# Distributional Semantics

LING 571 — Deep Processing Methods in NLP

November 8, 2021

Shane Steinert-Threlkeld

# Announcements

- In tarball:
  - Submit source code!
  - Beyond that, *only what we ask for* (e.g. not a copy of toy data, etc)
- Broken record, but:
  - Full paths to python binary!
  - Test your code on patas
- HW5: back to working individually on assignments, NOT pairs

# 571 in the News

# Underspecification, 1yr Anniversary Edition

“Lawyers News Conference Four Seasons, Philadelphia, 11 a.m.,” —DJT

mber 09, 2020 | **Today's Paper**

**The Philadelphia Inquirer**

Unlimited Acc

RTS BUSINESS OPINION ELECTIONS ENTERTAINMENT LIFE FOOD HEALTH REAL ESTATE OBITUARIES JOBS

**No, not that Four Seasons. How Team Trump’s news conference ended up at a Northeast Philly landscaping firm.**


“Four Seasons Total Landscaping”

# Distributional Similarity

# Distributional Similarity

- "You shall know a word by the company it keeps!" (*Firth, 1957*)

# Distributional Similarity

- "You shall know a word by the company it keeps!" (*Firth, 1957*)
  - A bottle of ***tezgüino*** is on the table.

# Distributional Similarity

- "You shall know a word by the company it keeps!" (*Firth, 1957*)
  - A bottle of ***tezgüino*** is on the table.
  - Everybody likes ***tezgüino***.

# Distributional Similarity

- "You shall know a word by the company it keeps!" (*Firth, 1957*)
  - A bottle of ***tezgüino*** is on the table.
  - Everybody likes ***tezgüino***.
  - ***Tezgüino*** makes you drunk.

# Distributional Similarity

- "You shall know a word by the company it keeps!" (*Firth, 1957*)
  - A bottle of ***tezgüino*** is on the table.
  - Everybody likes ***tezgüino***.
  - ***Tezgüino*** makes you drunk.
  - We make ***tezgüino*** from corn.

# Distributional Similarity

- "You shall know a word by the company it keeps!" ([Firth, 1957])
  - A bottle of ***tezgüino*** is on the table.
  - Everybody likes ***tezgüino***.
  - ***Tezgüino*** makes you drunk.
  - We make ***tezgüino*** from corn.
- Tezguino; corn-based alcoholic beverage. (From [Lin, 1998a])

# Distributional Similarity

- How can we represent the “company” of a word?

# Distributional Similarity

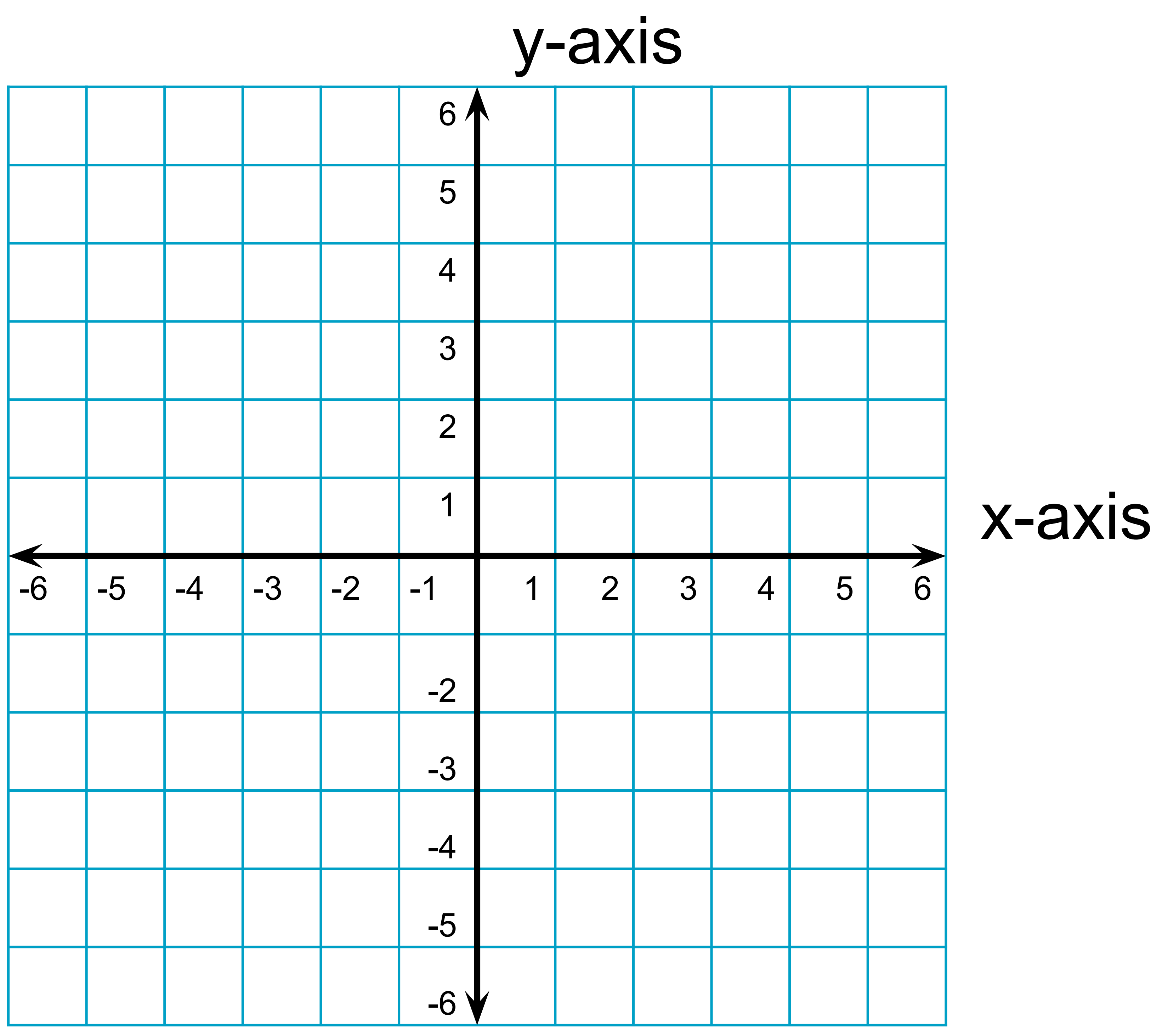
- How can we represent the “company” of a word?
- How can we make similar words have similar representations?

# Vectors: A Refresher

- A vector is a list of numbers
- Each number can be thought of as representing a "dimension"

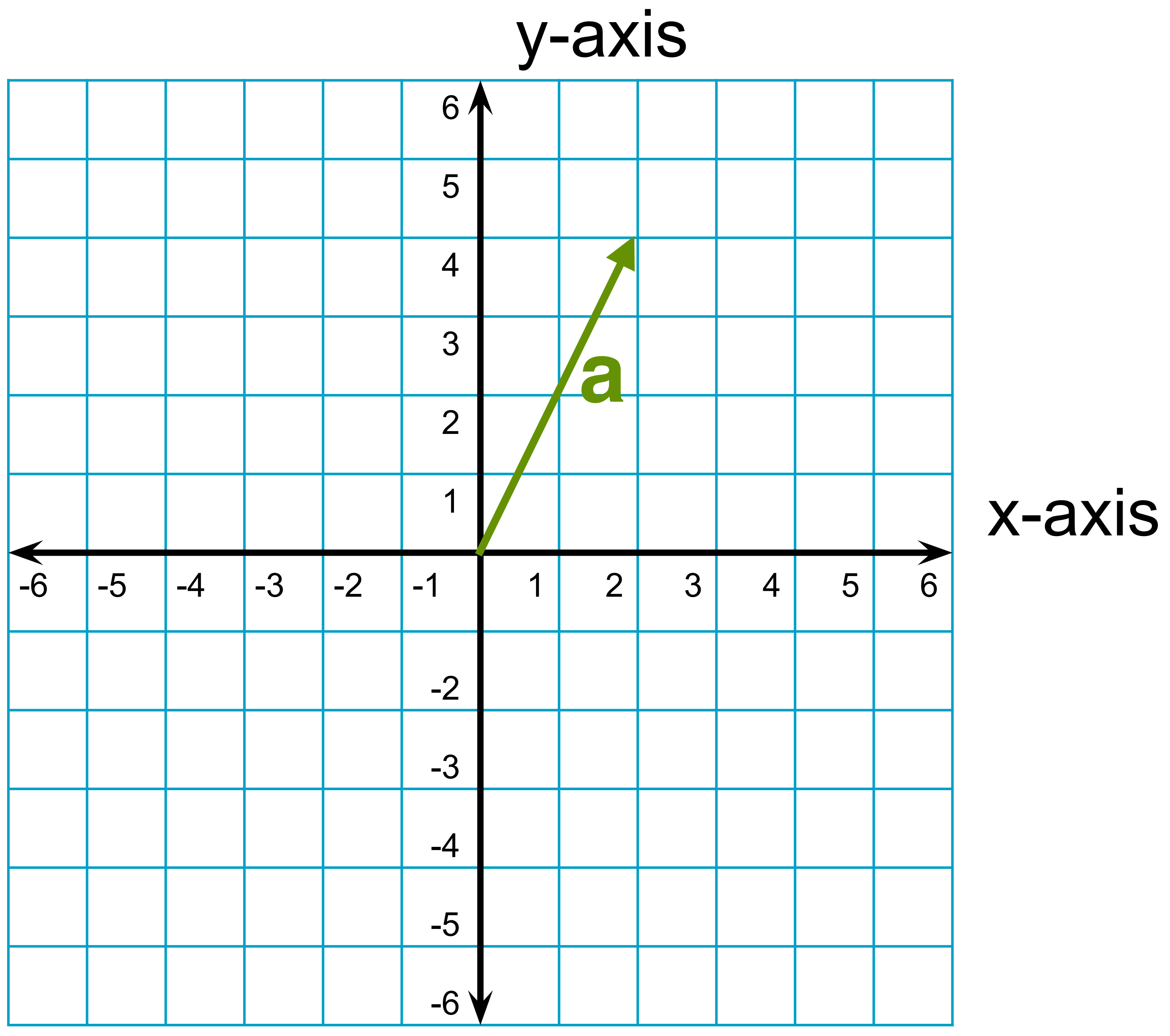
# Vectors: A Refresher

- A vector is a list of numbers
- Each number can be thought of as representing a “dimension”



