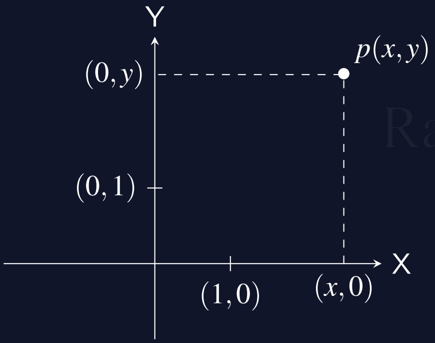
Advanced Programming

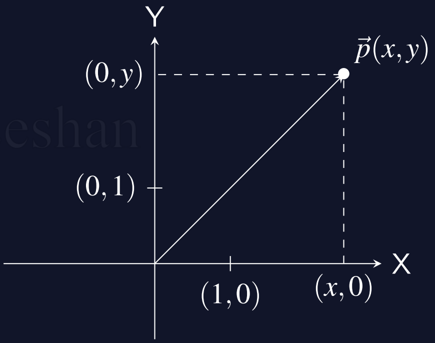
Word to Vector

Ramaseshan Ramachandran

2-D VECTOR SPACE

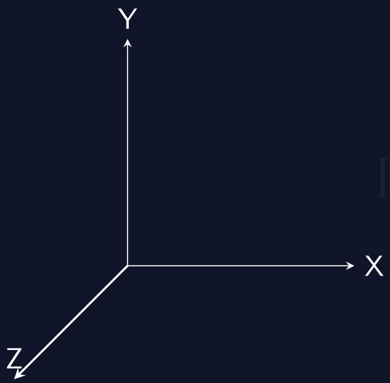
A 2-D vector-space is defined as a set of linearly independent basis vectors with 2 axes. Each axis corresponds to a dimension in the vector-space

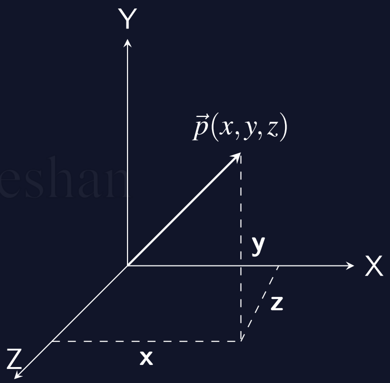




3-D VECTOR SPACE

A 3-D vector-space is defined as a set of linearly independent basis vectors with 3 axes. Each axis corresponds to a dimension in the vector-space





Linearly independent vectors of size $\mathscr{N}$ will result in $\mathscr{N}$-dimensional axes which are mutually orthogonal to each other

VECTOR SPACE MODEL FOR WORDS

Let us assume that the words in a corpus are considered as linearly independent basis vectors.
If a corpus contains $|\mathscr{V}|$ words which are linearly independent, then every word represents an axis in the continuous vector space $\mathscr{R}$.
Each word takes an independent axis which is orthogonal to other words/axes.
Then $\mathscr{R}$ will contain $|\mathscr{V}|$ axes.

Examples

1. The vocabulary size of *emma corpus* is 7079. If we plot all the words in the real space $\mathscr{R}$, we get 7079 axes
2. The vocabulary size of *Google News Corpus corpus* is 3 million. If we plot all the words in the real space $\mathscr{R}$, we get 3 million axes

DOCUMENT VECTOR SPACE MODEL

- Vector space models are used to represent words in a continuous vector space $\mathscr{R}$

DOCUMENT VECTOR SPACE MODEL

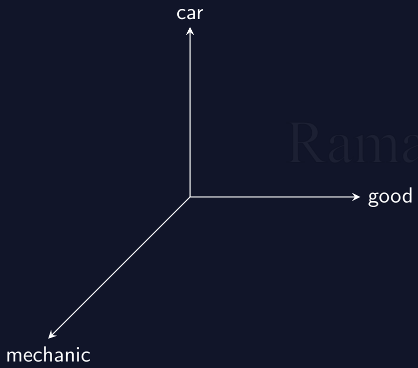
- Vector space models are used to represent words in a continuous vector space $\mathscr{R}$
- Combination of Terms represent a document vector in the word vector space

DOCUMENT VECTOR SPACE MODEL

- Vector space models are used to represent words in a continuous vector space $\mathscr{R}$
- Combination of Terms represent a document vector in the word vector space
- Very high dimensional space - several million axes, representing terms and several million documents containing several terms

