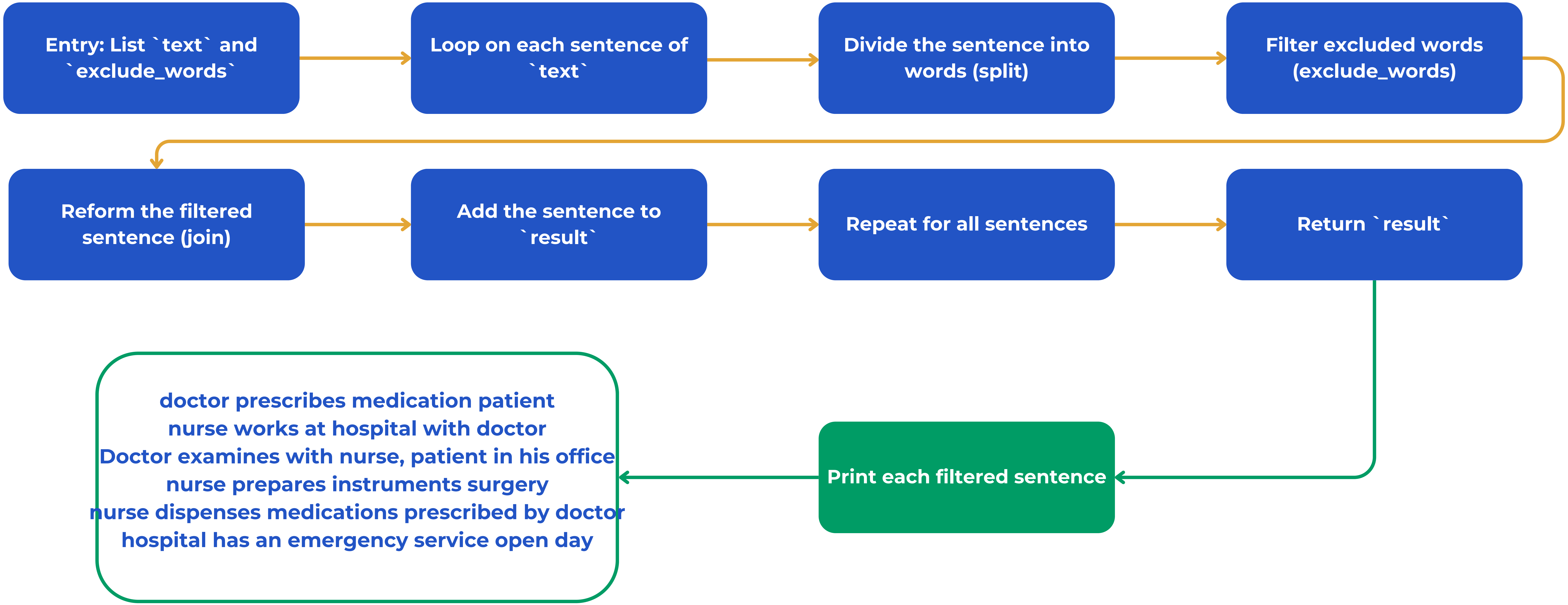
4- Analyse the results

5 - Representation of the results

# 1- Define the text

```
["The doctor prescribes a medication for the patient",
 "The nurse works at the hospital with the doctor",
 "Doctor examines with a nurse, the patient in his office",
 "The nurse prepares the instruments for surgery",
 "A nurse dispenses medications prescribed by a doctor",
 "The hospital has an emergency service open all the day"]
```

# 2- Delete the words



Entry: List `text` and `exclude_words`

Loop on each sentence of `text`

Divide the sentence into words (split)

Filter excluded words (exclude_words)

Reform the filtered sentence (join)