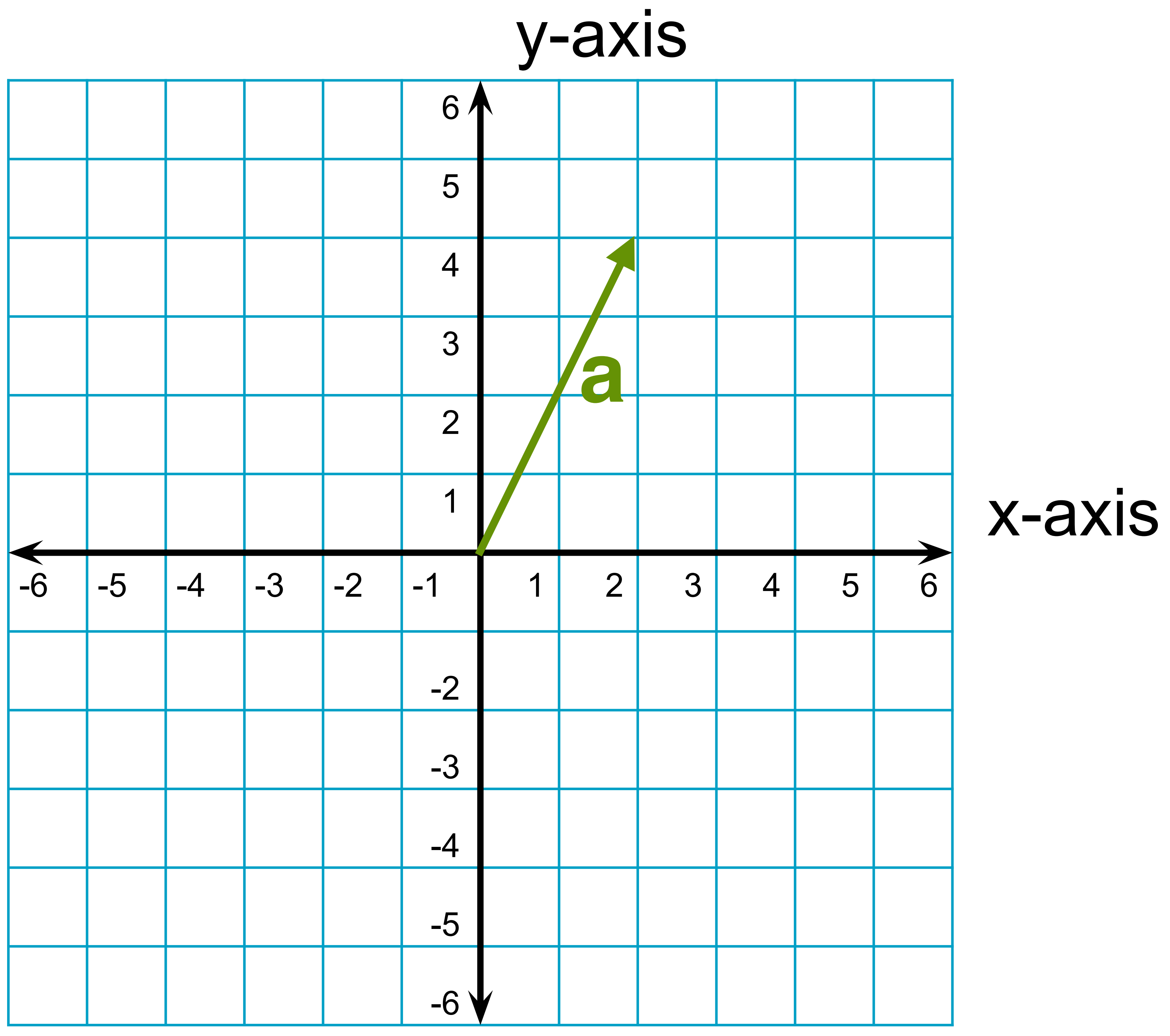
# Vectors: A Refresher

- A vector is a list of numbers
- Each number can be thought of as representing a "dimension"
  - $\vec{a} = \langle 2,4 \rangle$



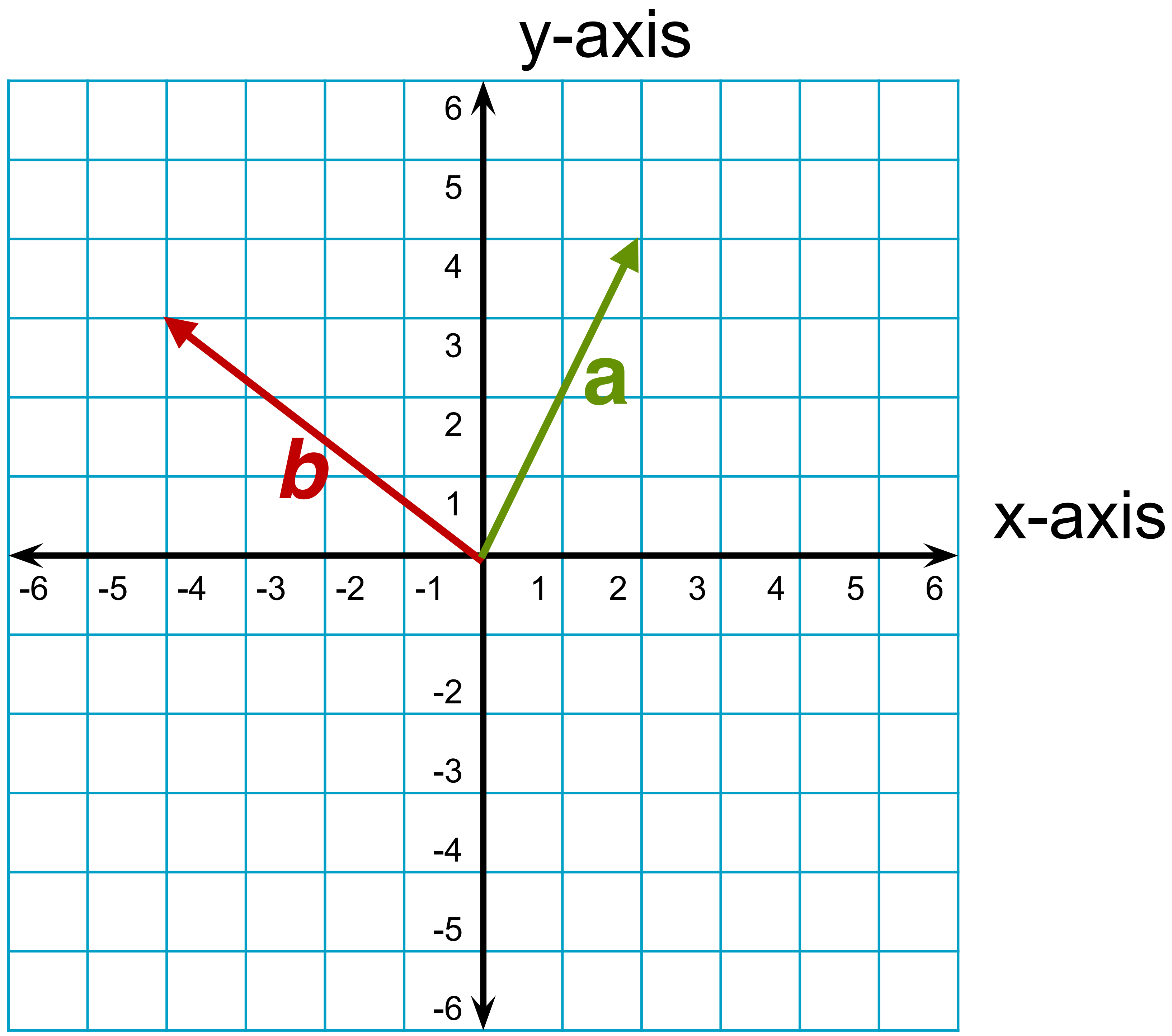
# Vectors: A Refresher

- A vector is a list of numbers
- Each number can be thought of as representing a “dimension”
  - $\vec{a} = \langle 2,4 \rangle$
  - $\vec{b} = \langle -4,3 \rangle$



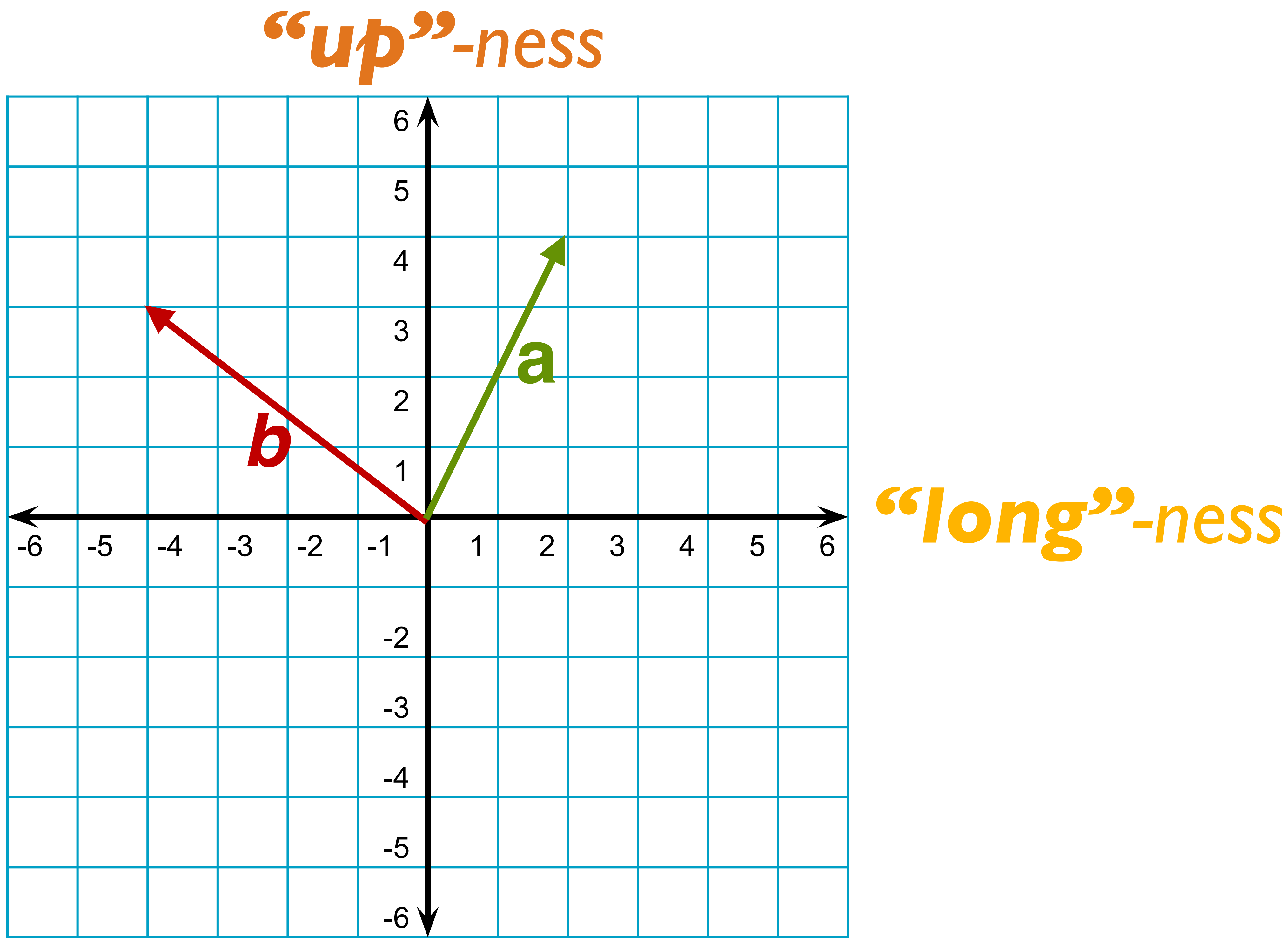
# Vectors: A Refresher

- A vector is a list of numbers
- Each number can be thought of as representing a “dimension”
  - $\vec{a} = \langle 2,4 \rangle$
  - $\vec{b} = \langle -4,3 \rangle$
- What if we thought of each dimension as “quantity” of a word, rather than an arbitrary dimension?