EXAMPLE - BINARY INCIDENCE MATRIX

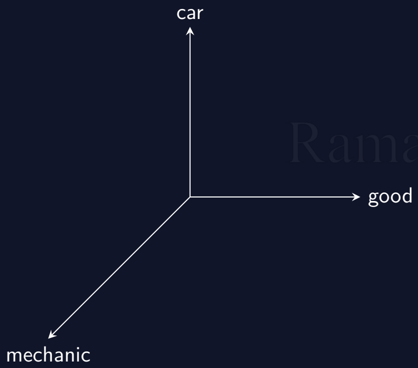
Let us consider three words - *good, car, mechanic* and we will represent these words in a 3-D vector space



	good	car	mechanic
D1	1	1	1
D2	1	0	1
D3	0	1	1

EXAMPLE - BINARY INCIDENCE MATRIX

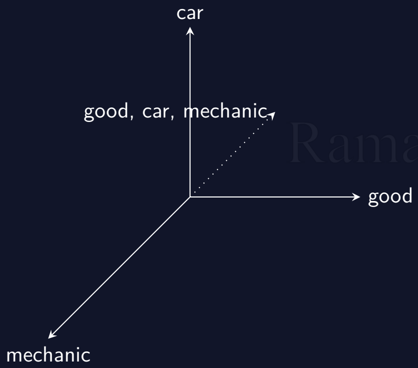
Let us consider three words - *good, car, mechanic* and we will represent these words in a 3-D vector space



	good	car	mechanic
D1	1	1	1
D2	1	0	1
D3	0	1	1

EXAMPLE - BINARY INCIDENCE MATRIX

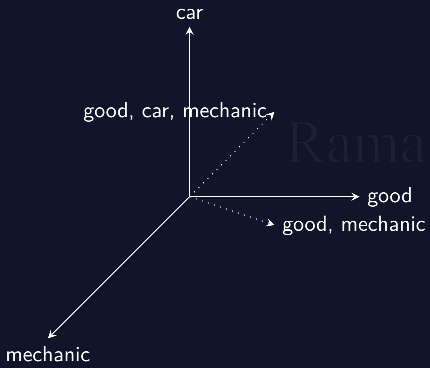
Let us consider three words - *good, car, mechanic* and we will represent these words in a 3-D vector space



	good	car	mechanic
D1	1	1	1
D2	1	0	1
D3	0	1	1

EXAMPLE - BINARY INCIDENCE MATRIX

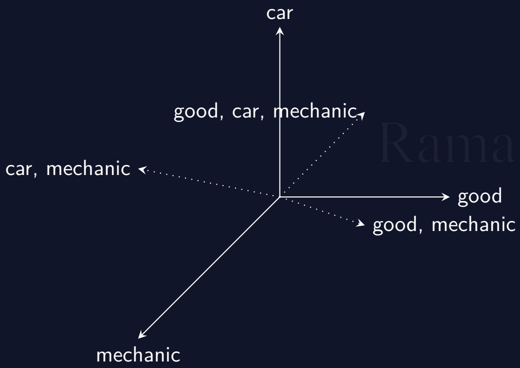
Let us consider three words - *good, car, mechanic* and we will represent these words in a 3-D vector space



	good	car	mechanic
D1	1	1	1
D2	1	0	1
D3	0	1	1

EXAMPLE - BINARY INCIDENCE MATRIX

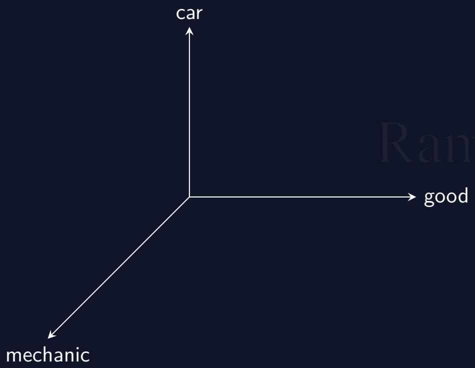
Let us consider three words - *good, car, mechanic* and we will represent these words in a 3-D vector space