Add the sentence to `result`

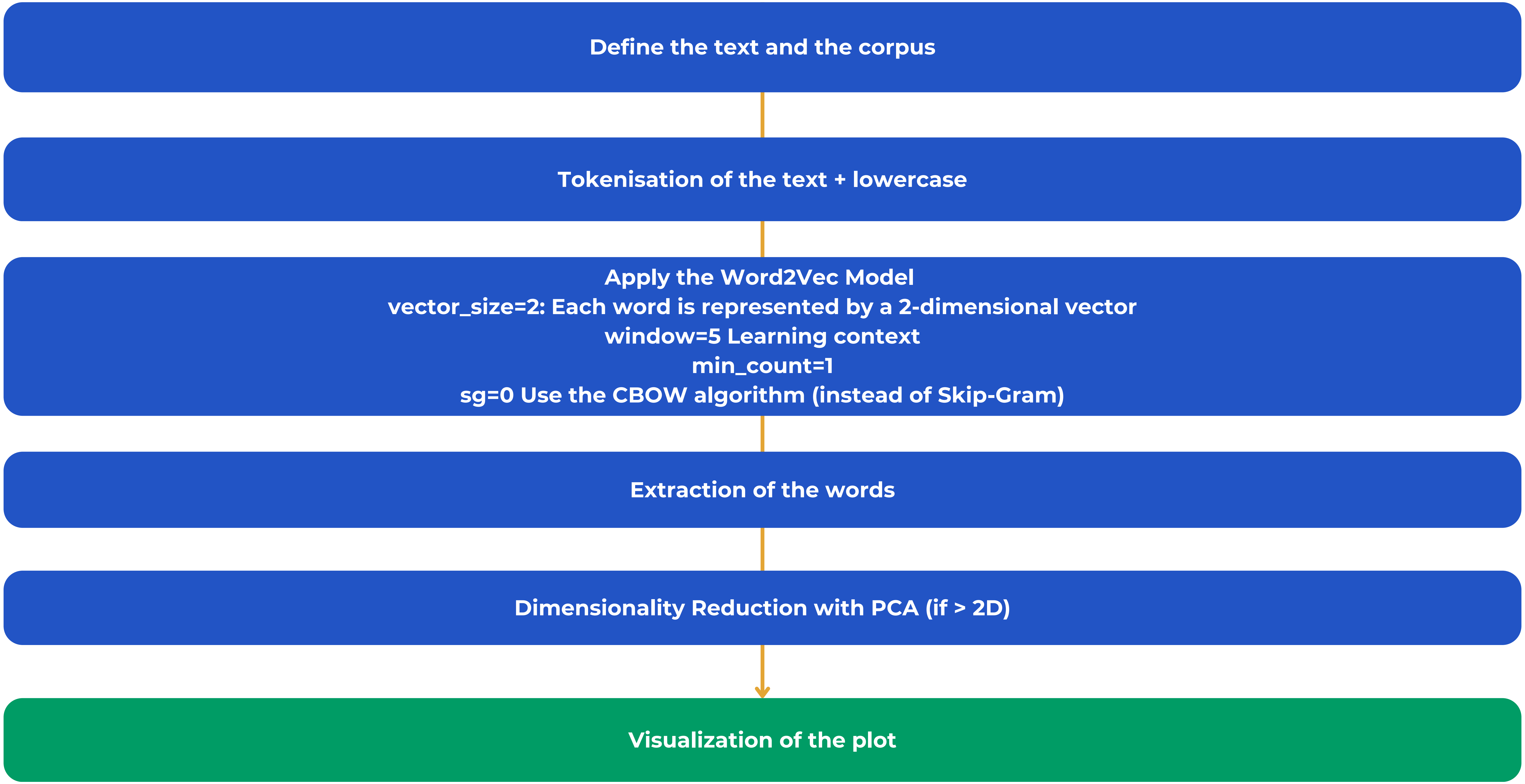
Repeat for all sentences

Return `result`

Print each filtered sentence

```
doctor prescribes medication patient
nurse works at hospital with doctor
Doctor examines with nurse, patient in his office
nurse prepares instruments surgery
nurse dispenses medications prescribed by doctor
hospital has an emergency service open day
```