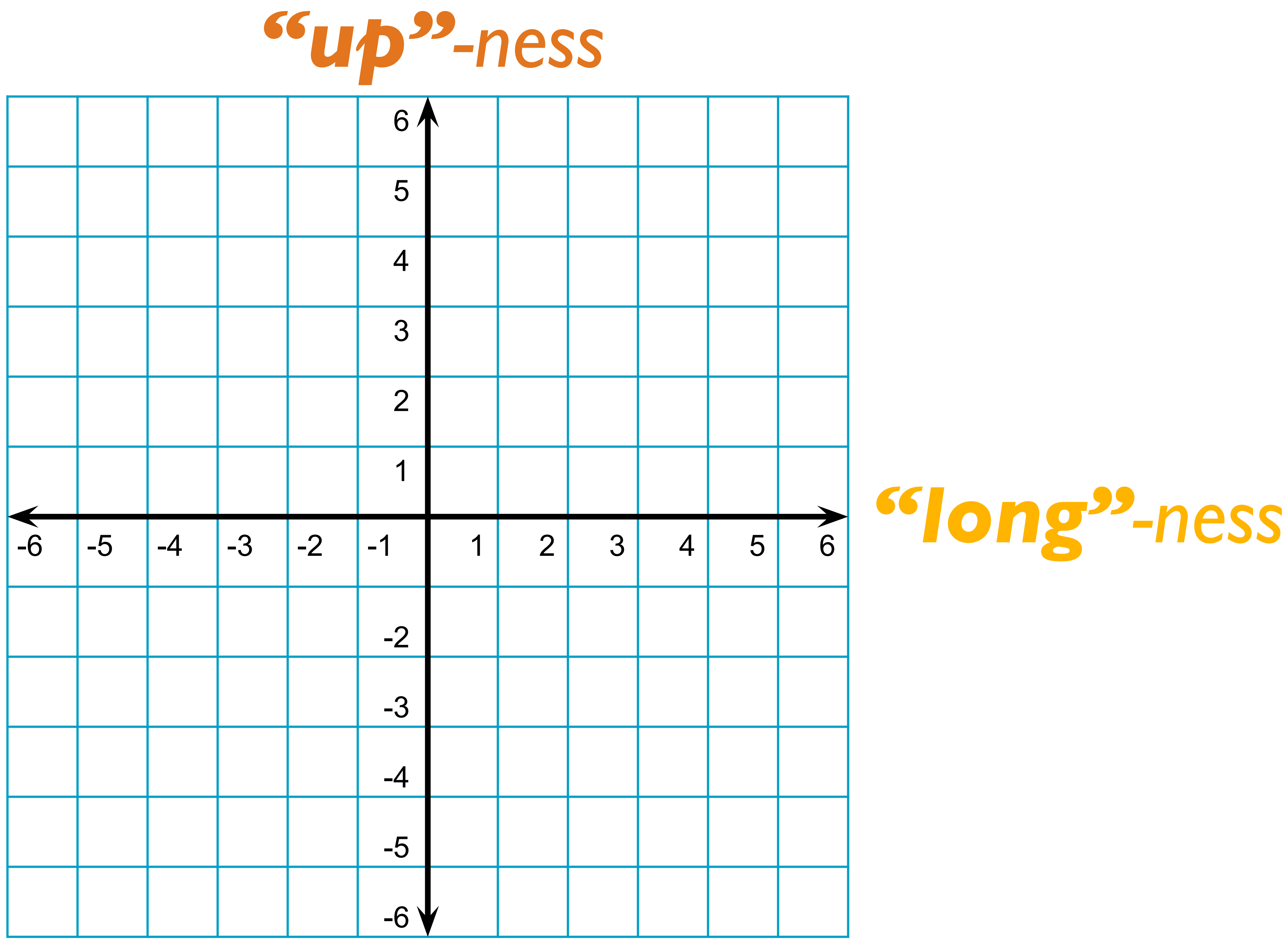
# Vectors: A Refresher

- A vector is a list of numbers
- Each number can be thought of as representing a “dimension”
  - $\vec{a} = \langle 2,4 \rangle$
  - $\vec{b} = \langle -4,3 \rangle$
- What if we thought of each dimension as “quantity” of a word, rather than an arbitrary dimension?



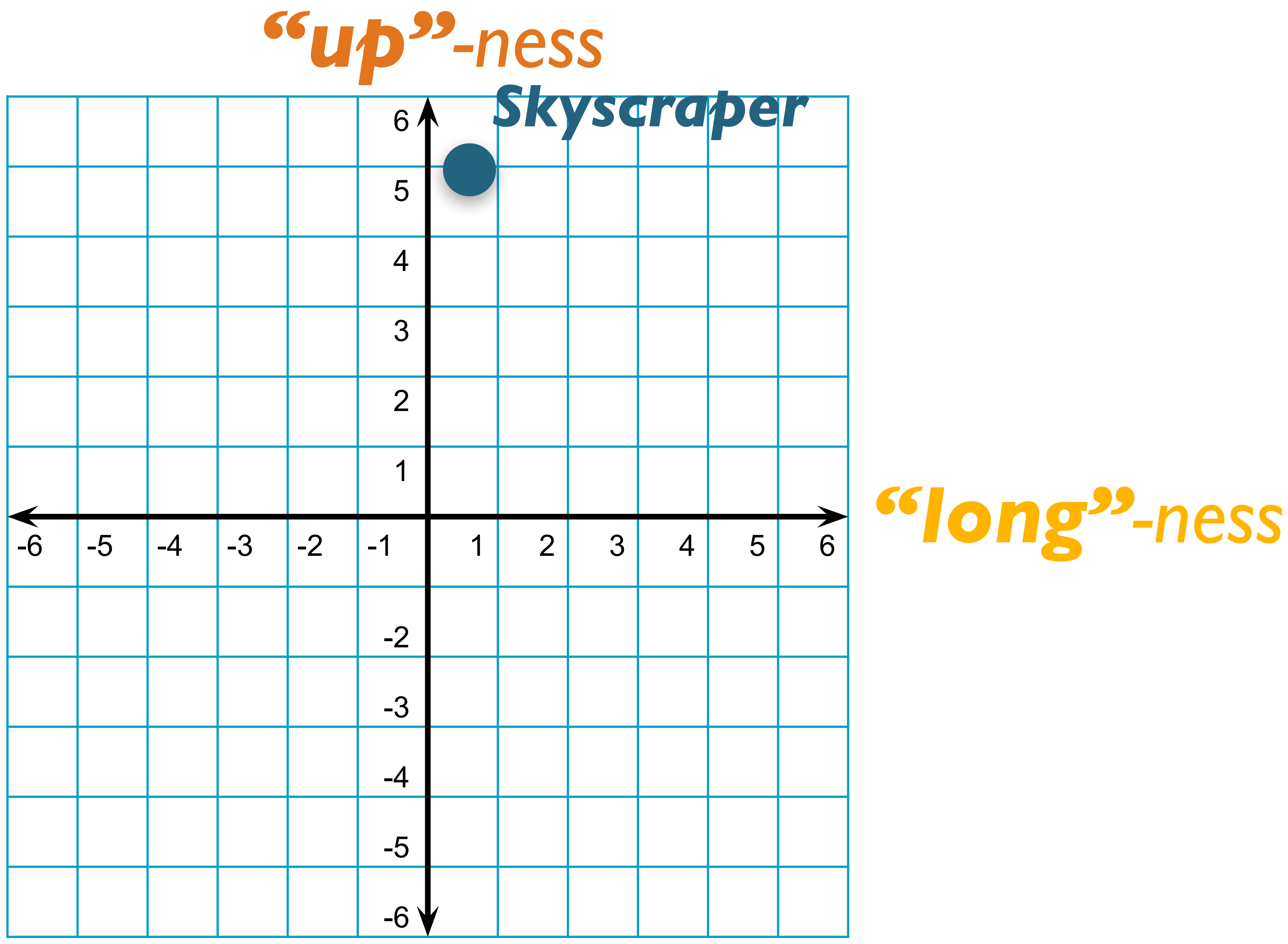
# Vectors: A Refresher

- A vector is a list of numbers
- Each number can be thought of as representing a “dimension”
  - $\vec{a} = \langle 2,4 \rangle$
  - $\vec{b} = \langle -4,3 \rangle$
- What if we thought of each dimension as “quantity” of a word, rather than an arbitrary dimension?



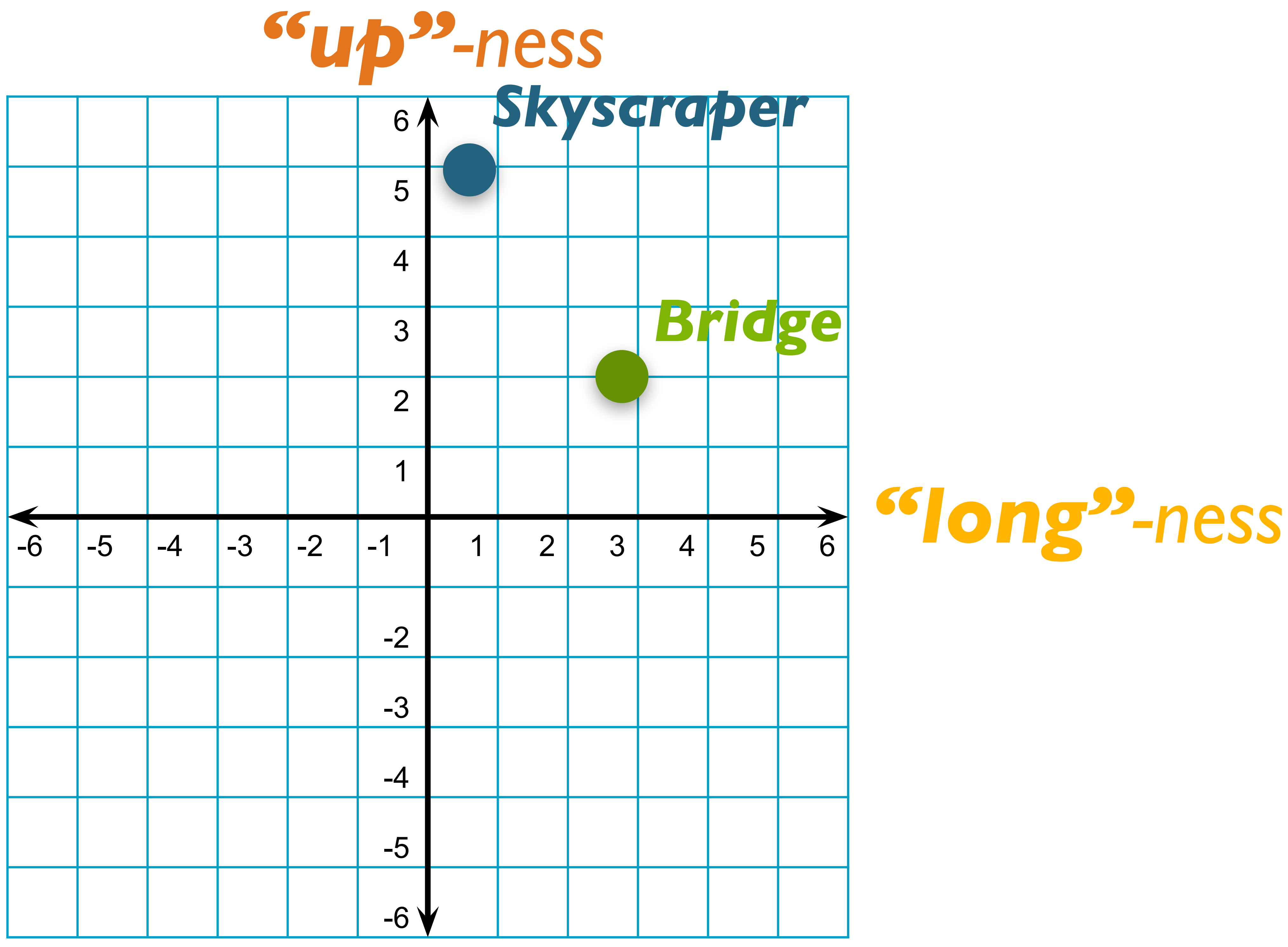
# Vectors: A Refresher

- A vector is a list of numbers
- Each number can be thought of as representing a “dimension”
  - $\vec{a} = \langle 2,4 \rangle$
  - $\vec{b} = \langle -4,3 \rangle$
- What if we thought of each dimension as “quantity” of a word, rather than an arbitrary dimension?