	good	car	mechanic
D1	1	1	1
D2	1	0	1
D3	0	1	1

EXAMPLE - TF-IDF INCIDENCE MATRIX

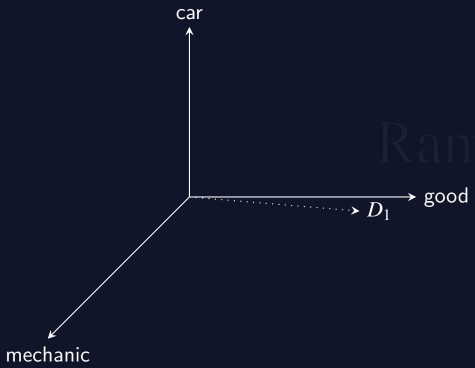
Let us consider three words - *good, car, mechanic* and we will represent these words in a 3-D vector space



	good	car	mechanic
D1	0.91	0	0.0011
D2	0.21	0	0.1
D3	0.15	0	0.921

EXAMPLE - TF-IDF INCIDENCE MATRIX

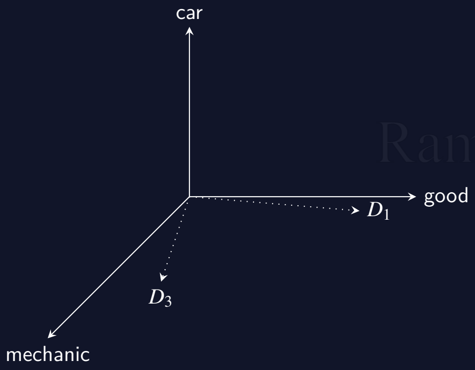
Let us consider three words - *good*, *car*, *mechanic* and we will represent these words in a 3-D vector space



	good	car	mechanic
D1	0.91	0	0.0011
D2	0.21	0	0.1
D3	0.15	0	0.921

EXAMPLE - TF-IDF INCIDENCE MATRIX

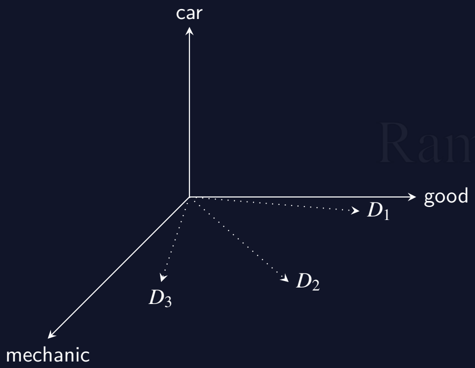
Let us consider three words - *good*, *car*, *mechanic* and we will represent these words in a 3-D vector space



	good	car	mechanic
D1	0.91	0	0.0011
D2	0.21	0	0.1
D3	0.15	0	0.921

EXAMPLE - TF-IDF INCIDENCE MATRIX

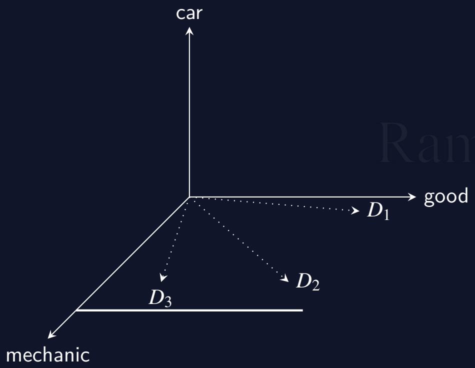
Let us consider three words - *good*, *car*, *mechanic* and we will represent these words in a 3-D vector space



	good	car	mechanic
D1	0.91	0	0.0011
D2	0.21	0	0.1
D3	0.15	0	0.921

EXAMPLE - TF-IDF INCIDENCE MATRIX

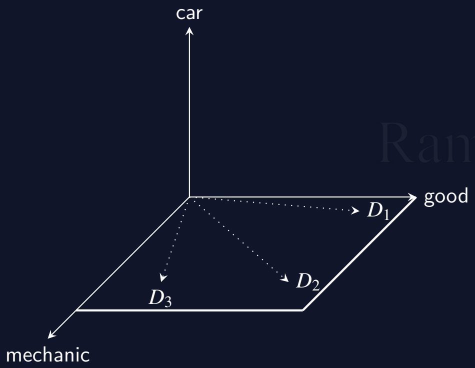
Let us consider three words - *good*, *car*, *mechanic* and we will represent these words in a 3-D vector space



	good	car	mechanic
D1	0.91	0	0.0011
D2	0.21	0	0.1
D3	0.15	0	0.921

EXAMPLE - TF-IDF INCIDENCE MATRIX

Let us consider three words - *good, car, mechanic* and we will represent these words in a 3-D vector space