# 3- Apply Word2Vec



Define the text and the corpus

Tokenisation of the text + lowercase

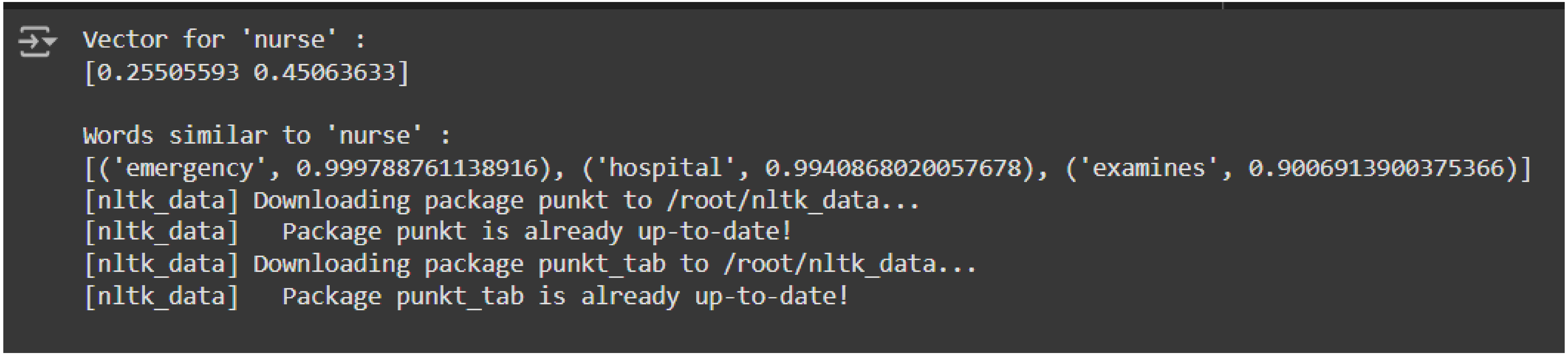
Apply the Word2Vec Model
vector_size=2: Each word is represented by a 2-dimensional vector
window=5 Learning context
min_count=1
sg=0 Use the CBOW algorithm (instead of Skip-Gram)

Extraction of the words

Dimensionality Reduction with PCA (if > 2D)