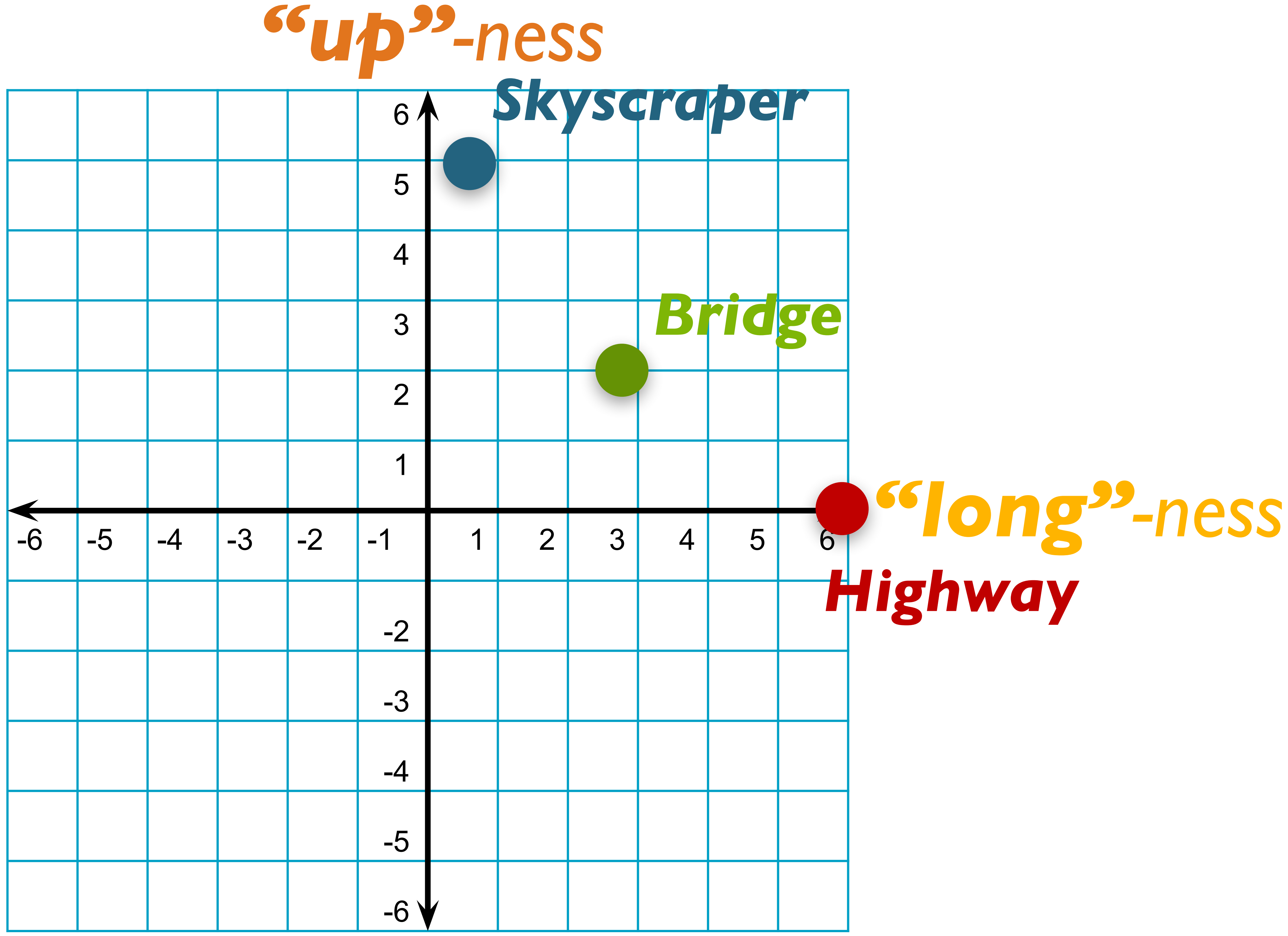
# Vectors: A Refresher

- A vector is a list of numbers
- Each number can be thought of as representing a “dimension”
  - $\vec{a} = \langle 2,4 \rangle$
  - $\vec{b} = \langle -4,3 \rangle$
- What if we thought of each dimension as “quantity” of a word, rather than an arbitrary dimension?



# Vectors: A Refresher

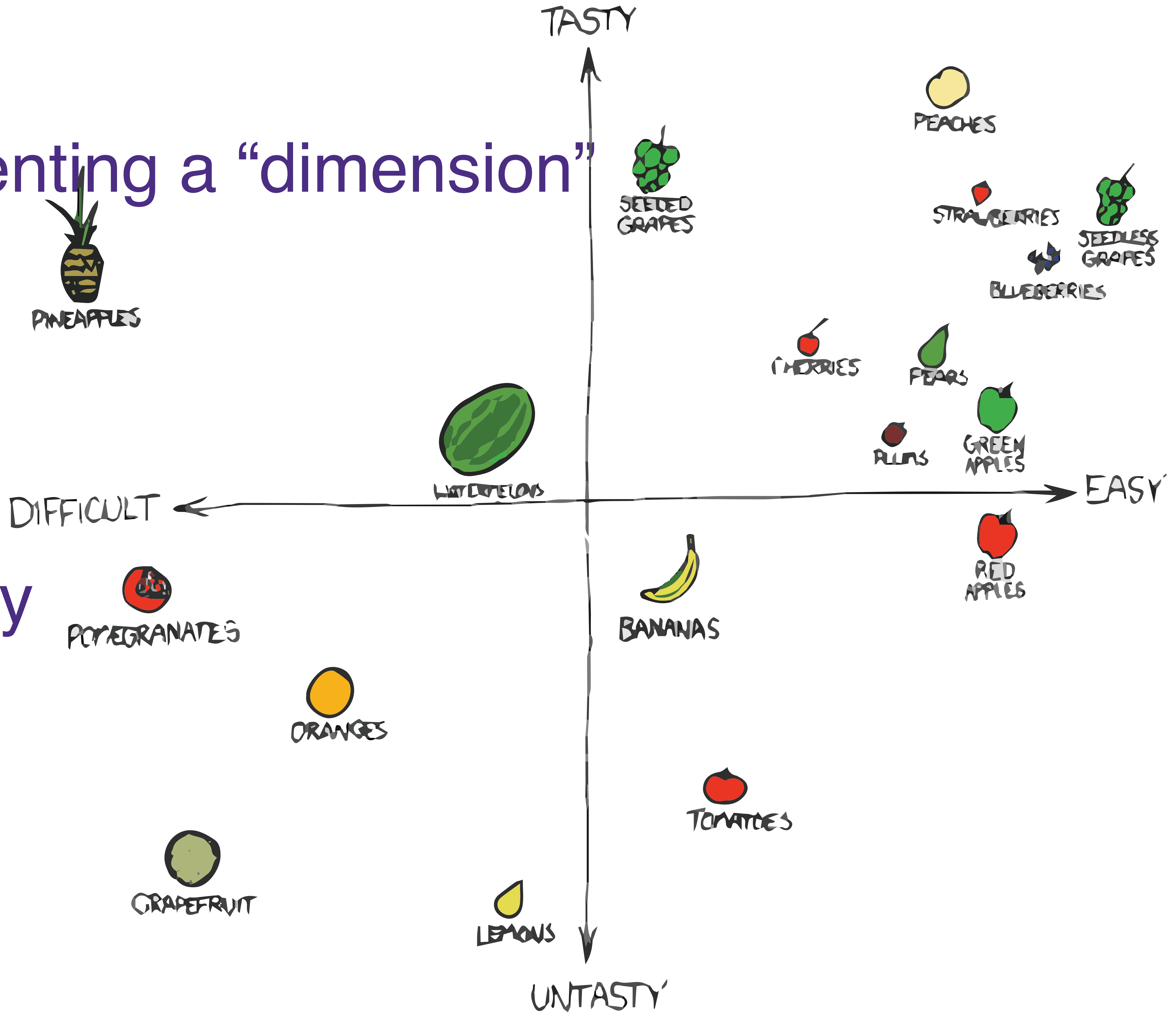
- A vector is a list of numbers
- Each number can be thought of as representing a "dimension"
  - $\vec{a} = \langle 2,4 \rangle$
  - $\vec{b} = \langle -4,3 \rangle$
- What if we thought of each dimension as "quantity" of a word, rather than an arbitrary dimension?



# Vectors: A Refresher

xkcd.com/388

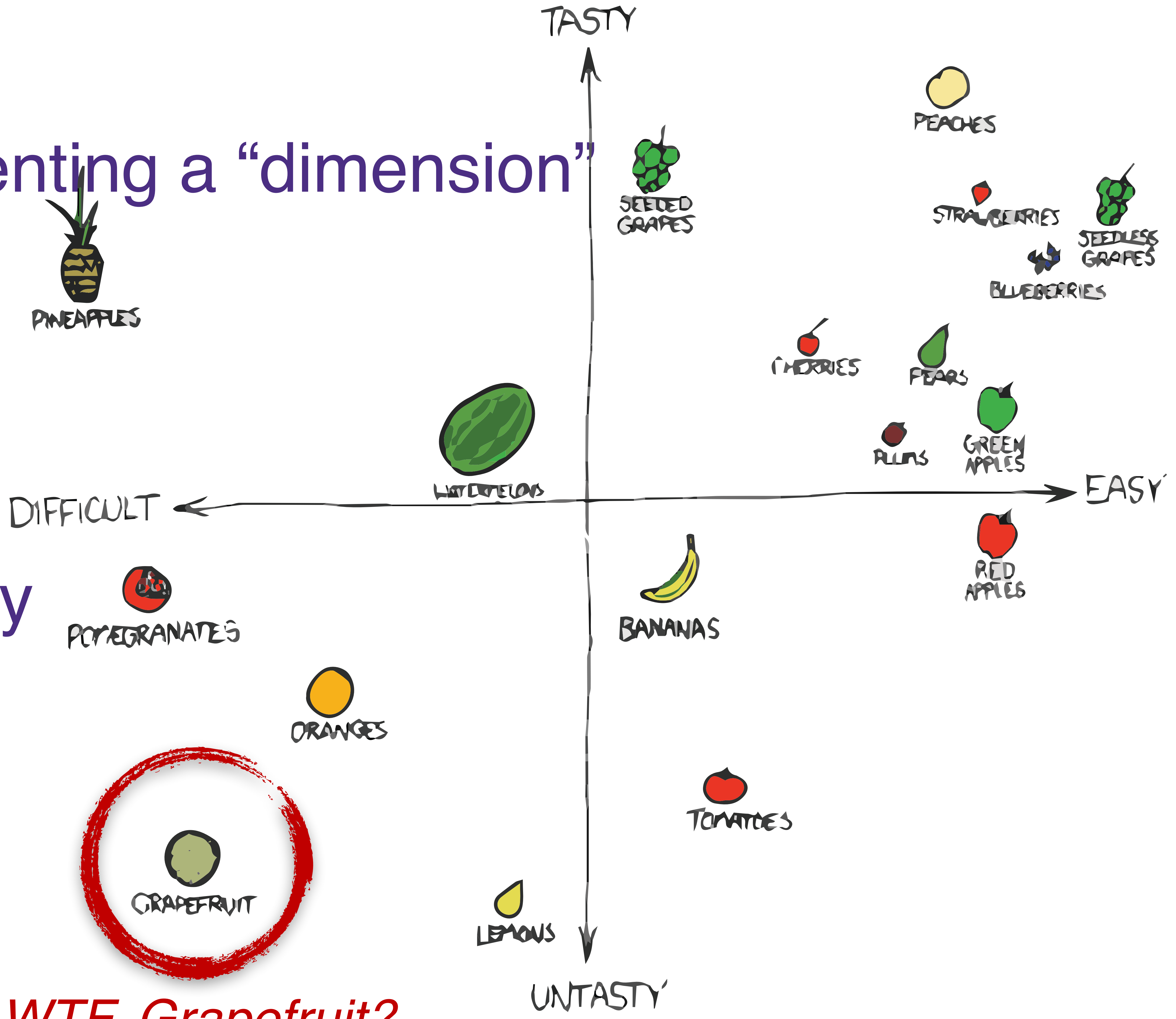
- A vector is a list of numbers
- Each number can be thought of as representing a “dimension”
  - $\vec{a} = \langle 2,4 \rangle$
  - $\vec{b} = \langle -4,3 \rangle$
- What if we thought of each dimension as “quantity” of a word, rather than an arbitrary dimension?