	good	car	mechanic
D1	0.91	0	0.0011
D2	0.21	0	0.1
D3	0.15	0	0.921

DOCUMENT-TERM MATRIX

	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	d11	d12
t1	0.1	0.0	0.4	0.1	0.2	0.0	0.1	0.9	0.9	0.3	0.0	0.8
t2	0.1	0.0	0.4	0.1	0.2	0.0	0.1	0.9	0.9	0.3	0.0	0.8
t3	0.0	0.9	0.0	0.2	0.3	0.1	0.7	0.0	0.2	0.7	0.5	0.5
t4	0.0	0.9	0.3	0.9	0.5	0.1	0.9	0.3	0.8	0.4	0.1	0.4
t5	0.4	0.0	0.3	0.2	0.5	0.9	0.3	0.7	0.4	0.6	0.0	0.3
t6	0.6	0.0	0.4	0.7	0.3	0.3	0.9	0.1	0.9	0.0	0.0	0.3
t7	0.0	0.8	0.5	0.6	0.6	0.6	0.0	0.1	0.4	0.9	0.3	0.1
t8	0.4	0.0	0.6	0.5	0.5	0.1	0.7	0.1	0.5	0.3	0.8	0.1
t9	0.3	0.0	0.7	0.9	0.8	0.7	0.7	0.8	0.6	0.6	0.8	0.0
t10	0.0	0.5	0.5	0.0	0.2	0.0	0.0	0.1	0.3	0.4	0.5	0.3

The columns of the matrix represent the document as vectors. A document vector is represented by the terms present in the document

TERM-TERM MATRIX

	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	t11	t12
t1	0.1	0.0	0.4	0.1	0.2	0.0	0.1	0.9	0.9	0.3	0.0	0.8
t2	0.1	0.0	0.4	0.1	0.2	0.0	0.1	0.9	0.9	0.3	0.0	0.8
t3	0.0	0.9	0.0	0.2	0.3	0.1	0.7	0.0	0.2	0.7	0.5	0.5
t4	0.0	0.9	0.3	0.9	0.5	0.1	0.9	0.3	0.8	0.4	0.1	0.4
t5	0.4	0.0	0.3	0.2	0.5	0.9	0.3	0.7	0.4	0.6	0.0	0.3
t6	0.6	0.0	0.4	0.7	0.3	0.3	0.9	0.1	0.9	0.0	0.0	0.3
t7	0.0	0.8	0.5	0.6	0.6	0.6	0.0	0.1	0.4	0.9	0.3	0.1
t8	0.4	0.0	0.6	0.5	0.5	0.1	0.7	0.1	0.5	0.3	0.8	0.1
t9	0.3	0.0	0.7	0.9	0.8	0.7	0.7	0.8	0.6	0.6	0.8	0.0
t10	0.0	0.5	0.5	0.0	0.2	0.0	0.0	0.1	0.3	0.4	0.5	0.3
t11	0.01	0.2	0.4	0.1	0.2	0.2	0.0	0.0	0.0	0.1	0.2	0.0
t12	0.1	0.12	0.54	0.01	0.02	0.0	0.0	0.0	0.0	0.6	0.7	0.0

The columns and rows of the matrix represent the words as vectors.

WORD SIMILARITY

A similarity measure is a real-valued function that quantifies the similarity between two objects - in this case words [**Manning2009**]. Some of the similarity measures are given below.

$$\textbf{Euclidean Distance - } \mathscr{E}(\vec{w}_1, \vec{w}_2) = \sqrt{w_1^2 - w_2^2} \tag{1}$$

$$\textbf{Cosine Similarity} = \frac{\vec{w}_1 . \vec{w}_2}{\|\vec{w}_1\| \, \|\vec{w}_2\|} = \frac{\vec{w}_1}{\|\vec{w}_1\|} . \frac{\vec{w}_2}{\|\vec{w}_2\|} \tag{2}$$

$$\textbf{Cosine distance} = 1 - \frac{\vec{w}_1 . \vec{w}_2}{\|\vec{w}_1\| \, \|\vec{w}_2\|} = \frac{\vec{w}_1}{\|\vec{w}_1\|} . \frac{\vec{w}_2}{\|\vec{w}_2\|} \tag{3}$$

$$\textbf{Cluster similarity-} \mathscr{L}(\vec{w}_1, \vec{w}_2) = \frac{\vec{w}_1 . \vec{w}_2}{\|\vec{w}_1\|_1} \tag{4}$$