Visualization of the plot

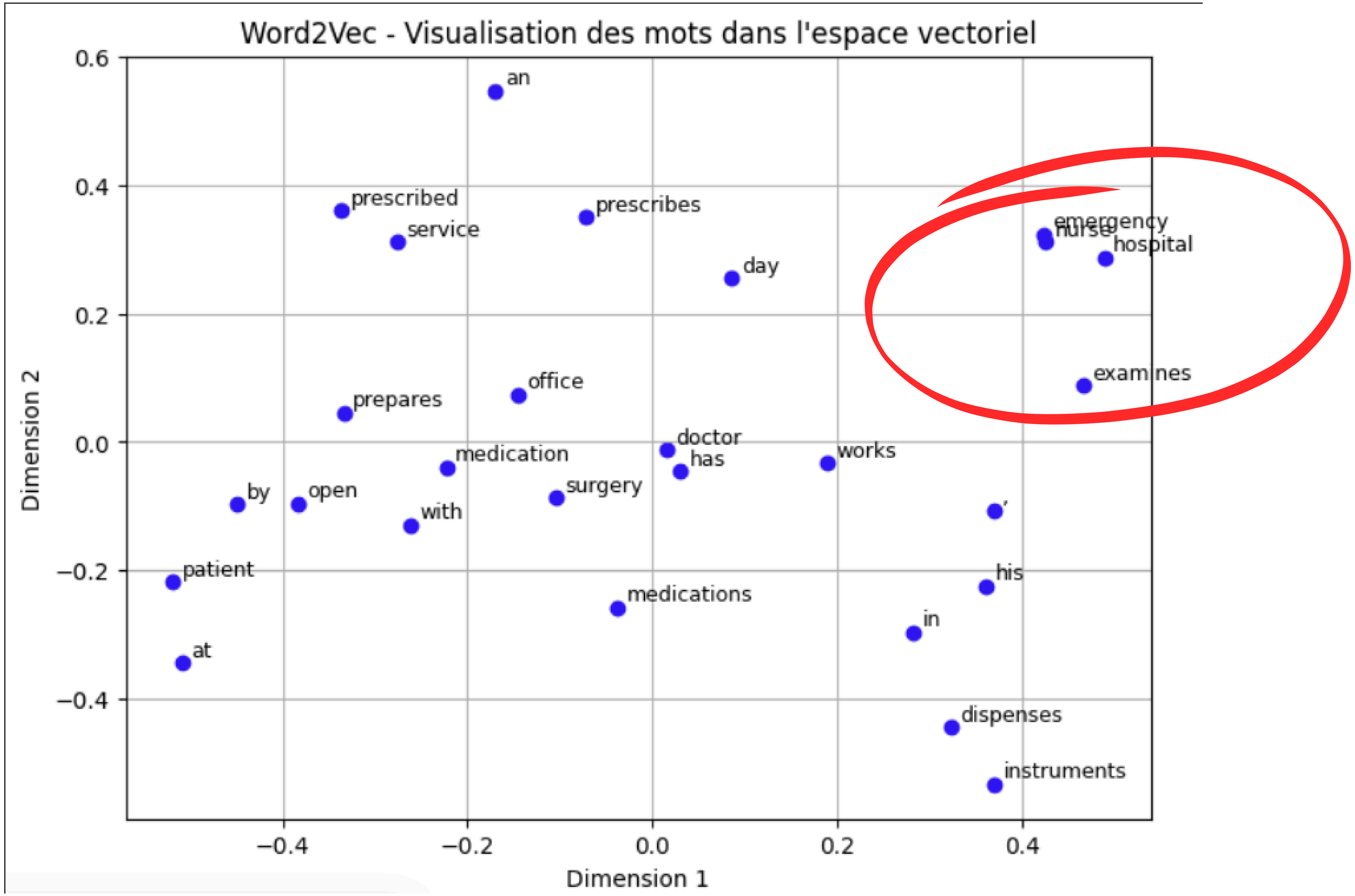
# 4 - Analyse the results



```
Vector for 'nurse' :
[0.25505593 0.45063633]

Words similar to 'nurse' :
[('emergency', 0.999788761138916), ('hospital', 0.9940868020057678), ('examines', 0.9006913900375366)]
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Package punkt_tab is already up-to-date!
```

# 5 - Representation of the results



# Benefits & Limits of the word2vec

## A. Benefits

### 1/ Captures semantic relationships:

- Similar words are close in the vector space (e.g., "cat" and "dog").
- Enables mathematical analogies.

### 2/ Compact vectors:

- Fixed size (e.g., 100 dimensions), regardless of vocabulary size.
- Reduces memory and computational requirements.

### 3/ Better contextual understanding:

- Represents words based on their context.
- Richer than traditional representations (e.g. one-hot encoding).

### 4/ Wide range of applications:

- Semantic search, machine translation, sentiment analysis.

## B. Limits

### 1/ Fixed global context:

- One word = one vector, regardless of the context (e.g., "bank" [finance] vs. "bank" [river] share the same vector).

### 2/ Issues with rare or unknown words:

- Words not present in the training corpus have no vector representation.

### 3/ Corpus-dependent quality:

- The quality of vectors depends on the diversity and relevance of the training corpus.