# Vectors: A Refresher

- A vector is a list of numbers
- Each number can be thought of as representing a "dimension"
  - $\vec{\mathbf{a}}=\langle 2,4\rangle$
  - $\vec{\mathbf{b}}=\langle -4,3\rangle$
- What if we thought of each dimension as "quantity" of a word, rather than an arbitrary dimension?

xkcd.com/388




*WTF, Grapefruit?*

# Vector Space: Documents

- We can represent documents as vectors, with each dimension being a count of a particular word

Shakespeare Plays x Counts of Words

| | As You Like It | Twelfth Night | Julius Caesar | Henry V |
|---|---|---|---|---|
| **battle** | 1 | 1 | 8 | 15 |
| **soldier** | 2 | 2 | 12 | 36 |
| **fool** | 37 | 58 | 1 | 5 |
| **clown** | 5 | 117 | 0 | 0 |

# Vector Space: Documents

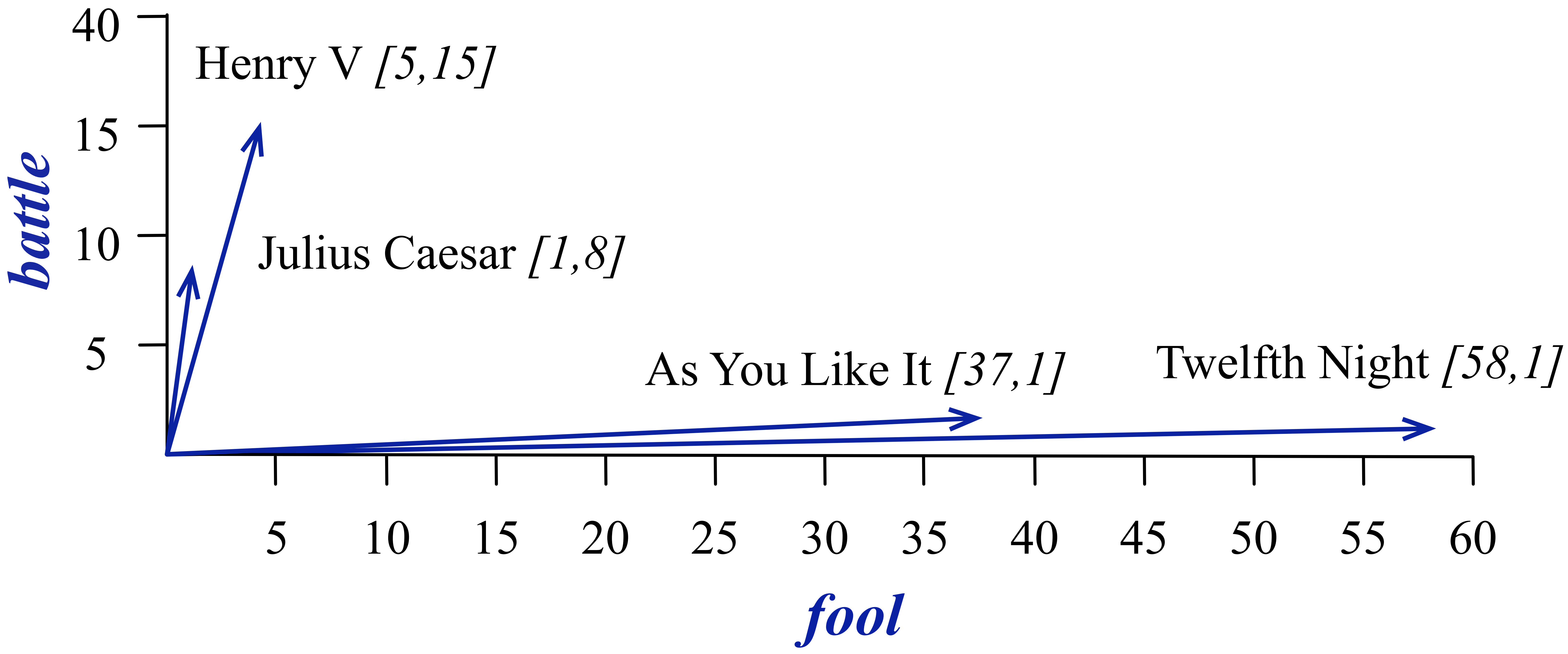
- We can represent documents as vectors, with each dimension being a count of a particular word

Shakespeare Plays x Counts of Words

| | *As You Like It* | *Twelfth Night* | *Julius Caesar* | *Henry V* |
|---|---|---|---|---|
| **battle** | 1 | 1 | 8 | 15 |
| **soldier** | 2 | 2 | 12 | 36 |
| **fool** | 37 | 58 | 1 | 5 |
| **clown** | 5 | 117 | 0 | 0 |

# Vector Space: Documents

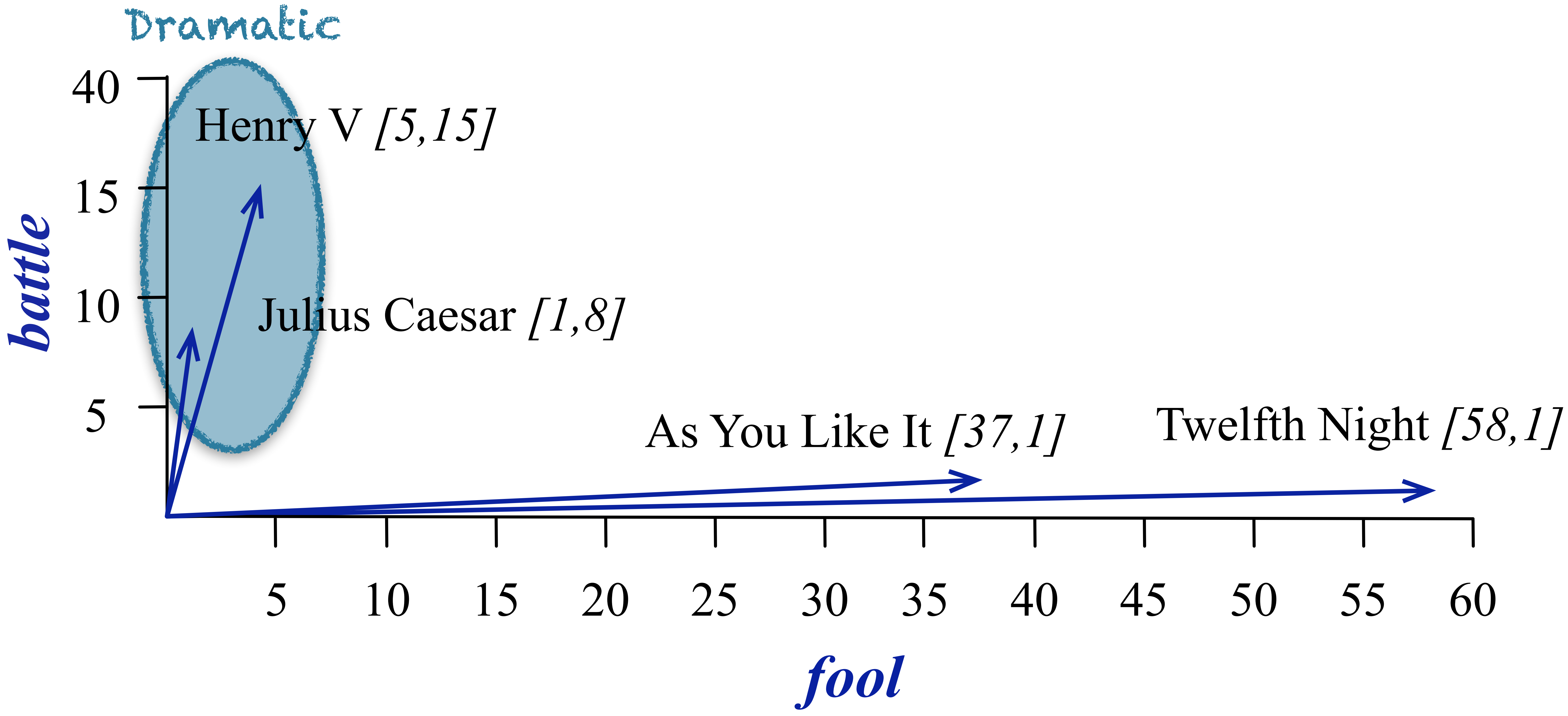
- We can represent documents as vectors, with each dimension being a count of a particular word




J&M 3rd ed, 6.3.1 [link]