VECTOR REPRESENTATION OF WORDS

Let V be the unique set of terms and $|V|$ be the size of the vocabulary. Then every vector representing the word $\mathscr{R}^{|V|x1}$ would point to a vector in the V-dimensional space

ONE-HOT VECTOR - 1

Consider all the ≈39000 words (estimated tokens in English is ≈ 13M) in the Oxford Learner's pocket dictionary. We can represent each word as an independent vector quantity as follows in the real space $\mathscr{R}^{|V|X1}$

$$t^{a} = \begin{pmatrix} 1 \\ 0 \\ \dots \\ 0 \\ \dots \\ 0 \\ 0 \end{pmatrix} \; t^{aback} = \begin{pmatrix} 0 \\ 1 \\ \dots \\ 0 \\ \dots \\ 0 \\ 0 \end{pmatrix} \dots t^{zoom} = \begin{pmatrix} 0 \\ 0 \\ \dots \\ 0 \\ \dots \\ 1 \\ 0 \end{pmatrix} \; t^{zucchini} = \begin{pmatrix} 0 \\ 0 \\ \dots \\ 0 \\ \dots \\ 0 \\ 1 \end{pmatrix}$$

This is a very simple codification scheme to represent words independently in the vector space. This is known as ***one-hot vector***.

ONE-HOT VECTOR - 2

In one-hot vector, every word is represented independently. The terms, *home, house, apartments, flats* are independently coded. With one-hot vector based model, the dot product

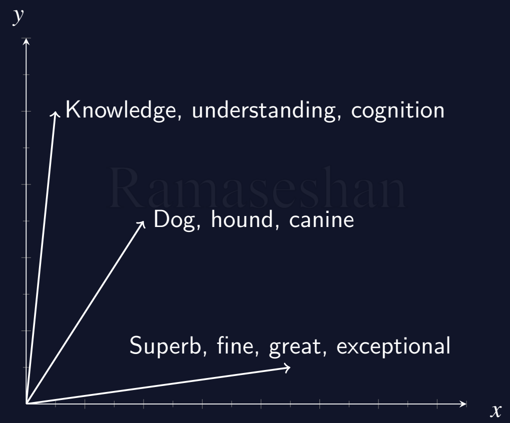
$$\left(t^{House}\right)^{T} \cdot t^{Apartment} = 0 \tag{5}$$

$$\left(t^{Home}\right)^{T} \cdot t^{House} = 0 \tag{6}$$

With one-Hot vector, there is no notion of similarity or synonyms.

RELATIONSHIP AMONG TERMS - SYNONYMS

We could represent all the synonyms of a word in one axis



IDEAL PROPERTIES OF WORD VECTORS

- Reduce word-vector space into a smaller sub-space
- Encode the relationship among words
- Identify similar words using
  - Law of similarity
  - Law of contiguity
  - Law of contrast
  - Law of frequency
- Extract semantic information
- Represent polysemous words

You shall know a word by the company it keeps[1]

[1] Firth, J. R. 1957

CONTEXTUAL UNDERSTANDING OF WORDS

- The study of *meaning* and *context* should be central to linguistics
- Exploiting the context-dependent nature of words
- Language patterns cannot be accounted for in terms of a single system
- The *collocation*, gives enough clue to understand a word and its meaning
- *No study of meaning apart from context can be taken seriously*[2]

[2] Firth, J. R. 1957

POLYSEMOUS WORD - BANK

Synset('bank.n.01')	sloping land (especially the slope beside a body of water)
Synset('depository-financial-institution.n.01')	a financial institution that accepts deposits and channels the money into lending activities
Synset('bank.n.03')	a long ridge or pile
Synset('bank.n.10')	a flight maneuver; aircraft tips laterally about its longitudinal axis (especially in turning)
Synset('trust.v.01')	have confidence or faith in

POLYSEMOUS WORD - PROGRAM

Synset('plan.n.01')	a series of steps to be carried out or goals to be accomplished
Synset('program.n.02')	a system of projects or services intended to meet a public need
Synset('broadcast.n.02')	a radio or television show
Synset('platform.n.02')	a document stating the aims and principles of a political party
Synset('program.n.05')	an announcement of the events that will occur as part of a theatrical or sporting event
Synset('course_of_study.n.01')	an integrated course of academic studies
Synset('program.n.07')	(computer science) a sequence of instructions that a computer can interpret and execute
Synset('program.n.08')	a performance (or series of performances) at a public presentation
Synset('program.v.01')	arrange a program of or for
Synset('program.v.02')	write a computer program