### 4/ Outdated by modern models:

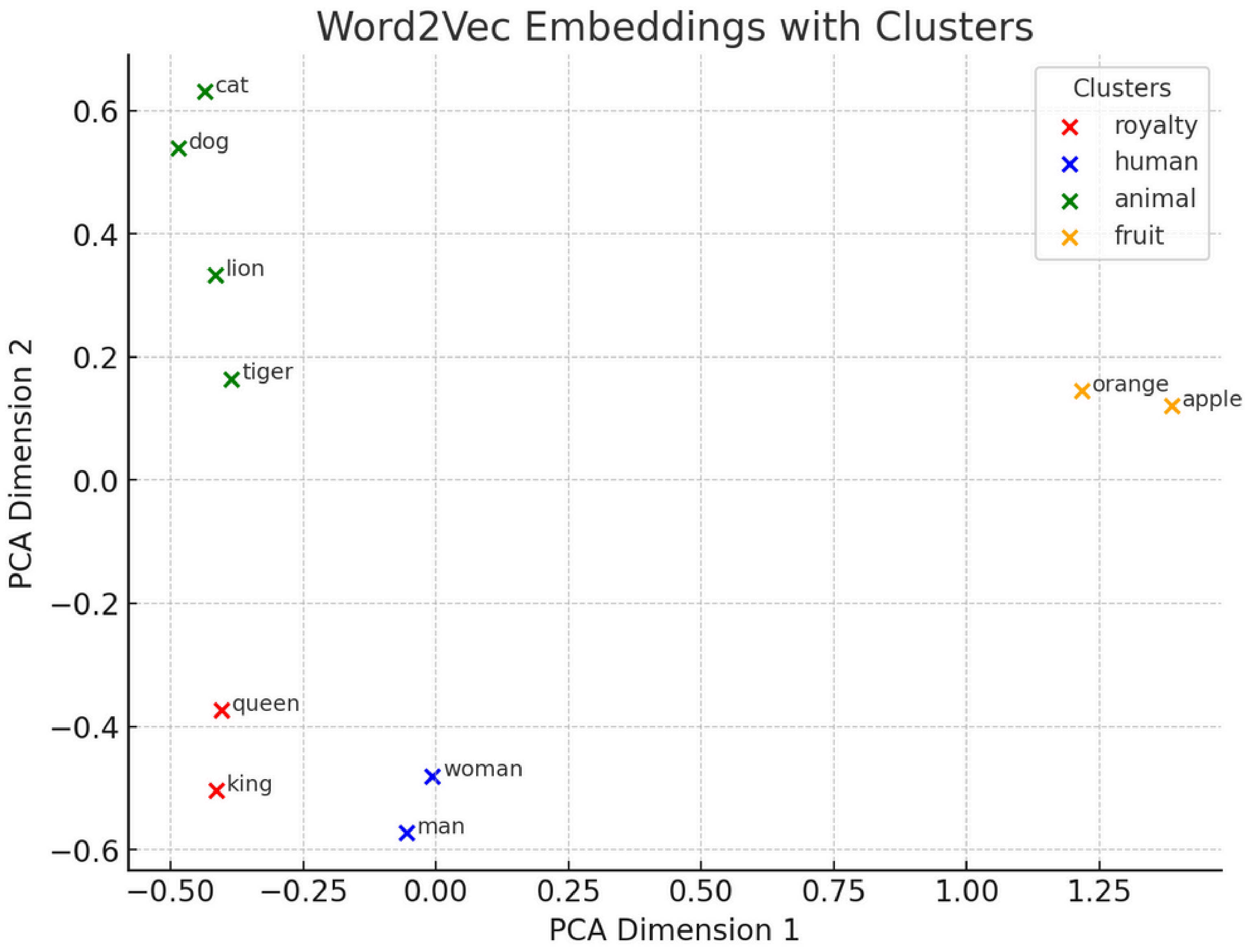
- Techniques like BERT or GPT handle dynamic contexts and complex sentences better.