# Vector Space: Documents

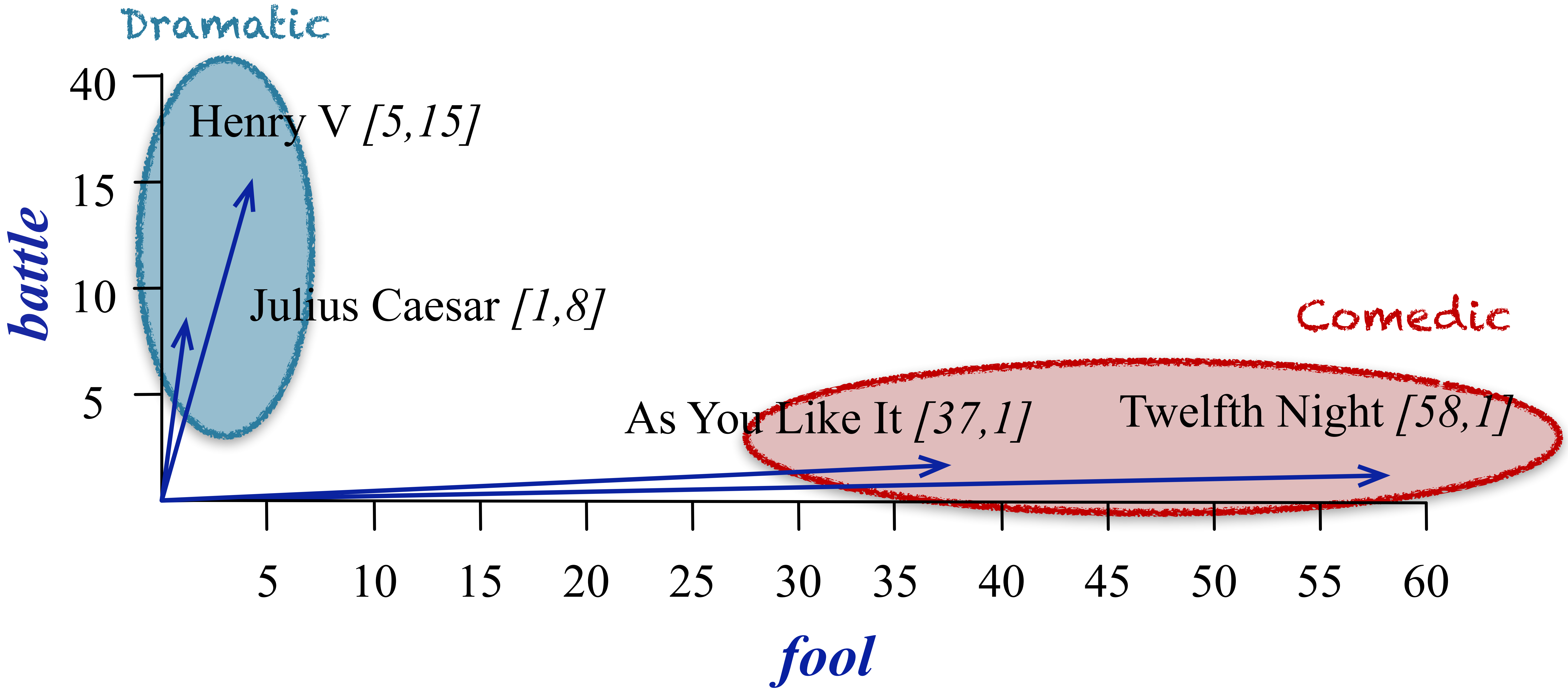
- We can represent documents as vectors, with each dimension being a count of a particular word




J&M 3rd ed, 6.3.1 [link]

# Vector Space: Documents

- We can represent documents as vectors, with each dimension being a count of a particular word




J&M 3rd ed, 6.3.1 [link]

# Vector Space: Words

- Find thematic clusters for ***words*** based on words that occur ***around*** them.

# Distributional Similarity

- Represent ‘company’ of word such that similar words will have similar representations
  - ‘Company’ = context

# Distributional Similarity

- Represent ‘company’ of word such that similar words will have similar representations
  - ‘Company’ = context
- Word represented by context feature vector
  - Many alternatives for vector

# Distributional Similarity

- Represent 'company' of word such that similar words will have similar representations
  - 'Company' = context
- Word represented by context feature vector
  - Many alternatives for vector
- Initial representation:
  - 'Bag of words' feature vector
  - Feature vector length $N$, where $N$ is size of vocabulary
    - $f_i$+=1 if $word_i$ within window size $w$ of $word$

There are more kinds of **plants** and animals in the rainforests than anywhere else on Earth. Over half of the millions of known species of **plants** and animals live in the rainforest. Many are found nowhere else. There are even **plants** and animals in the rainforest that we have not yet discovered.

***Biological Example***

The Paulus company was founded in 1938. Since those days the product range has been the subject of constant expansions and is brought up continuously to correspond with the state of the art. We're engineering, manufacturing and commissioning world-wide ready-to-run **plants** packed with our comprehensive know-how. Our Product Range includes pneumatic conveying systems for carbon, carbide, sand, lime and many others. We use reagent injection in molten metal for the…

***Industrial Example***

Label the First Use of "Plant"

-1 +1

There are more kinds of **plants** and animals in the rainforests than anywhere else on Earth. Over half of the millions of known species of **plants** and animals live in the rainforest. Many are found nowhere else. There are even **plants** and animals in the rainforest that we have not yet discovered.

plant: (and: 1, of: 1)

-2 +2

There are more kinds of **plants** and animals in the rainforests than anywhere else on Earth. Over half of the millions of known species of **plants** and animals live in the rainforest. Many are found nowhere else. There are even **plants** and animals in the rainforest that we have not yet discovered.

plant: (and: 1, animal: 1, kind: 1, of: 1)

-3 +3

There are more kinds of **plants** and animals in the rainforests than anywhere else on Earth. Over half of the millions of known species of **plants** and animals live in the rainforest. Many are found nowhere else. There are even **plants** and animals in the rainforest that we have not yet discovered.

plant: (and: 1, animal: 1, in: 1, kind: 1, more: 1, of: 1)

-4 +4

There are more kinds of **plants** and animals in the rainforests than anywhere else on Earth. Over half of the millions of known species of **plants** and animals live in the rainforest. Many are found nowhere else. There are even **plants** and animals in the rainforest that we have not yet discovered.

plant: (and: 1, animal: 1, are: 1, in: 1, kind: 1, more: 1, of: 1, the: 1)

-5 +5

There are more kinds of **plants** and animals in the rainforests than anywhere else on Earth. Over half of the millions of known species of **plants** and animals live in the rainforest. Many are found nowhere else. There are even **plants** and animals in the rainforest that we have not yet discovered.

plant: (and: 1, animal: 1, are: 1, in: 1, kind: 1, more: 1, of: 1, rainforest: 1, the: 1, there: 1)

There are more kinds of **plants** and animals in the rainforests than anywhere else on Earth. Over half of the millions of known species of **plants** and animals live in the rainforest. Many are found nowhere else. There are even **plants** and animals in the rainforest that we have not yet discovered.

plant: (and: 1, animal: 2, are: 1, in: 1, kind: 1, more: 1, of: 1, rainforest: 1, the: 1, there: 1, species: 1)

There are more kinds of **plants** and animals in the rainforests than anywhere else on Earth. Over half of the millions of known species of **plants** and animals live in the rainforest. Many are found nowhere else. There are even **plants** and animals in the rainforest that we have not yet discovered.

plant: (and: 1, animal: 3, are: 2, in: 1, kind: 1, more: 1, of: 1, rainforest: 1, the: 1, there: 1, species: 1)

There are more kinds of **plants** and animals in the rainforests than anywhere else on Earth. Over half of the millions of known species of **plants** and animals live in the rainforest. Many are found nowhere else. There are even **plants** and animals in the rainforest that we have not yet discovered.

plant: (and: 1, animal: 3, are: 2, in: 1, kind: 1, more: 1, of: 1, rainforest: 2, the: 1, there: 1, species: 1, nowhere: 1)

There are more kinds of **plants** and animals in the rainforests than anywhere else on Earth. Over half of the millions of known species of **plants** and animals live in the rainforest. Many are found nowhere else. There are even **plants** and animals in the rainforest that we have not yet discovered.

plant: (and: 1, animal: 3, are: 2, in: 1, kind: 1, more: 1, of: 1, rainforest: 2, the: 1, there: 1, species: 1, nowhere: 1)

# Context Feature Vector

| | aardvark | … | computer | data | pinch | result | sugar |
|---|---|---|---|---|---|---|---|
| apricot | 0 | … | 0 | 0 | 1 | 0 | 1 |
| pineapple | 0 | … | 0 | 0 | 1 | 0 | 1 |
| digital | 0 | … | 2 | 1 | 0 | 1 | 0 |
| information | 0 | … | 1 | 6 | 0 | 4 | 0 |

# Distributional Similarity Questions

**What is the right neighborhood?**

How should we weight the features?

How can we compute the similarity between vectors?

# Similarity "Neighborhood"

1. Fixed window
   - How many words in the neighborhood?
     - +/– 500 words: 'topical context'
     - +/– 1 or 2 words: collocations, predicate-argument

2. Only words in some grammatical relation (Hindle, 1990)
   - Parse text (dependency)
     - Include *subj-verb*; *verb-obj*; *adj-mod*
     - $N \times R$ vector: word × relation

# Similarity "Neighborhood": Fixed Window

# Similarity "Neighborhood": Fixed Window

- Same corpus, different windows
  - British National Corpus (BNC)
  - Nearest neighbors of "dog"

# Similarity "Neighborhood": Fixed Window

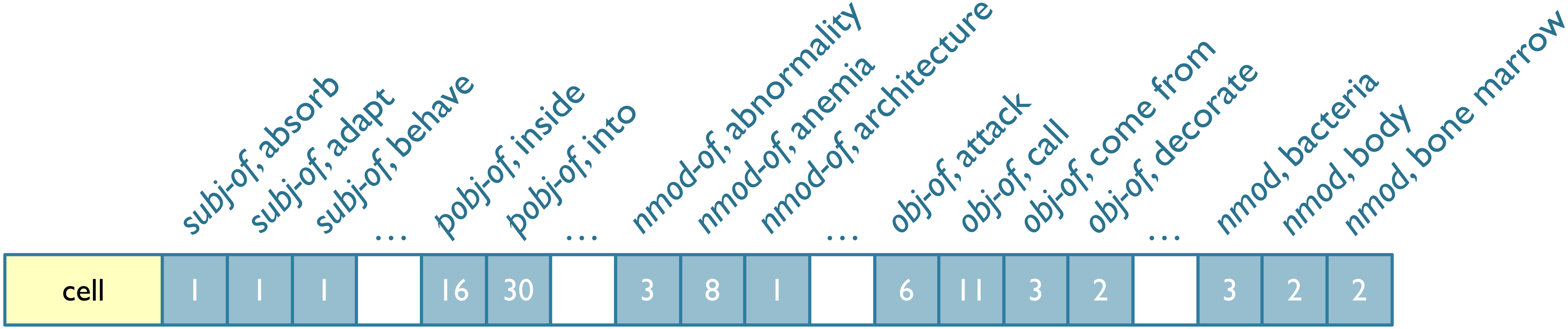
- Same corpus, different windows
  - British National Corpus (BNC)
  - Nearest neighbors of "dog"
- 2-word window:
  - *Cat, horse, fox, pet, rabbit, pig, animal, mongrel, sheep, pigeon*

# Similarity "Neighborhood": Fixed Window

- Same corpus, different windows
  - British National Corpus (BNC)
  - Nearest neighbors of "dog"
- 2-word window:
  - *Cat, horse, fox, pet, rabbit, pig, animal, mongrel, sheep, pigeon*
- 30-word window:
  - *Kennel, puppy, pet, terrier, Rottweiler, canine, cat, to bark, Alsatian*

# Similarity "Neighborhood": Grammatical Relations

- Build a vector from dependency triples: (Lin, 1998)
  - `(w1 dep_rel w2)`




*Dependency vector for "cell," counts from 64M word corpus.*

# "Neighborhood": Window vs. Grammatical Relations

- Grammatical relations:
  - Richer representation
  - Much more POS information
- Window:
  - Only need text!
  - Scales very, very well.  (Maybe too well.)
  - Adding explicit supervision from parsers often doesn't help dramatically

# Distributional Similarity Questions

What is the right neighborhood?

**How should we weight the features?**

How can we compute the similarity between vectors?