SYNONYMS

small.a.01	['small', 'little']
minor.s.10	['minor', 'modest', 'small', 'small-scale', 'pocket-size', 'pocket-sized']
humble.s.01	['humble', 'low', 'lowly', 'modest', 'small']
little.s.07	['little', 'minuscule', 'small']
belittled.s.01	['belittled', 'diminished', 'small']
potent.a.03	['potent', 'strong', 'stiff']
impregnable.s.01	['impregnable', 'inviolable', 'secure', 'strong', 'unassailable', 'hard'] He has such an impregnable defense (Cricket-Very hard to find the gap between the bat and the pad)
solid.s.07	['solid', 'strong', 'substantial']
strong.s.09	['strong', 'warm']
firm.s.03	['firm', 'strong'] - firm grasp of fundamentals

UNDERSTANDING A WORD FROM ITS CONTEXT

The view from the top of the mountain was
The view from the summit was
La vue du sommet de la montagne *é*tait
Mtazamo wa juu wa mlima huo ulikuwa

awesome/(*impressionnante*, *impressionnant*)
breathtaking
amazing, அற்புதமான/അത്ഭുതകരമായ/ಅದ್ಭುತ/అద్భుతమైన
stunning/(*superbe*)
astounding अद्भुत/চমকপ্রদ
astonishing
awe-inspiring
extraordinary
incredible/(*incroyable*)
unbelievable
magnificent शानदार/ഗംഭീരമായ/ભવ્ય
wonderful/(*ajabu*)
spectacular
remarkable/(*yakuvutia*)

SEMANTICALLY CONNECTED VECTORS

- Identify a model that enumerates the relationships between terms
- Identify a model that tries to put similar items closer to each other in some space or structure
- Build a model that discovers/uncovers the semantic similarity between words and documents in the latent semantic domain
- Develop a distributed word vectors or dense vectors that captures the linear combination of word vectors in the transformed domain
- Transform the term-document space into a synonymy and a semantic space

METHODS TO CREATE WORD VECTORS

- Brown clustering - statistical algorithms for assigning words to classes based on the frequency of their co-occurrence with other words
- Hyperspace Analogue to Language - HAL
- Correlated Occurrence Analogue to Lexical Semantic - COALS
- Latent Semantic Analysis or Latent Semantic Indexing
- Global Vectors - GloVe
- Neural networks using skip grams and CBOW
  - CBOW - uses surrounding words to predict the center of words
  - Skip grams use center of words to predict the surrounding words

WORD SIMILARITY

- Sparse vectors are too long and not very convenient as features machine learning
- Abstracts more than just frequency counts
- It captures neighborhood words that are connected by synonyms

WORD VECTOR EXAMPLES

Using 6B words and 100 element word vector

Similar words for apple

'apple', 0
'iphone', 0.266
'ipad', 0.287
'apples', 0.356
'blackberry', 0.361
'ipod', 0.365
'macbook', 0.383
'mac', 0.391
'android', 0.391
'google', 0.395
'microsoft', 0.418
'ios', 0.433
'iphones', 0.445

Similar words for - american

'american', 0
'america', 0.255
'americans', 0.312
'u.s.', 0.320
'british', 0.323
'canadian', 0.329
'history', 0.356
'national', 0.364
'african', 0.374
'society', 0.375
'states', 0.386
'european', 0.387
'world', 0.394
'nation', 0.399

VECTOR DIFFERENCE BETWEEN TWO WORDS

Word vectors closer to the operation $\boxed{vec(apple) - vec(iphone)}$

```
('raisin', 0.5744591153088133)
('pecan', 0.576061737414159)
('cranberry', 0.584001617225404)
('butternut', 0.5882322018694753)
('cider', 0.5910795032086132)
('apricot', 0.6036644437522422)
('tomato', 0.6073715970323961)
('rosemary', 0.6150986936477657)
('rhubarb', 0.6157884153793192)
('feta', 0.6183016129045151)
('apples', 0.6226003361980218)
('avocado', 0.6235366677962004)
('fennel', 0.6306016018912576)
('chutney', 0.6312524337590703)
('spiced', 0.6327632200841328)
```