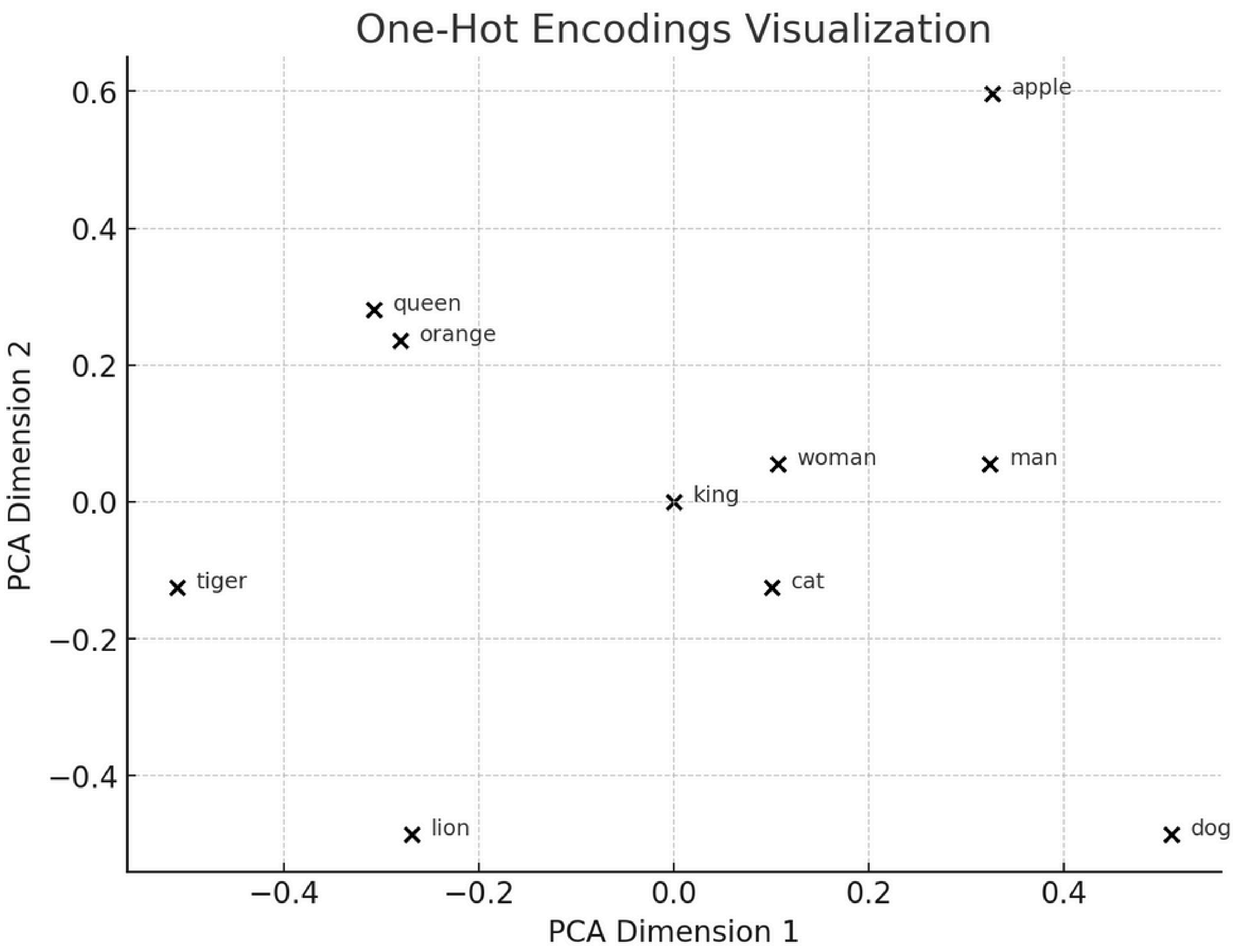
# Why using Word2vec instead of one-hot encoding

| Criteria | One-Hot Encoding | Word2Vec |
|---|---|---|
| Vector size | Long, proportional to vocabulary size (too many dimensions) | Short and fixed (e.g., 100 dimensions) |
| Word relationships | None (independent vectors) | Captures relationships (e.g., king → queen) |
| Information content | None | Rich (contextual relationships) |
| Efficiency | Inefficient for large vocabularies | Compact and fast |
| Training | Slow and resource-intensive | Faster convergence |

➡ Word2Vec is meaningful, and efficient, making it ideal.

# Why using Word2vec instead of one-hot encoding





# Thank You