# Weighting Features: Binary vs. Nonbinary?

- Binary?
  - Minimally informative
  - Can't capture intuition that frequent features more indicative of relationship.
- Frequency
  - Or rather, probability: $assoc_{prob}(w, f) = P(f|w)$
  - …but how do we know which words are informative?
  - *the, it, they* — not likely to help differentiate target word

# Weighting Features: Pointwise Mutual Information

- PMI is measure of how often two events $x$ and $y$ occur, vs. expected frequency if they were independent (Fano, 1961)

$$PMI(x, y) = \log_2 \frac{P(x, y)}{P(x) \cdot P(y)}$$

# Weighting Features: Pointwise Mutual Information

- We can formulate for word/feature occurrence: $assoc_{PMI}(w,f) = \log_2 \frac{P(w,f)}{P(w) \cdot P(f)}$
- Generally only use positive values
  - Negatives inaccurate unless corpus huge
- Can also rescale/smooth context values

# Weighting Features: (Positive) Pointwise Mutual Information

$$assoc_{PMI}(w,f) = \log_2 \frac{P(w,f)}{P(w) \cdot P(f)}$$

# Weighting Features: (Positive) Pointwise Mutual Information

$$assoc_{PMI}(w,f) = \log_2 \frac{P(w,f)}{P(w) \cdot P(f)}$$

$$p_{ij} = \frac{f_{ij}}{\sum_{i=1}^{W} \sum_{j=1}^{C} f_{ij}}$$

probability of feature $f$ relating $i$ to $j$

$$p_{i*} = \frac{\sum_{j=1}^{C} f_{ij}}{\sum_{i=1}^{W} \sum_{j=1}^{C} f_{ij}}$$

probability of feature $f$ relating $i$ to *anything*

$$p_{*j} = \frac{\sum_{i=1}^{W} f_{ij}}{\sum_{i=1}^{W} \sum_{j=1}^{C} f_{ij}}$$

probability of feature $f$ relating *anything* to $j$

# Weighting Features: (Positive) Pointwise Mutual Information

$$assoc_{PMI}(w,f) = \log_2 \frac{P(w,f)}{P(w) \cdot P(f)}$$

$$p_{ij} = \frac{f_{ij}}{\sum_{i=1}^{W} \sum_{j=1}^{C} f_{ij}}$$

probability of feature $f$ relating $i$ to $j$

$$p_{i*} = \frac{\sum_{j=1}^{C} f_{ij}}{\sum_{i=1}^{W} \sum_{j=1}^{C} f_{ij}}$$

probability of feature $f$ relating $i$ to *anything*

$$p_{*j} = \frac{\sum_{i=1}^{W} f_{ij}}{\sum_{i=1}^{W} \sum_{j=1}^{C} f_{ij}}$$

probability of feature $f$ relating *anything* to $j$

$$PPMI_{ij} = \max(\log_2 \frac{p_{ij}}{p_{i*} \cdot p_{*j}}, 0)$$

Get (non-negative) ratio

# Weighting Features: (Positive) Pointwise Mutual Information

- For pure word co-occurrence, feature ***f*** is the colocated word.

# Weighting Features: Pointwise Mutual Information

- Total words (sum of whole table) $=$ **19**

| | aardvark | computer | data | pinch | result | sugar |
|---|---|---|---|---|---|---|
| **apricot** | 0 | 0 | 0 | 1 | 0 | 1 |
| **pineapple** | 0 | 0 | 0 | 1 | 0 | 1 |
| **digital** | 0 | 2 | 1 | 0 | 1 | 0 |
| **information** | 0 | 1 | 6 | 0 | 4 | 0 |

# Weighting Features: Pointwise Mutual Information

- Total words (sum of whole table) $= 19$
  - $P(w)$, where $w$ is *information* = **11**/19 = **.579**

| | aardvark | computer | data | pinch | result | sugar |
|---|---|---|---|---|---|---|
| apricot | 0 | 0 | 0 | 1 | 0 | 1 |
| pineapple | 0 | 0 | 0 | 1 | 0 | 1 |
| digital | 0 | 2 | 1 | 0 | 1 | 0 |
| information | 0 | 1 | 6 | 0 | 4 | 0 |

# Weighting Features:
# Pointwise Mutual Information

- Total words (sum of whole table) $= 19$
  - $P(w)$, where $w$ is *information* = 11/19 = .579
  - $P(f)$, where $f$ is *data* = **7/19 = .368**

| | aardvark | computer | data | pinch | result | sugar |
|---|---|---|---|---|---|---|
| **apricot** | 0 | 0 | 0 | 1 | 0 | 1 |
| **pineapple** | 0 | 0 | 0 | 1 | 0 | 1 |
| **digital** | 0 | 2 | 1 | 0 | 1 | 0 |
| **information** | 0 | 1 | 6 | 0 | 4 | 0 |

# Weighting Features: Pointwise Mutual Information

- Total words (sum of whole table) $=$ 19
  - $P(w)$, where $w$ is *information* = 11/19 = .579
  - $P(f)$, where $f$ is *data* = 7/19 = .368
  - $P(w,f)$, where $(w,f)$ is (*information*,*data*) = **6**/19 = **.316**

| | aardvark | computer | data | pinch | result | sugar |
|---|---|---|---|---|---|---|
| **apricot** | 0 | 0 | 0 | 1 | 0 | 1 |
| **pineapple** | 0 | 0 | 0 | 1 | 0 | 1 |
| **digital** | 0 | 2 | 1 | 0 | 1 | 0 |
| **information** | 0 | 1 | $\boxed{6}$ | 0 | 4 | 0 |

# Weighting Features: Pointwise Mutual Information

- Total words (sum of whole table) $=$ 19
  - $P(w)$, where $w$ is *information* = 11/19 = .579
  - $P(f)$, where $f$ is *data* = 7/19 = .368
  - $P(w,f)$, where $(w,f)$ is (*information*,*data*) = 6/19 = .316

$$PPMI_{assoc} = \log_2 \frac{P(w,f)}{P(w) \cdot P(f)}$$

$$= \log_2 \frac{0.316}{0.579 \cdot 0.368}$$

$$= 0.568$$

| | aardvark | computer | data | pinch | result | sugar |
|---|---|---|---|---|---|---|
| **apricot** | 0 | 0 | 0 | 1 | 0 | 1 |
| **pineapple** | 0 | 0 | 0 | 1 | 0 | 1 |
| **digital** | 0 | 2 | 1 | 0 | 1 | 0 |
| **information** | 0 | 1 | 6 | 0 | 4 | 0 |

# PPMI re-scaling

| | computer | data | result | pie | sugar | count(w) |
|---|---|---|---|---|---|---|
| **cherry** | 2 | 8 | 9 | 442 | 25 | 486 |
| **strawberry** | 0 | 0 | 1 | 60 | 19 | 80 |
| **digital** | 1670 | 1683 | 85 | 5 | 4 | 3447 |
| **information** | 3325 | 3982 | 378 | 5 | 13 | 7703 |
| **count(context)** | 4997 | 5673 | 473 | 512 | 61 | 11716 |

**Figure 6.9** Co-occurrence counts for four words in 5 contexts in the Wikipedia corpus, together with the marginals, pretending for the purpose of this calculation that no other words/contexts matter.

J&M 3rd ed. sec. 6.7

# PPMI re-scaling

| | computer | data | result | pie | sugar |
|---|---|---|---|---|---|
| **cherry** | 0 | 0 | 0 | 4.38 | 3.30 |
| **strawberry** | 0 | 0 | 0 | 4.10 | 5.51 |
| **digital** | 0.18 | 0.01 | 0 | 0 | 0 |
| **information** | 0.02 | 0.09 | 0.28 | 0 | 0 |

**Figure 6.11** The PPMI matrix showing the association between words and context words, computed from the counts in Fig. 6.10. Note that most of the 0 PPMI values are ones that had a negative PMI; for example PMI(*cherry,computer*) = -6.7, meaning that *cherry* and *computer* co-occur on Wikipedia less often than we would expect by chance, and with PPMI we replace negative values by zero.

J&M 3rd ed. sec. 6.7

# Weighting Features: Pointwise Mutual Information

- Downside:
  - PPMI favors rare events
- Solutions:
  - Change the $P(f)$ to be raised to the power of α
    - Increases the probability assigned to rare contexts
  - Laplace smoothing (add-$n$)

# Distributional Similarity Questions

What is the right neighborhood?

How should we weight the features?

**How can we compute the similarity between vectors?**

# Vector Distances: Manhattan & Euclidean

- **Manhattan Distance** $$dist_{manhattan} = (\vec{x}, \vec{y}) = \sum_{i=1}^{N} |x_i - y_i|$$
  - (Distance as cumulative horizontal + vertical moves)

# Vector Distances: Manhattan & Euclidean

- **Manhattan Distance**

$$dist_{manhattan} = (\vec{x}, \vec{y}) = \sum_{i=1}^{N} |x_i - y_i|$$

  - (Distance as cumulative horizontal + vertical moves)

- **Euclidean Distance**

$$dist_{euclidean} = \sqrt{\sum_{i=1}^{N} (x_i - y_i)^2}$$

# Vector Distances: Manhattan & Euclidean

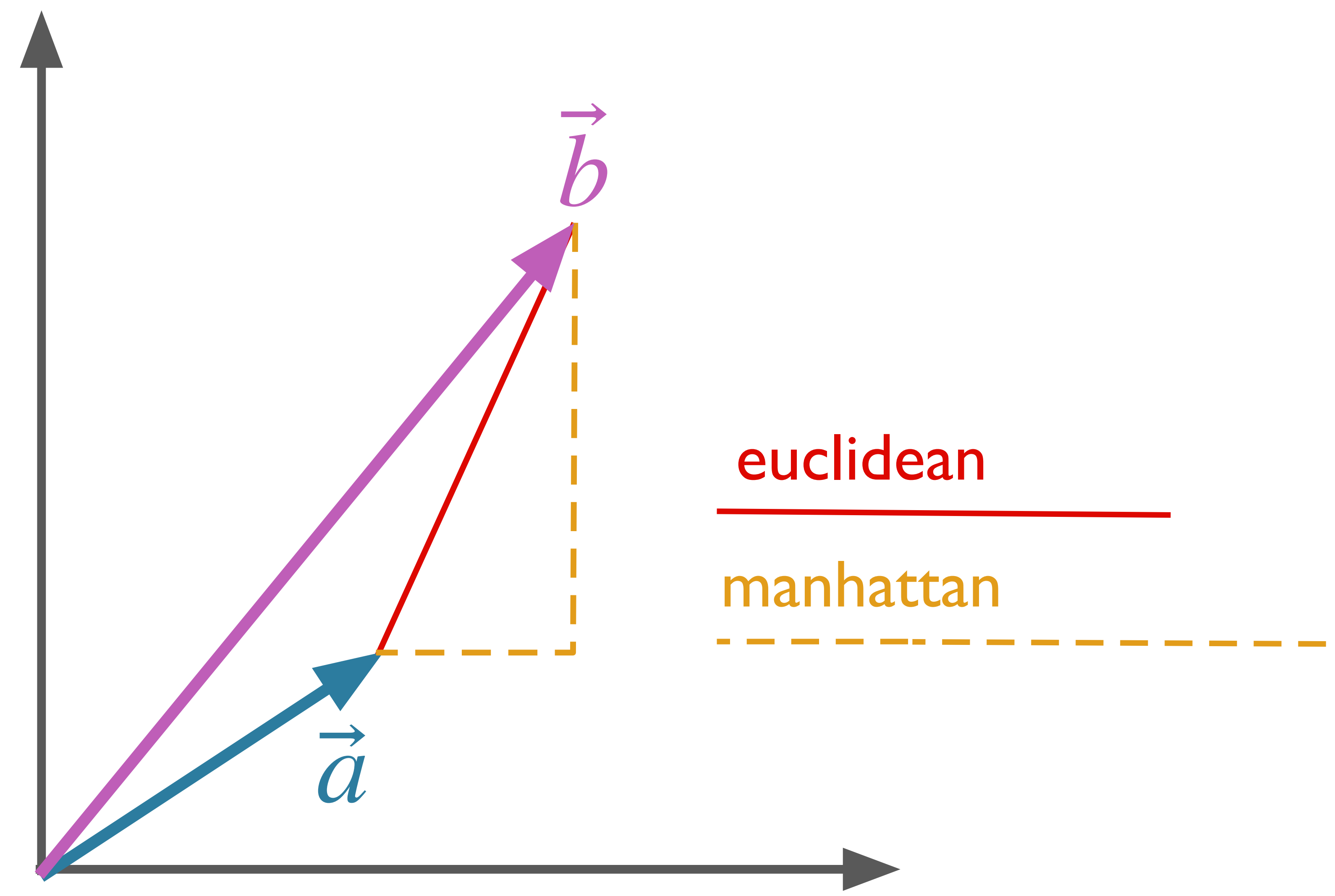
- **Manhattan Distance**

$$dist_{manhattan} = (\vec{x}, \vec{y}) = \sum_{i=1}^{N} |x_i - y_i|$$

  - (Distance as cumulative horizontal + vertical moves)

- **Euclidean Distance**

$$dist_{euclidean} = \sqrt{\sum_{i=1}^{N} (x_i - y_i)^2}$$



# Vector Distances: Manhattan & Euclidean

- **Manhattan Distance**

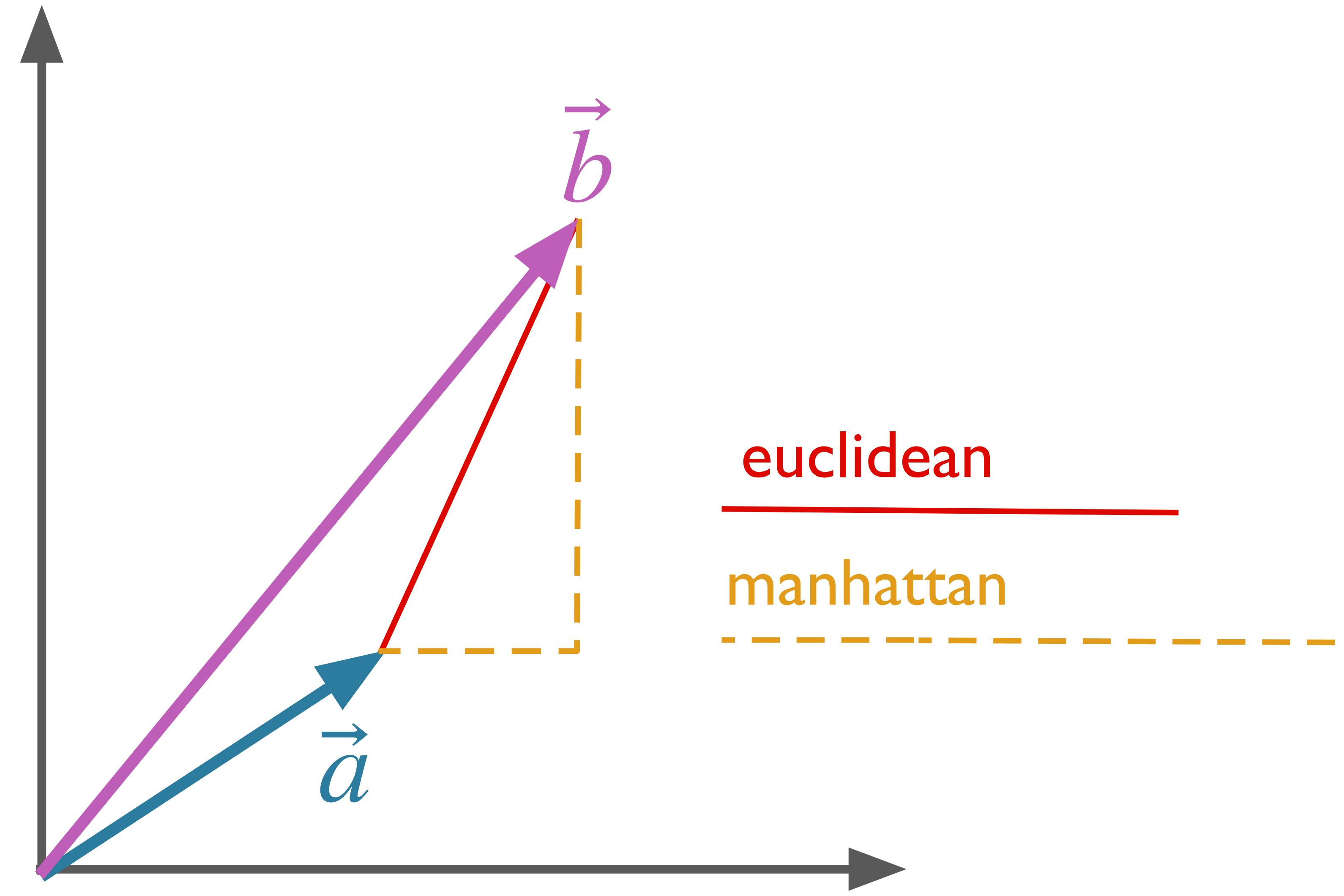
$$dist_{manhattan} = (\vec{x}, \vec{y}) = \sum_{i=1}^{N} |x_i - y_i|$$

  - (Distance as cumulative horizontal + vertical moves)

- **Euclidean Distance**

$$dist_{euclidean} = \sqrt{\sum_{i=1}^{N} (x_i - y_i)^2}$$

- Too sensitive to extreme values



# Vector Similarity: Dot Product

- Produces real number scalar from product of vectors' components

$$sim_{dot-product}(\vec{v}, \vec{w}) = \vec{v} \cdot \vec{w} = \sum_{i=1}^{N} v_i \times w_i$$

- Biased toward ***longer*** (larger magnitude) vectors
  - In our case, vectors with fewer zero counts

# Vector Similarity: Cosine

- If you normalize the dot product for vector magnitude…
- …result is same as cosine of angle between the vectors.

$$sim_{cosine}(\vec{v}, \vec{w}) = \frac{\vec{v} \cdot \vec{w}}{|\vec{v}||\vec{w}|} = \frac{\sum_{i=1}^{N} v_i \times w_i}{\sqrt{\sum_{i=1}^{N} v_i^2}\sqrt{\sum_{i=1}^{N} w_i^2}}$$

# Sample Results

- Based on Lin dependency model
  - **Hope (N)**: optimism, chance, expectation, prospect, dream, desire, fear
  - **Hope (V)**: would like, wish, plan, say, believe, think

  - **Brief (N)**: legal brief, affidavit, filing, petition, document, argument, letter
  - **Brief (A)**: lengthy, hour-long, short, extended, frequent, recent, short-lived, prolonged, week-long

# Recap

- We can build feature vectors to represent context of a word
- These features could be:

A. A bottle of ***tezgüino*** is on the table.
B. Everybody likes ***tezgüino***.
C. ***Tezgüino*** makes you drunk.
D. We make ***tezgüino*** from corn.

| |
|---|
| *tezgüino* |
| *tequila* |
| *apricots* |
| *pizza* |

# Recap

- We can build feature vectors to represent context of a word

- These features could be:
  1. Occurs before ***drunk***

A. A bottle of ***tezgüino*** is on the table.
B. Everybody likes ***tezgüino***.
*C.* ***Tezgüino*** makes you drunk.
*D.* We make ***tezgüino*** from corn.

| | 1 |
|---|---|
| *tezgüino* | 1 |
| *tequila* | 1 |
| *apricots* | 0 |
| *pizza* | 0 |

# Recap

- We can build feature vectors to represent context of a word

- These features could be:
    1. Occurs before ***drunk***
    2. Occurs after ***bottle***

A. A bottle of ***tezgüino*** is on the table.
B. Everybody likes ***tezgüino***.
*C.* ***Tezgüino*** makes you drunk.
*D.* We make ***tezgüino*** from corn.

| | 1 | 2 |
|---|---|---|
| *tezgüino* | 1 | 1 |
| *tequila* | 1 | 1 |
| *apricots* | 0 | 0 |
| *pizza* | 0 | 0 |

# Recap

- We can build feature vectors to represent context of a word

- These features could be:
  1. Occurs before *drunk*
  2. Occurs after *bottle*
  3. Is direct object of *likes*

A. A bottle of ***tezgüino*** is on the table.
B. Everybody likes ***tezgüino***.
*C.* ***Tezgüino*** makes you drunk.
*D.* We make ***tezgüino*** from corn.

| | 1 | 2 | 3 |
|---|---|---|---|
| *tezgüino* | 1 | 1 | 1 |
| *tequila* | 1 | 1 | 1 |
| *apricots* | 0 | 0 | 1 |
| *pizza* | 0 | 0 | 1 |

# Recap

- We can build feature vectors to represent context of a word

- These features could be:
  1. Occurs before *drunk*
  2. Occurs after *bottle*
  3. Is direct object of *likes*
  4. Is direct object of *make*

A. A bottle of ***tezgüino*** is on the table.
B. Everybody likes ***tezgüino***.
*C.* ***Tezgüino*** makes you drunk.
*D.* We make ***tezgüino*** from corn.

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| *tezgüino* | 1 | 1 | 1 | 1 |
| *tequila* | 1 | 1 | 1 | 1 |
| *apricots* | 0 | 0 | 1 | 0 |
| *pizza* | 0 | 0 | 1 | 1 |

# Recap

- These feature vectors can be as simple as co-occurrence
- …for vocabulary $V$
  - …for each element $i$
  - is word $v_i$ within window $w$ of target?

A. A bottle of ***tezgüino*** is on the table.
B. Everybody likes ***tezgüino***.
*C.* ***Tezgüino*** makes you drunk.
*D.* We make ***tezgüino*** from corn.

| bottle | drunk | matrix | table |
|---|---|---|---|
| 1 | 1 | 0 | 0 |

Context matrix for *tezgüino* with $w$=3

# Recap

- Intuition:
  - These co-occurrence vectors should be able to tell us something about words' similarities

| | arts | boil | data | function | large | sugar | summarized | water |
|---|---|---|---|---|---|---|---|---|
| **Apricot** | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| **Pineapple** | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| **Digital** | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 |
| **Information** | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 |

# Problem: Sparse Vectors!

- Big problem:
  - The vast majority of word pairs will be zero!
  - This leads to very sparse vectors.
- In the exercise:
  - (*election*, *primary*) is 2
  - (*election, midterm*) is 0
- …how can we generalize better?

# Problem: Sparse Vectors!

- Term x document:

| | c1 | c2 | c3 | c4 | c5 | m1 | m2 | m3 | m4 |
|---|---|---|---|---|---|---|---|---|---|
| human | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| interface | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| computer | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| user | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| system | 0 | 1 | 1 | 2 | 0 | 0 | 0 | 0 | 0 |
| response | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| time | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| EPS | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| survey | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| trees | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| graph | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| minors | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |