Pontificia Universidad JAVERIANA Colombia

# Procesamiento de Lenguaje Natural

## Tópicos Avanzados en Analítica

## Maestría en Analítica para la Inteligencia de Negocios

Sergio Alberto Mora Pardo - H2 2023

# Text Representation

## Feature Engineering

1. Space Vector Models
2. Basic Vectorization Approaches
3. Distributed Representations
4. Visualizing Embeddings***

***opcional

“Conocerás una palabra por la compañía que tiene”

JR Fiordo - 1957

# Text Representation

## Feature Engineering

**Similitud Distributiva**

*El significado de una palabra puede entenderse a partir del contexto en el que aparece.*

*Connotación*

*Significado definido por contexto*

*Denotación*

*Significado literal de la palabra*

Ej.: “NLP rocks”

# Text Representation

## Feature Engineering

**Similitud Distributiva**

*El significado de una palabra puede entenderse a partir del contexto en el que aparece.*

*Connotación*

*Significado definido por contexto*

*Denotación*

*Significado literal de la palabra*

Ej.: “NLP rocks”

*Bueno y de moda.*

*Roca, piedra*

# Text Representation

## Feature Engineering

**Hipótesis Distributiva**

*En lingüística, plantea la hipótesis: “Las palabras que ocurren en contextos similares, tienen significados similares”.*



*VSM: vector space model*

*Ejemplo:*

*Palabra 1: “Perro”*

*Palabra 2: “Gato”*

*1) Aparecen en contextos similares.*

*2) Deben tener significado similar*

*3) Vectores de representación deben estar uno cerca al otro en VSM.*

Firth, John R. "Una sinopsis de la teoría lingüística 1930-1955". *Estudios de Análisis Lingüístico* (1968).

# Text Representation

## Feature Engineering

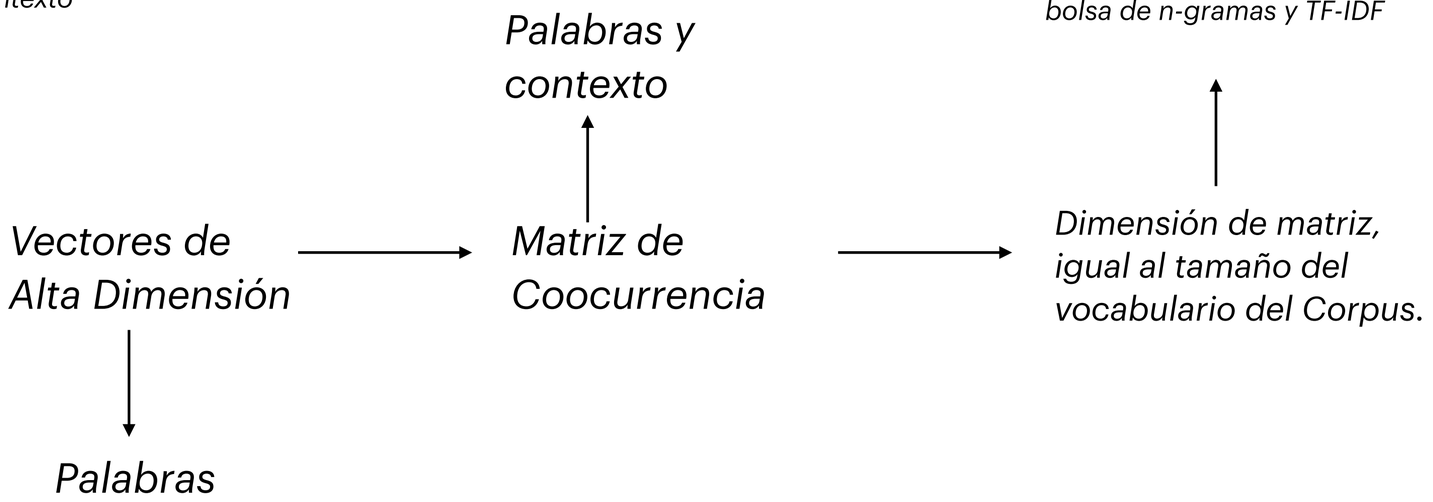
**Representación Distributiva**

*Esquemas de representación que se obtienen en base a la distribución de palabras del contexto en el que aparecen.*

*Estos esquemas se basan en hipótesis distributivas.*

*Vecindad textual:*

*La propiedad distributiva se induce a partir del contexto*




Ferrone, Lorenzo y Fabio Massimo Zanzotto. “Representaciones simbólicas, distribuidas y distributivas para el procesamiento del lenguaje natural en la era del aprendizaje profundo: una encuesta” , (2017).

# Text Representation

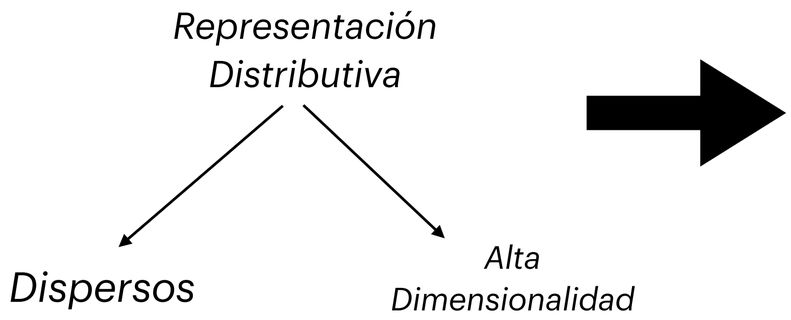
## Feature Engineering

**Representación**
**Distribuida**

*Estos esquemas también se basan en hipótesis distributivas pero comprimen dimensionalidad.*

*Vecindad textual:*

*La propiedad distributiva se induce a partir del contexto*




*1. Computacionalmente ineficientes*

*2. Dificulta el entrenamiento*

Ferrone, Lorenzo y Fabio Massimo Zanzotto. “Representaciones simbólicas, distribuidas y distributivas para el procesamiento del lenguaje natural en la era del aprendizaje profundo: una encuesta” , (2017).

# Text Representation

## Feature Engineering

**Representación**
**Distribuida**

*Estos esquemas también se basan en hipótesis distributivas pero comprimen dimensionalidad.*

**Representación**
**Distribuida** → *Espacio vectorial resultante de comprimir significativamente la dimensionalidad*

*1. Computacionalmente ineficientes*

*2. Dificulta el entrenamiento*

**Comprimen la dimensionalidad**

*1. Vectores compactos*
*De baja dimensión*

*2. Vectores densos.*
*Casi sin ceros.*

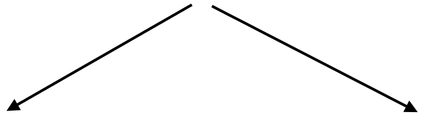
Ferrone, Lorenzo y Fabio Massimo Zanzotto. “Representaciones simbólicas, distribuidas y distributivas para el procesamiento del lenguaje natural en la era del aprendizaje profundo: una encuesta” , (2017).

# Text Representation

## Feature Engineering

### Embeddings

*Mapeo entre:*



*Espacio Vectorial de la representación distributiva*

*Espacio Vectorial de la representación distribuida*

### Semantica vectorial

Esto se refiere al conjunto de métodos de PNL que tienen como objetivo aprender las representaciones de palabras basadas en las propiedades distributivas de las palabras en un corpus grande.

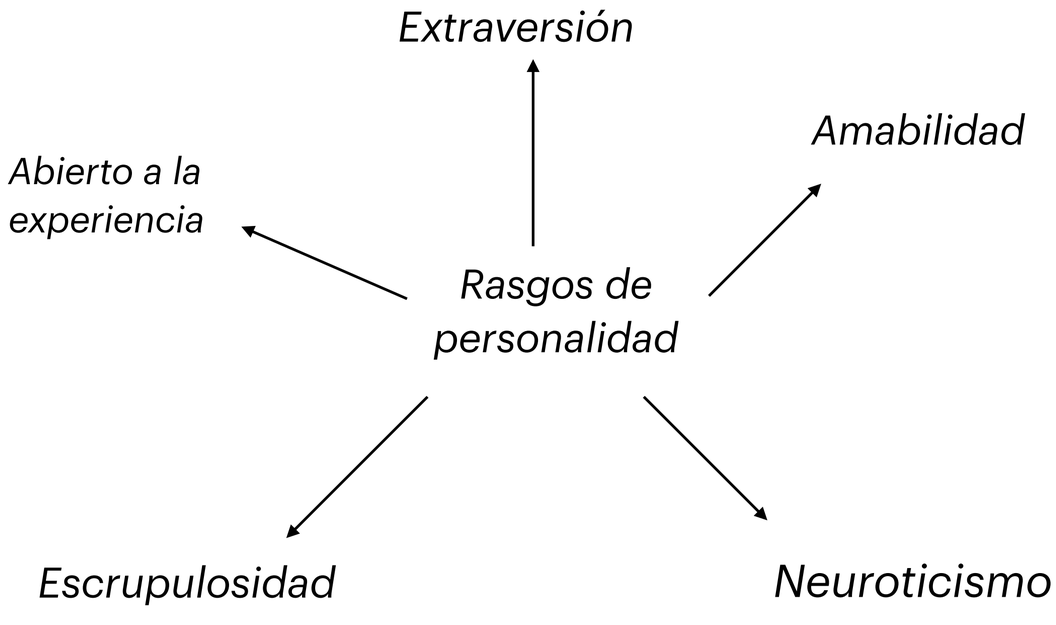
# Text Representation

## Embeddings

### Personality Embeddings: What are you like?

| | |
|---|---|
| Openness to experience | 79 out of 100 |
| Agreeableness | 75 out of 100 |
| Conscientiousness | 42 out of 100 |
| Negative emotionality | 50 out of 100 |
| Extraversion | 58 out of 100 |

*Un vector de 5 dimensiones puede representar mucho sobre tu personalidad.*

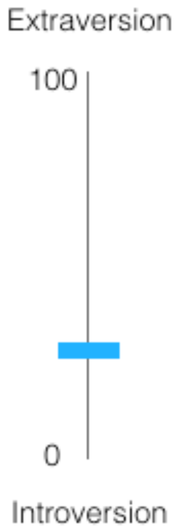


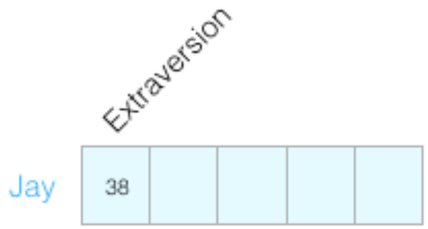

Rothmann S, Coetzer EP (24 October 2003). "The big five personality dimensions and job performance". *SA Journal of Industrial Psychology*. **29**. doi:10.4102/sajip.v29i1.88.

# Text Representation

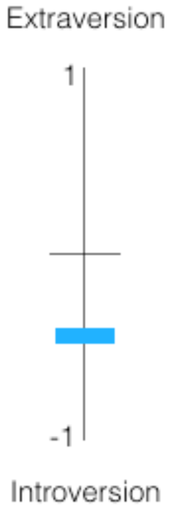
## Embeddings

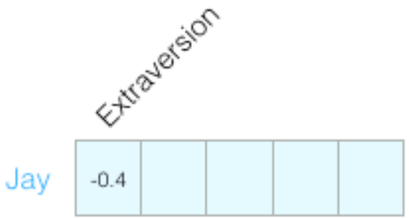
*Jay tuvo 38/100 en Extraversión* → *Se usa una escala de 0 a 100*




*Jay tuvo -0.4 en Extraversión* → *Rescatamos de -1 a 1.*

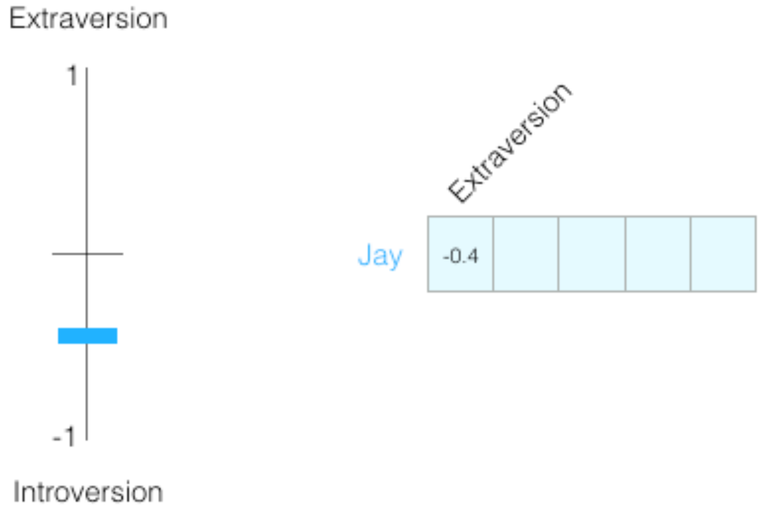


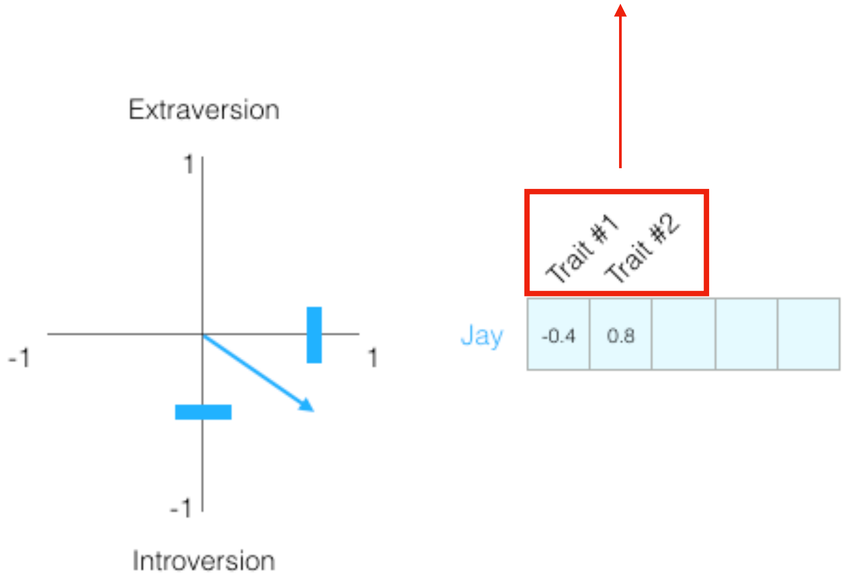
# Text Representation

## Embeddings

*Jay tuvo -0.4 en Extraversión* ⟶ *Rescatamos de -1 a 1.*



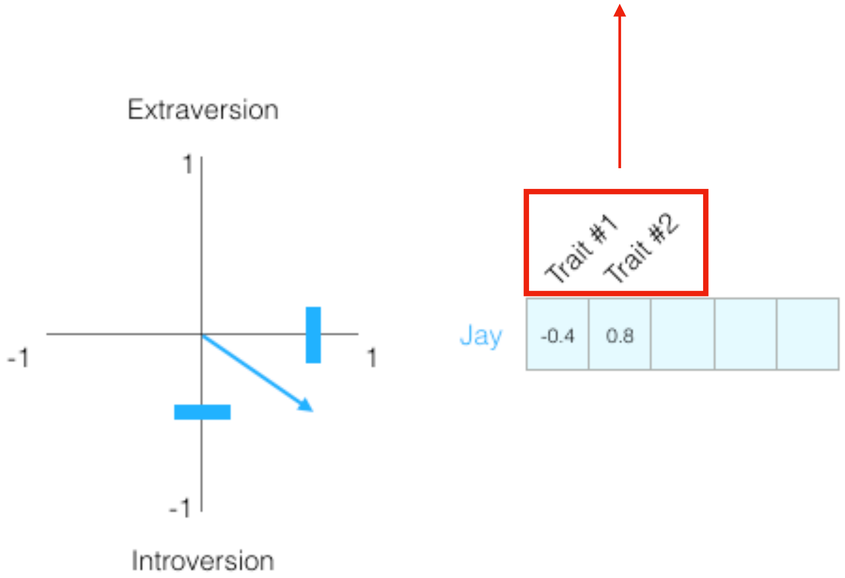

*Ocultemos los rasgos para acostumbrarnos a no saber que representa cada dimensión*



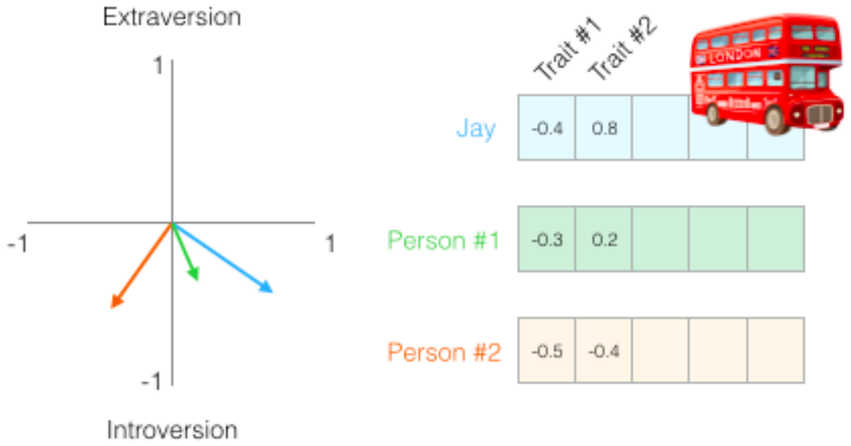
# Text Representation

## Embeddings

*Ocultemos los rasgos para acostumbrarnos a no saber que representa cada dimensión*




*Supongamos que un bus golpea a Jay. Tendríamos que reemplazarlo con una personalidad similar.*

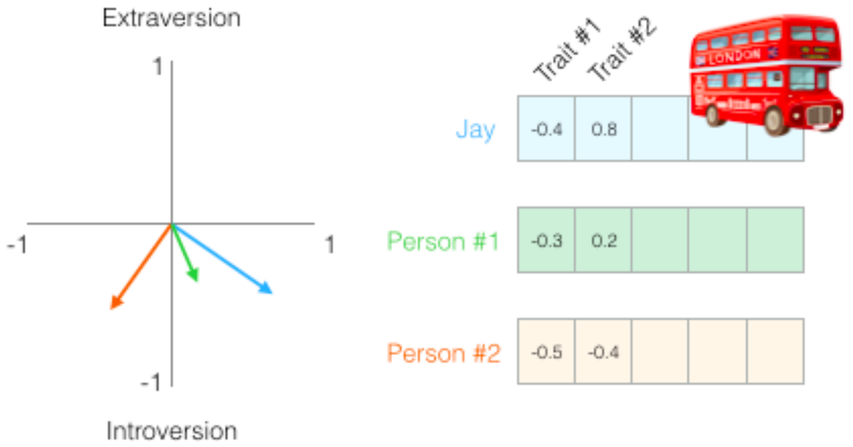



*¿Cuál de las personas se parecen a Jay?*

# Text Representation

## Embeddings

*Supongamos que un bus golpea a Jay. Tendríamos que reemplazarlo con una personalidad similar.*




*¿Cuál de las personas se parecen a Jay?*

```
                    Jay                 Person #1
cosine_similarity( -0.4  0.8 ,  -0.3  0.2 ) = 0.87
```

✓

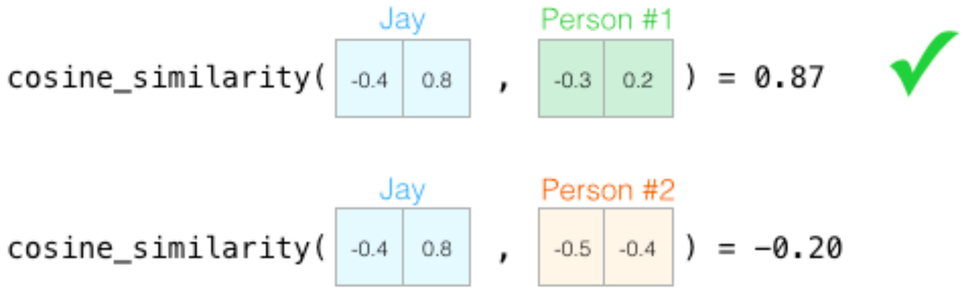
```
                    Jay                 Person #2
cosine_similarity( -0.4  0.8 ,  -0.5  -0.4 ) = -0.20
```

La persona número 1 se parece más a Jay en personalidad.

*Los vectores que apuntan en la misma dirección (la longitud también influye) tienen una puntuación de similitud de coseno más alta.*

# Text Representation

## Embeddings




La persona número 1 se parece más a Jay en personalidad.

*Los vectores que apuntan en la misma dirección (la longitud también influye) tienen una puntuación de similitud de coseno más alta.*

*Usemos ahora todos las dimensiones para capturar suficiente información, sobre cuán diferentes son dos personas*

*Cinco rasgos principales y muchos subrasgos.*

| | Trait #1 | Trait #2 | Trait #3 | Trait #4 | Trait #5 |
|---|---|---|---|---|---|
| Jay | -0.4 | 0.8 | 0.5 | -0.2 | 0.3 |
| Person #1 | -0.3 | 0.2 | 0.3 | -0.4 | 0.9 |
| Person #2 | -0.5 | -0.4 | -0.2 | 0.7 | -0.1 |

# Text Representation

## Embeddings

Similitud de coseno funciona para cualquier número de dimensiones.

Jay Person #1

cosine_similarity( -0.4 0.8 0.5 -0.2 0.3 , -0.3 0.2 0.3 -0.4 0.9 ) = 0.66 ✓

Jay Person #2

cosine_similarity( -0.4 0.8 0.5 -0.2 0.3 , -0.5 -0.4 -0.2 0.7 -0.1 ) = -0.37

1- We can represent things (and people) as vectors of numbers
(Which is great for machines!)

Jay -0.4 0.8 0.5 -0.2 0.3

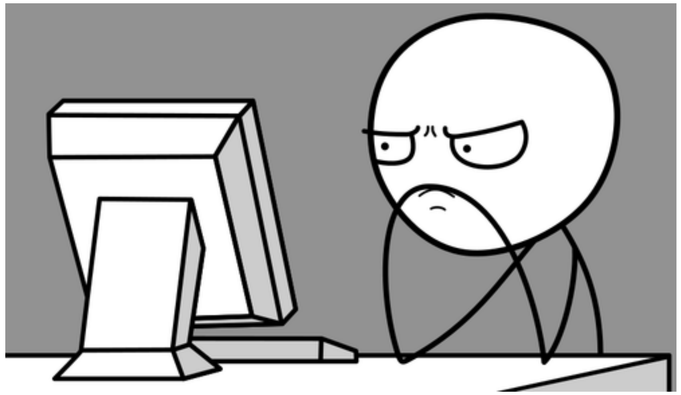
2- We can easily calculate how similar vectors are to each other

The people most similar to Jay are:

| | cosine_similarity ▼ |
|---|---|
| Person #1 | 0.86 |
| Person #2 | 0.5 |
| Person #3 | -0.20 |

# Text Representation

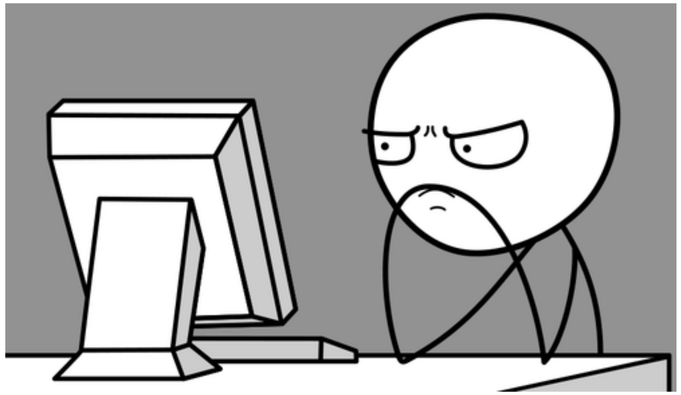
## Embeddings



queen = king - he + she?

# Text Representation

## Embeddings

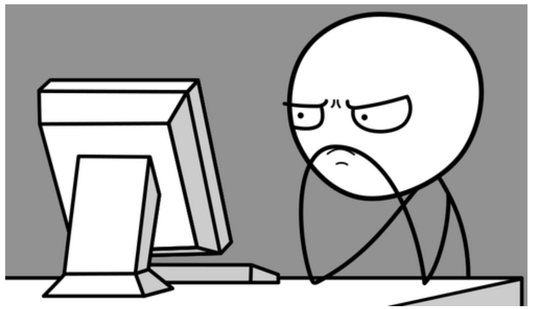


queen = king - he + she?

$$P(w_{i-m}, \dots, w_{i-1}, w_{i+1}, \dots, w_{i+m} | w_i) = \prod_{j \neq i \wedge j = i-m}^{i+m} P(w_j | w_i)$$

# Text Representation

## Embeddings



queen = king - he + she

**There was a very rich king. He had a beautiful queen. She was very kind.**



*was rich king he had beautiful queen she was kind*

# Text Representation

## Embeddings

(*target word --> context word 1, context word 2*)



**queen = king - he + she**

*was rich king he had beautiful queen she was kind*

*was --> rich*

*rich --> was, king*

*king --> rich, he*

*he --> king, had*

*had --> he, beautiful*

*beautiful --> had, queen*

*queen --> beautiful, she*

*she --> queen, was*

*was --> she, kind*

*kind --> was*

# Text Representation

## Embeddings

(*target word --> context word 1, context word 2*)



**queen = king - he + she**

*was rich king he had beautiful queen she was kind*

*Iniciamos asumiendo un vector para la palabra ‘rich’:*

*rich --> [0,0]*

# Text Representation

## Embeddings

(*target word --> context word 1, context word 2*)



**queen = king - he + she**

*was rich king he had beautiful queen she was kind*

*Para pronósticar bien “was” y “king”. Estás dos palabras deberían tener una alta similitud con la palabra “rich”*

*rich --> [0,0]*

*king --> [0,1]*

*was --> [-1,0]*

# Text Representation

## Embeddings

(*target word --> context word 1, context word 2*)



**queen = king - he + she**

*was rich king he had beautiful queen she was kind*

*Para pronósticar bien “was” y “king”. Estás dos palabras deberían tener una alta similitud con la palabra “rich”*

*Dist(rich,king) = 1.0*

*Dist(rich,was) = 1.0*

# Text Representation

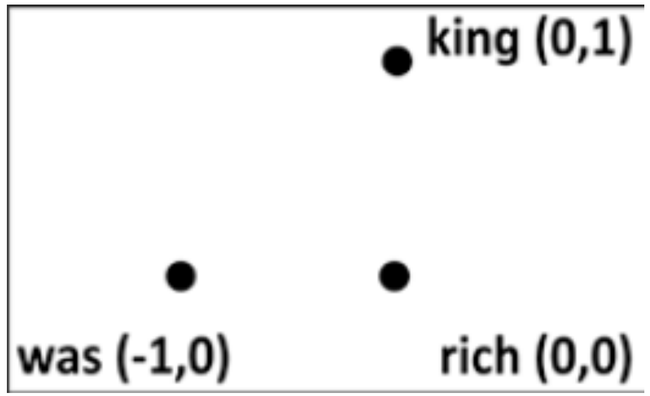
## Embeddings

(*target word --> context word 1, context word 2*)

**queen = king - he + she**

*was rich king he had beautiful queen she was kind*

*Para pronósticar bien “was” y “king”. Estás dos palabras deberían tener una alta similitud con la palabra “rich”*



# Text Representation

## Embeddings

(*target word --> context word 1, context word 2*)



**queen = king - he + she**

*was rich king he had beautiful queen she was kind*

*Ahora consideremos la siguiente tupla.*

*king --> rich, he*

*Entre más veamos una relación, más cercanas deben ser esas dos palabras.*

*king --> [0,0.8]*

# Text Representation

## Embeddings

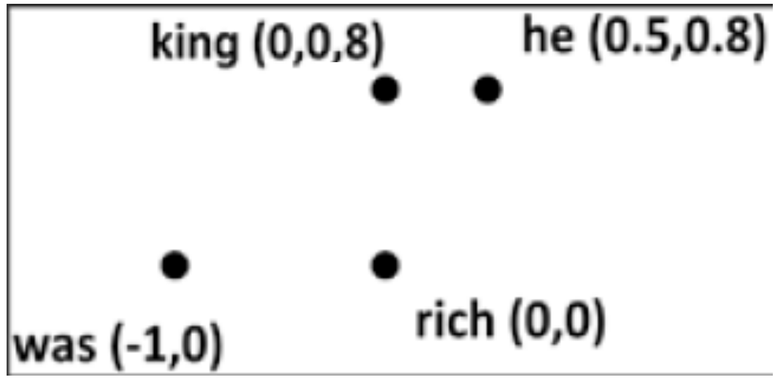
(*target word --> context word 1, context word 2*)

**queen = king - he + she**

*was rich king he had beautiful queen she was kind*

*Ahora agreguemos la palabra “he” que debería estar cerca a la palabra “king”*

*he --> [0.5,0.8].*



# Text Representation

## Embeddings

(*target word --> context word 1, context word 2*)



**queen = king - he + she**

*was rich king he had beautiful queen she was kind*

*Continuando con la siguientes*** dos tuplas, tenemos:*

*queen --> beautiful, she*

*she --> queen, was*

****Se hace un intercambio en el orden de las tuplas para mantener sencillo el ejercicio.*

# Text Representation

## Embeddings

(*target word --> context word 1, context word 2*)



**queen = king - he + she**

*was rich king he had beautiful queen she was kind*

*Deberíamos igual mantener la distancia desde “was” con “she”. Así como es con “was” y con “he”.*

*she --> [0.5,0.6]*

*Ahora, dejemos la palabra “queen” cerca de la palabra “she”.*

*queen --> [0.0,0.6]*

# Text Representation

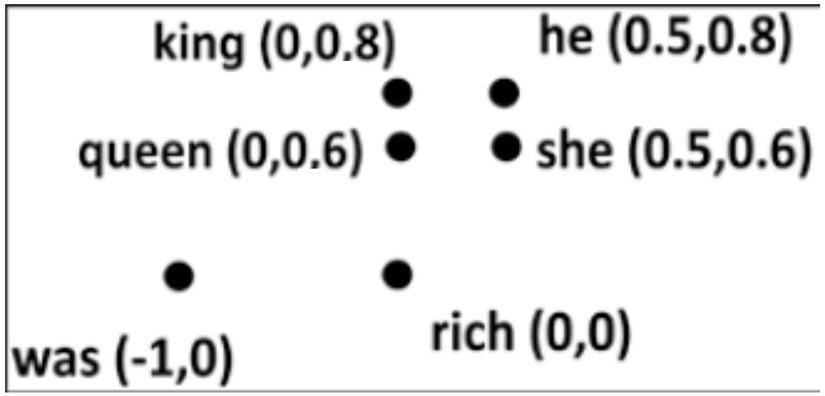
## Embeddings

(*target word --> context word 1, context word 2*)

**queen = king - he + she**

*was rich king he had beautiful queen she was kind*

*Tendríamos así el siguiente espacio euclidiano:*



# Text Representation

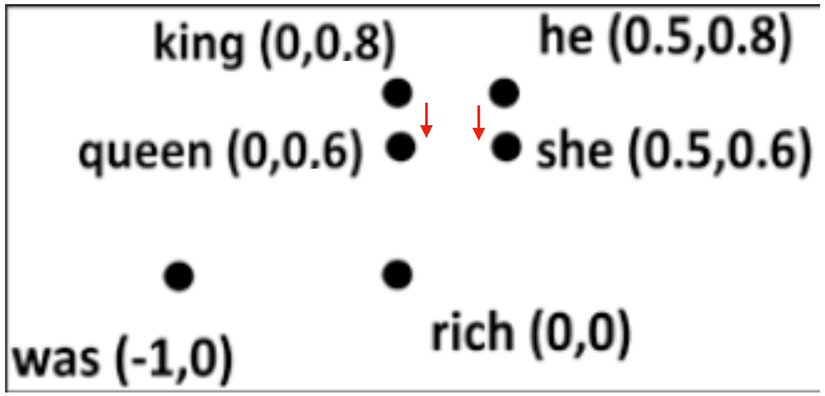
## Embeddings

(*target word --> context word 1, context word 2*)

**queen = king - he + she**

*was rich king he had beautiful queen she was kind*

*Tendríamos así el siguiente espacio euclidiano:*



# Text Representation

## Embeddings

(*target word --> context word 1, context word 2*)



**queen = king - he + she**

*was rich king he had beautiful queen she was kind*

*Ahora agreguemos solo una tupla más:*

*queen --> beautiful, she*

*Esta debería estar entre las palabras “queen” y la palabra “she”.*

*beautiful --> [0.25,0]*

# Text Representation

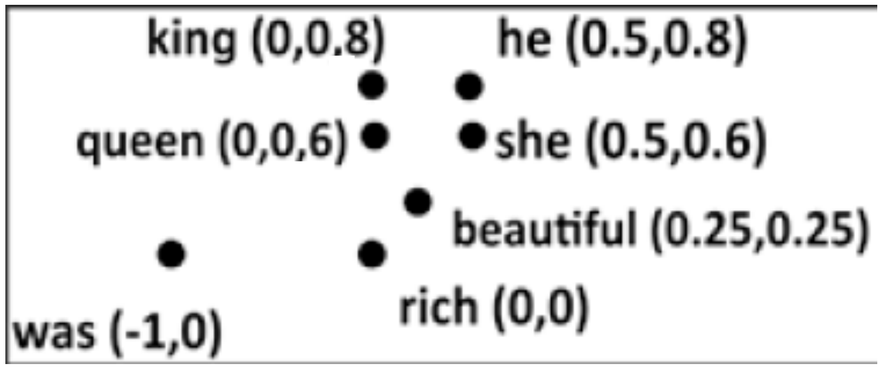
## Embeddings

(*target word --> context word 1, context word 2*)

**queen = king - he + she**

*was rich king he had beautiful queen she was kind*

*Tendríamos la siguiente representación:*



# Text Representation

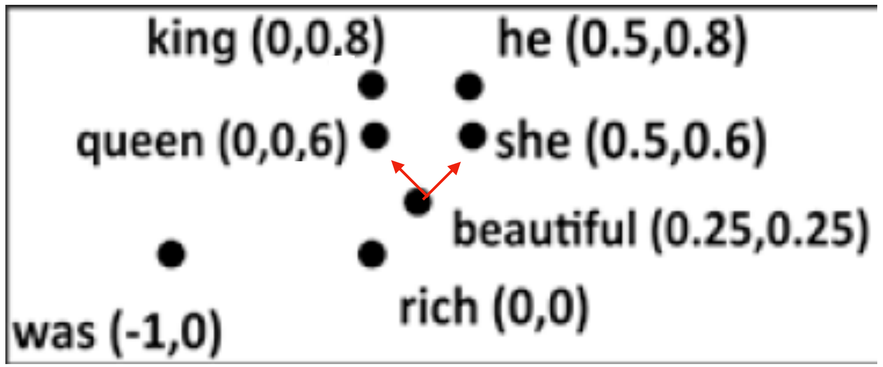
## Embeddings

(*target word --> context word 1, context word 2*)

**queen = king - he + she**

*was rich king he had beautiful queen she was kind*

*Tendríamos la siguiente representación:*



# Text Representation

## Embeddings

(*target word --> context word 1, context word 2*)

**queen = king - he + she**

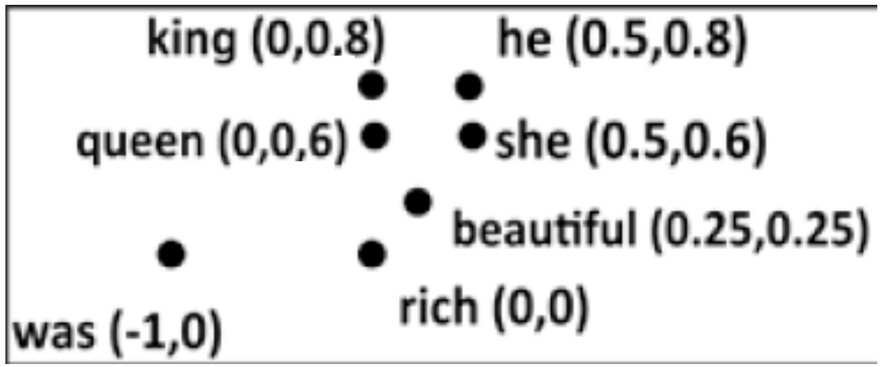
*= king – he + she*

*= [0,0.8] – [0.5,0.8] + [0.5,0.6]*

*= [0,0.6]*

*was rich king he had beautiful queen she was kind*

*Tendríamos la siguiente representación:*




*¡Al final funcionó!*

# Text Representation

## Embeddings

GloVe: Global Vectors for Word Representation

**GloVe: Global Vectors for Word Representation**

**Jeffrey Pennington, Richard Socher, Christopher D. Manning**
Computer Science Department, Stanford University, Stanford, CA 94305
jpennin@stanford.edu, richard@socher.org, manning@stanford.edu

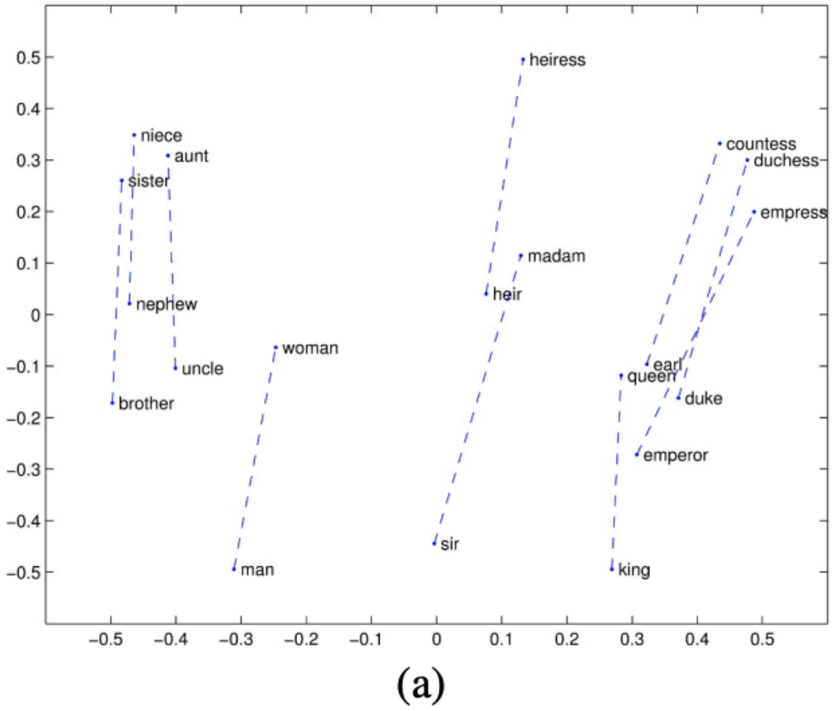
**Abstract**

Recent methods for learning vector space representations of words have succeeded in capturing fine-grained semantic and syntactic regularities using vector arithmetic, but the origin of these regularities has remained opaque. We analyze and make explicit the model properties needed for such regularities to emerge in word vectors. The result is a new global log-bilinear regression model that combines the advantages of the two major model families in the literature: global matrix

the finer structure of the word vector space by examining not the scalar distance between word vectors, but rather their various dimensions of difference. For example, the analogy "king is to queen as man is to woman" should be encoded in the vector space by the vector equation *king − queen = man − woman*. This evaluation scheme favors models that produce dimensions of meaning, thereby capturing the multi-clustering idea of distributed representations (Bengio, 2009).

The two main model families for learning word vectors are: 1) global matrix factorization methods, such as latent semantic analysis (LSA) (Deer-

Propiedad semántica de embeddings en Glove.

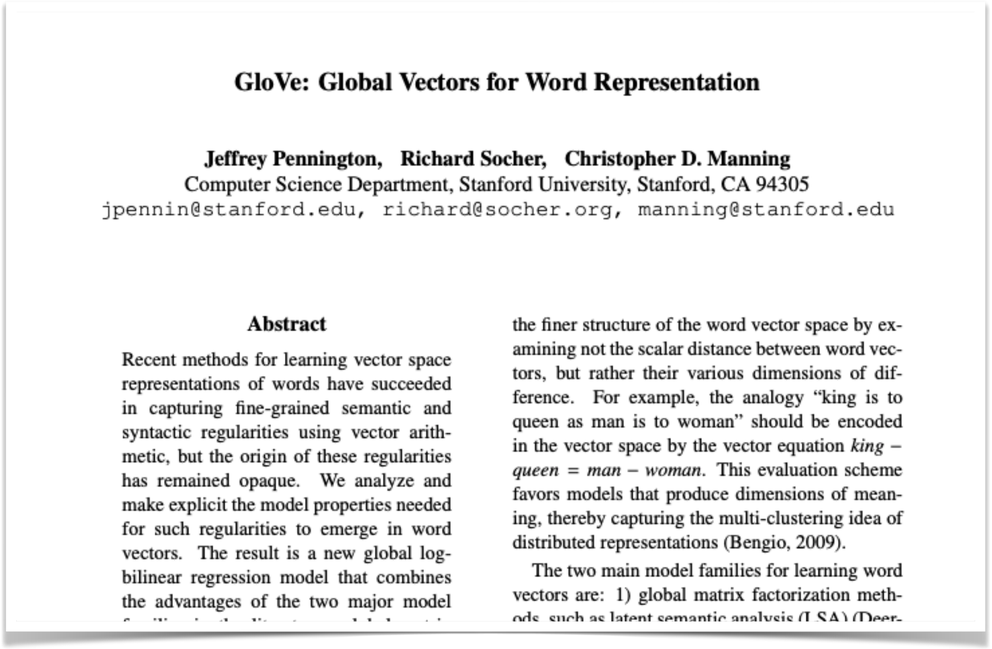



king − man + woman = queen (semantic feature)

# Text Representation

## Embeddings

GloVe: Global Vectors for Word Representation



**GloVe: Global Vectors for Word Representation**

**Jeffrey Pennington, Richard Socher, Christopher D. Manning**
Computer Science Department, Stanford University, Stanford, CA 94305
jpennin@stanford.edu, richard@socher.org, manning@stanford.edu

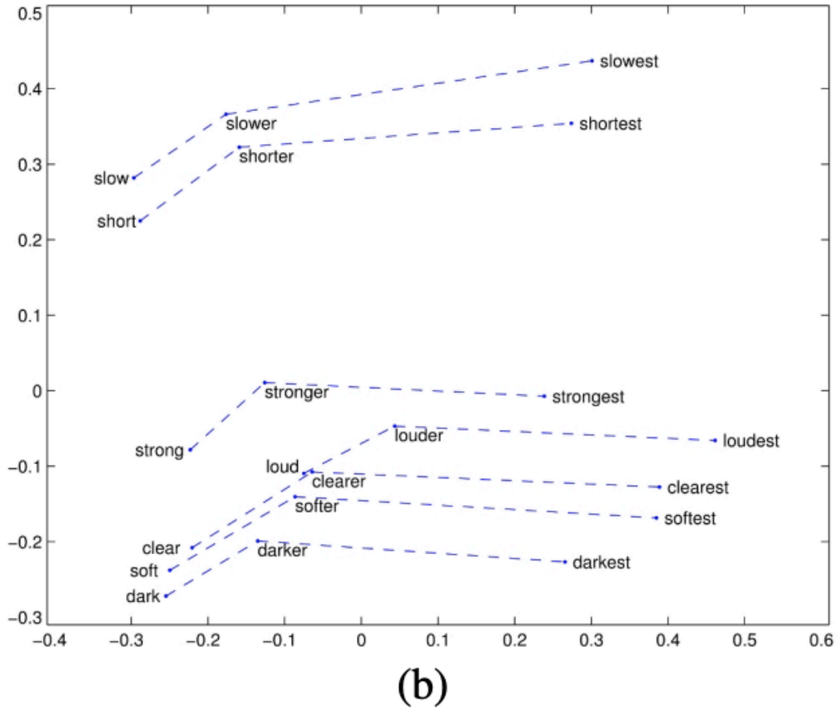
**Abstract**

Recent methods for learning vector space representations of words have succeeded in capturing fine-grained semantic and syntactic regularities using vector arithmetic, but the origin of these regularities has remained opaque. We analyze and make explicit the model properties needed for such regularities to emerge in word vectors. The result is a new global log-bilinear regression model that combines the advantages of the two major model

the finer structure of the word vector space by examining not the scalar distance between word vectors, but rather their various dimensions of difference. For example, the analogy "king is to queen as man is to woman" should be encoded in the vector space by the vector equation *king − queen = man − woman*. This evaluation scheme favors models that produce dimensions of meaning, thereby capturing the multi-clustering idea of distributed representations (Bengio, 2009).

The two main model families for learning word vectors are: 1) global matrix factorization meth-

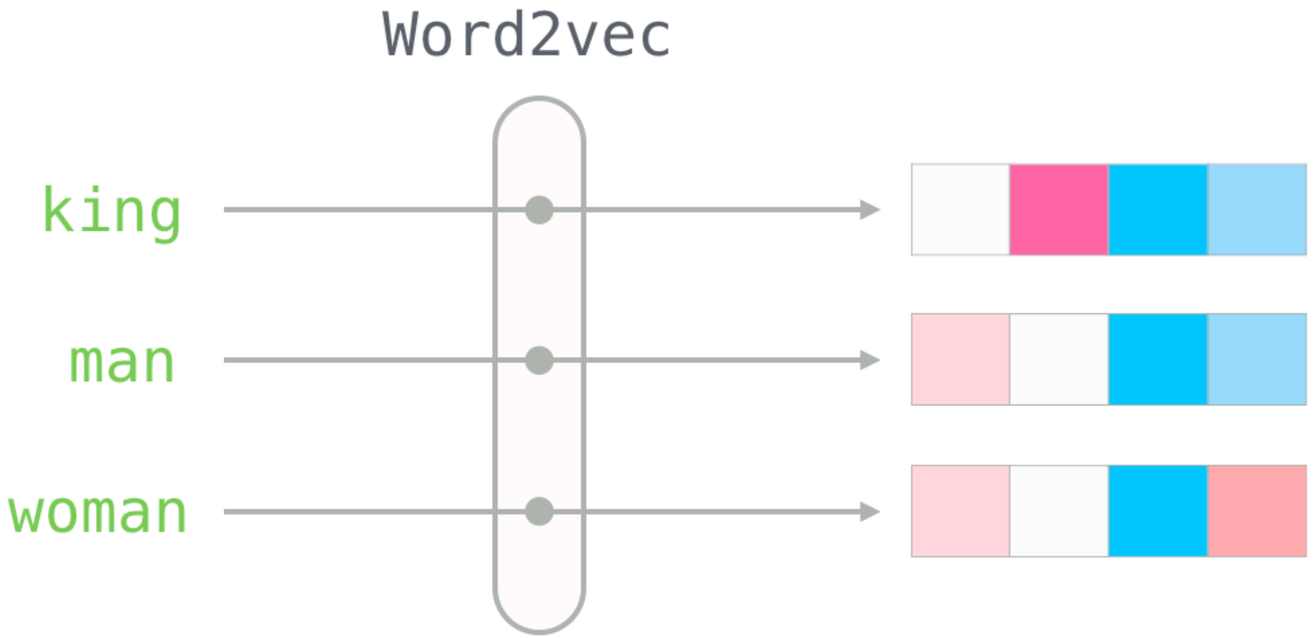
Propiedad semántica de embeddings en Glove.




Capturan una morfología comparativa y superlativa

# Text Representation

## Word Embeddings



# Text Representation

## Word Embeddings

Applied Data Science Track Paper KDD 2018, August 19-23, 2018, London, United Kingdom

### Real-time Personalization using Embeddings for Search Ranking at Airbnb

Mihajlo Grbovic
Airbnb, Inc.
San Francisco, California, USA
mihajlo.grbovic@airbnb.com

Haibin Cheng
Airbnb, Inc.
San Francisco, California, USA
haibin.cheng@airbnb.com

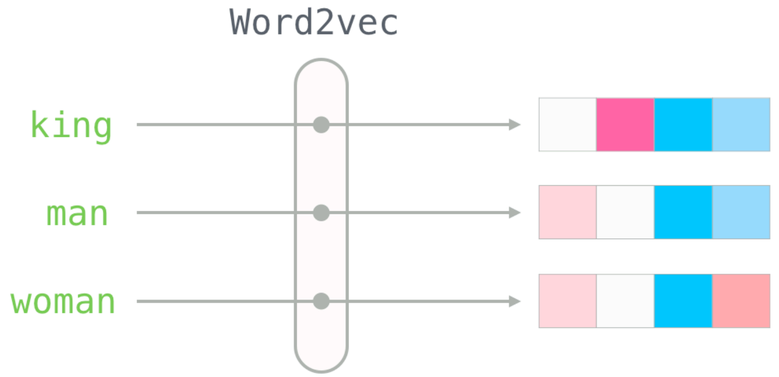
**ABSTRACT**

Search Ranking and Recommendations are fundamental problems of crucial interest to major Internet companies, including web search engines, content publishing websites and marketplaces. However, despite sharing some common characteristics a one-size-fits-all solution does not exist in this space. Given a large difference in content that needs to be ranked, personalized and recommended, each marketplace has a somewhat unique challenge. Correspond-

**1 INTRODUCTION**

During last decade Search architectures, which were typically based on classic Information Retrieval, have seen an increased presence of Machine Learning in its various components [2], especially in Search Ranking which often has challenging objectives depending on the type of content that is being searched over. The main reason behind this trend is the rise in the amount of search data that can be collected and analyzed. The large amounts of collected data

https://dl.acm.org/doi/pdf/10.1145/3219819.3219885



# Text Representation

## Word Embeddings

Applied Data Science Track Paper KDD 2018, August 19-23, 2018, London, United Kingdom

### Billion-scale Commodity Embedding for E-commerce Recommendation in Alibaba

Jizhe Wang, Pipei Huang*
Alibaba Group
Hangzhou and Beijing, China
{jizhe.wjz,pipei.hpp}@alibaba-inc.com

Huan Zhao
Department of Computer Science and Engineering
Hong Kong University of Science and Technology
Kowloon, Hong Kong
hzhaoaf@cse.ust.hk

Zhibo Zhang, Binqiang Zhao
Alibaba Group
Beijing, China
{shaobo.zzb,binqiang.zhao}@alibaba-inc.com

Dik Lun Lee
Department of Computer Science and Engineering
Hong Kong University of Science and Technology
Kowloon, Hong Kong
dlee@cse.ust.hk

**ABSTRACT**

Recommender systems (RSs) have been the most important technology for increasing the business in Taobao, the largest online consumer-to-consumer (C2C) platform in China. There are three major challenges facing RS in Taobao: scalability, sparsity and cold start. In this paper, we present our technical solutions to address these three challenges. The methods are based on a well-known graph embedding framework. We first construct an item graph from users' behavior history, and learn the embeddings of all items in the graph. The item embeddings are employed to compute pairwise similarities between all items, which are then used in the recommendation process. To alleviate the sparsity and cold start problems, side information is incorporated into the graph

*algorithms*; • **Computing methodologies** → *Learning latent representations*;

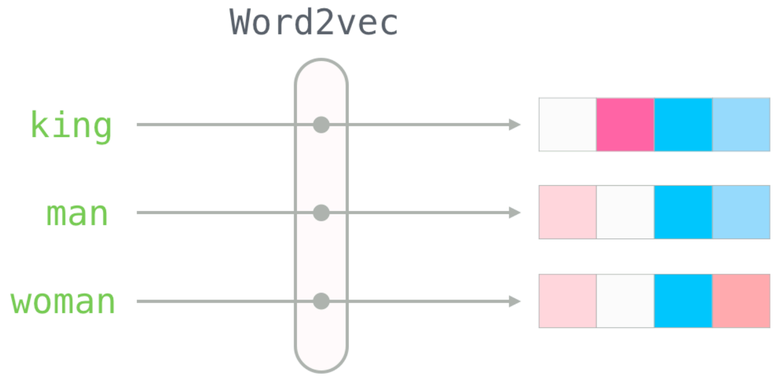
**KEYWORDS**

Recommendation system; Collaborative filtering;
Graph Embedding; E-commerce Recommendation.

**1 INTRODUCTION**

Internet technology has been continuously reshaping the business landscape, and online businesses are everywhere nowadays. Alibaba, the largest provider of online business in China, makes it possible for people or companies all over the world to do business

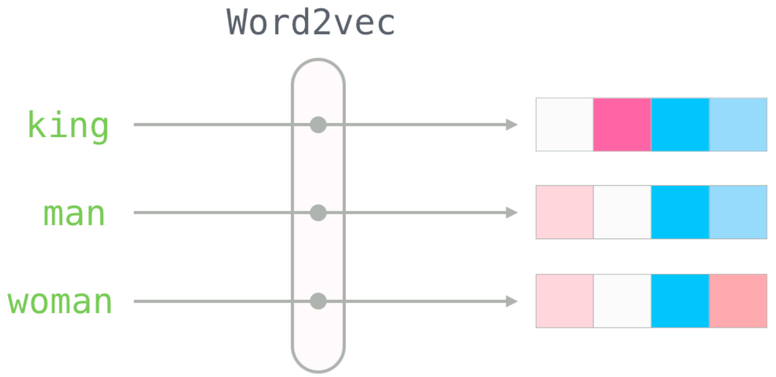
https://dl.acm.org/doi/pdf/10.1145/3219819.3219869



# Text Representation

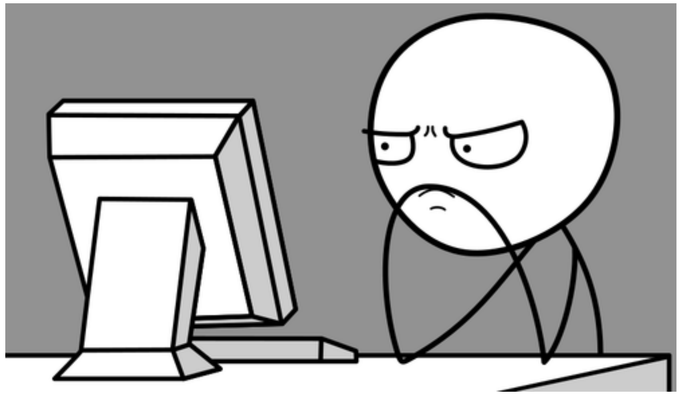
## Word Embeddings

https://towardsdatascience.com/using-word2vec-for-music-recommendations-bb9649ac2484



# Text Representation

## Word Embeddings

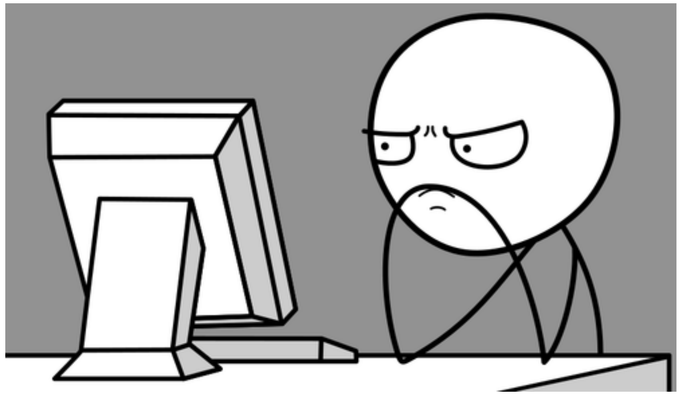


## ¿Cómo entrenar un modelo así?

$$P(w_{i-m}, \dots, w_{i-1}, w_{i+1}, \dots, w_{i+m} | w_i) = \prod_{j \neq i \wedge j = i-m}^{i+m} P(w_j | w_i)$$

# Text Representation

## Word2Vec



## ¿Cómo entrenar un modelo así? -> Language Modeling

$$P(w_{i-m}, \dots, w_{i-1}, w_{i+1}, \dots, w_{i+m} | w_i) = \prod_{j \neq i \wedge j = i-m}^{i+m} P(w_j | w_i)$$

# Text Representation

## Word2Vec

## Language Modeling

*La predicción de la siguiente palabra.*

# Text Representation

## Word2Vec

### Language Modeling

*La predicción de la siguiente palabra.*



*Un modelo de lenguaje puede tomar una lista de palabras (dos por ejemplo) e intentar predecir la siguiente.*

| input/feature #1 | input/feature #2 | output/label |
| --- | --- | --- |
| Thou | shalt | ____ |

# Text Representation

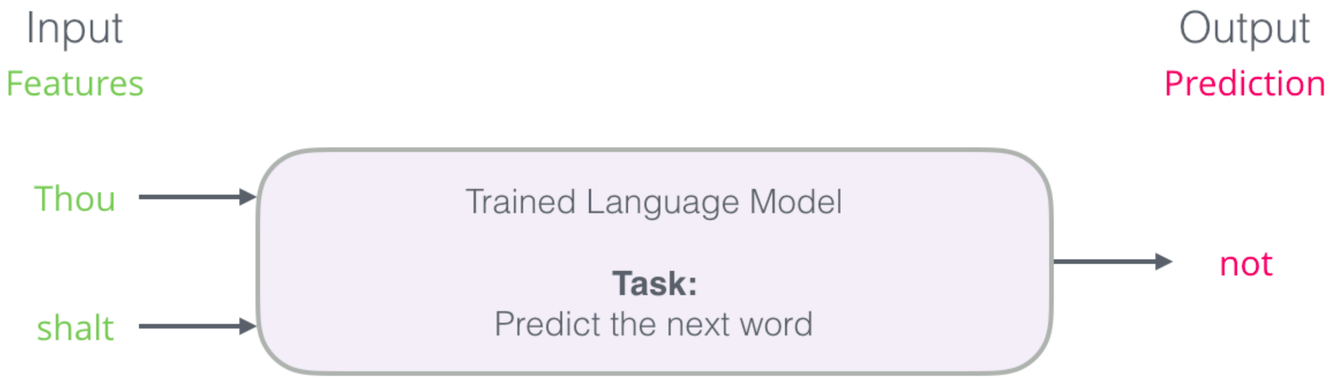
## Word2Vec

### Language Modeling

*La predicción de la siguiente palabra.*



*¿Cuál sería el output del modelo?*



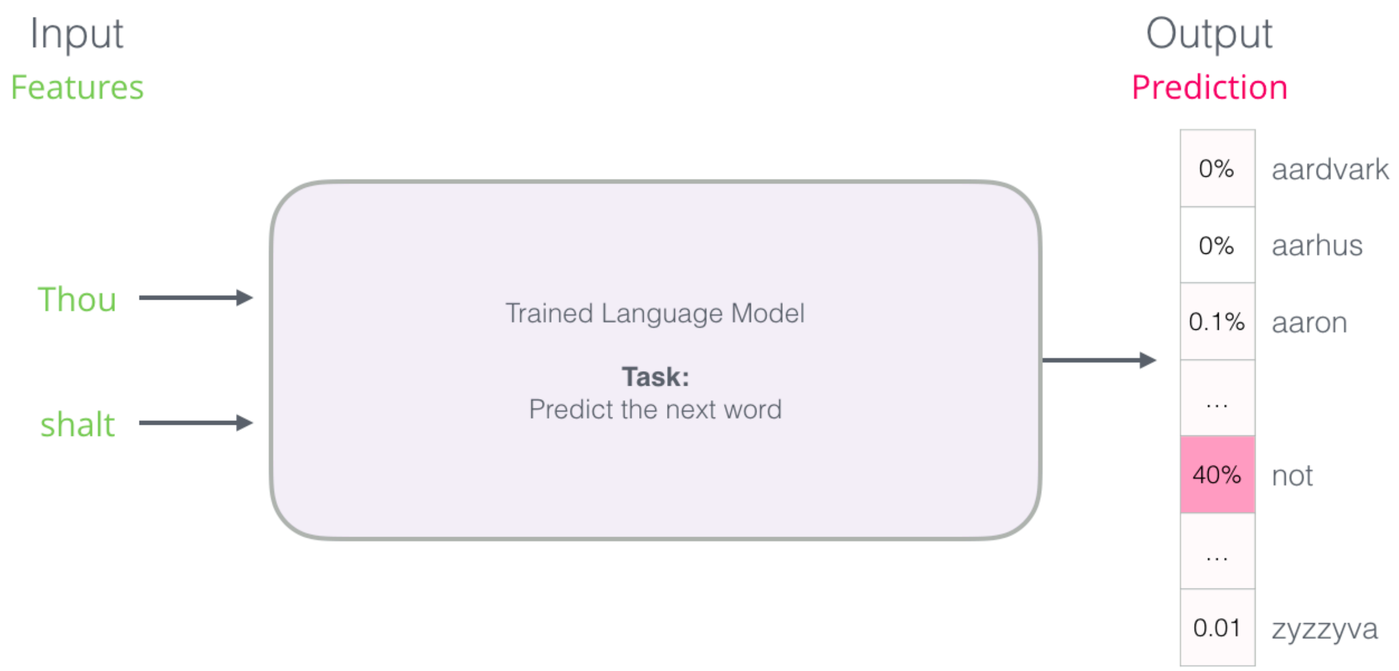
# Text Representation

## Word2Vec

### Language Modeling

*La predicción de la siguiente palabra.*

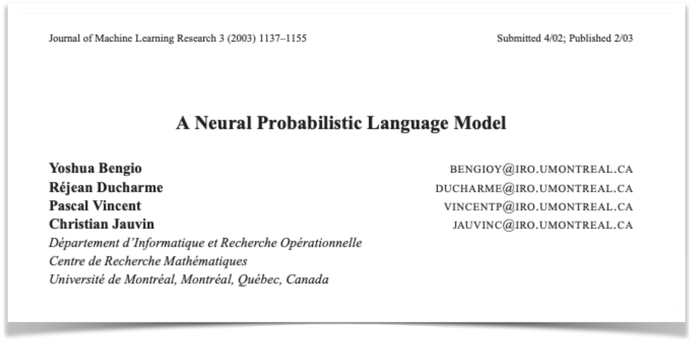
*Probabilidad para todas las palabras del vocabulario*



# Text Representation

## Word2Vec

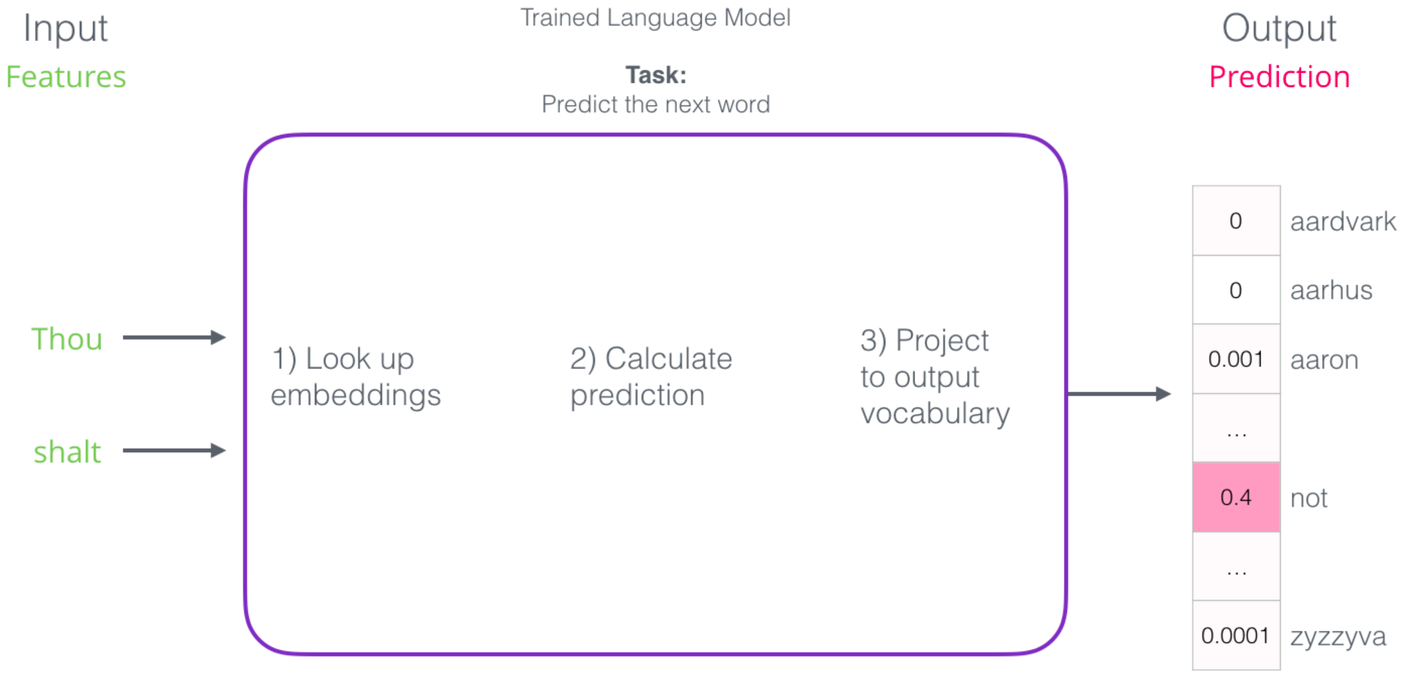
*Predicción en tres pasos, de acuerdo con Bengio 2003*



Journal of Machine Learning Research 3 (2003) 1137–1155 Submitted 4/02; Published 2/03

**A Neural Probabilistic Language Model**


**Yoshua Bengio** BENGIOY@IRO.UMONTREAL.CA
**Réjean Ducharme** DUCHARME@IRO.UMONTREAL.CA
**Pascal Vincent** VINCENTP@IRO.UMONTREAL.CA
**Christian Jauvin** JAUVINC@IRO.UMONTREAL.CA
*Département d'Informatique et Recherche Opérationnelle*
*Centre de Recherche Mathématiques*
*Université de Montréal, Montréal, Québec, Canada*

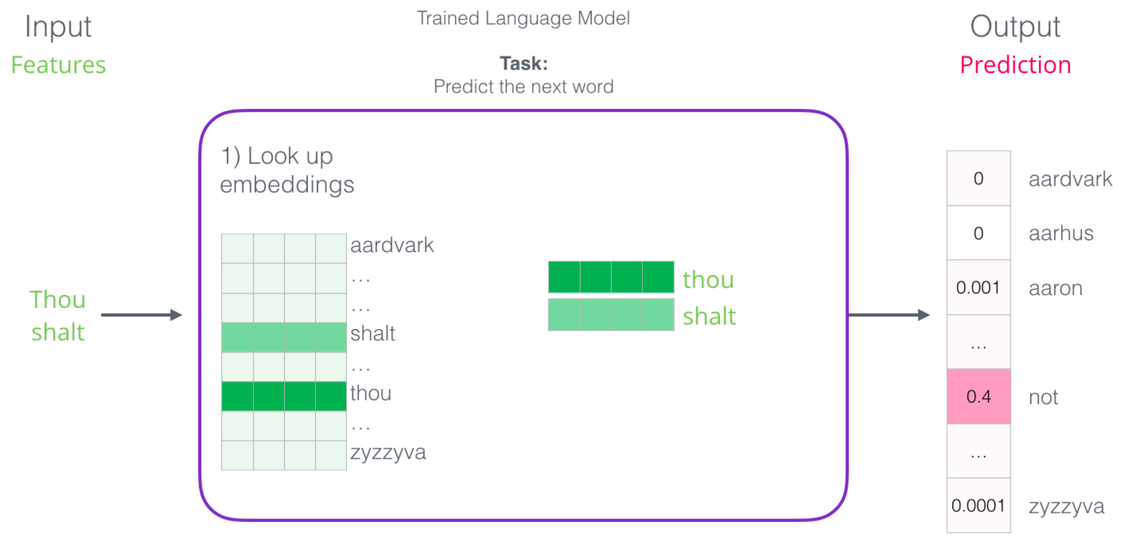
# Text Representation

## Word2Vec

### Language Modeling

*La predicción de la siguiente palabra.*

*Matriz que contiene un embedding para cada palabra en nuestro vocabulario.*




*simplemente buscamos los embeddings de la palabra de entrada y las usamos para calcular la predicción*

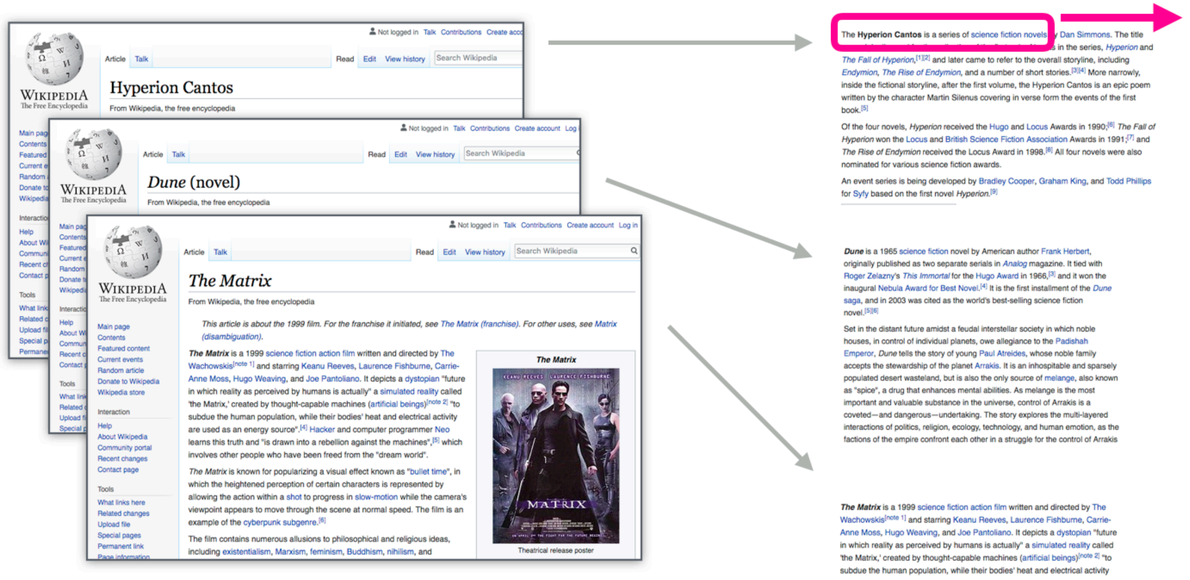
# Text Representation

## Word2Vec

## Training Language Modeling

*La predicción de la siguiente palabra.*

*Podemos entrenarlos para ejecutar texto, algo que tenemos en abundancia*




*Las palabras obtienen sus embeddings cuando observamos junto a qué otras palabras tienden a aparecer*

# Text Representation

## Word2Vec

### Training Language Modeling

*La predicción de la siguiente palabra.*



*Podemos entrenarlos para ejecutar texto, algo que tenemos en abundancia*

*1. Recibimos una gran cantidad de datos de texto (digamos, todos los artículos de Wikipedia, por ejemplo)*

*2. Tenemos una ventana (digamos, de tres palabras) que deslizamos sobre todo ese texto.*

*3. La ventana deslizante genera muestras de entrenamiento para nuestro modelo.*

*Las palabras obtienen sus embeddings cuando observamos junto a qué otras palabras tienden a aparecer*

# Text Representation

## Word2Vec

### Training Language Modeling

*La predicción de la siguiente palabra.*

*A medida que esta ventana se desliza contra el texto, generamos (virtualmente) un conjunto de datos que usamos para entrenar un modelo.*

*“Thou shalt not make a machine in the likeness of a human mind”*
*~Dune*

Thou shalt not make a machine in the likeness of a human mind

Sliding window across running text

| thou | shalt | not | make | a | machine | in | the | … |
|---|---|---|---|---|---|---|---|---|

Dataset

| input 1 | input 2 | output |
|---|---|---|
| | | |

*La ventana está en las tres primeras palabras de la oración*

# Text Representation

## Word2Vec

### Training Language Modeling

*La predicción de la siguiente palabra.*

*A medida que esta ventana se desliza contra el texto, generamos (virtualmente) un conjunto de datos que usamos para entrenar un modelo.*

Thou shalt not make a machine in the likeness of a human mind

Sliding window across running text

| thou | shalt | not | make | a | machine | in | the | … |
|---|---|---|---|---|---|---|---|---|

Dataset

| input 1 | input 2 | output |
|---|---|---|
| | | |

*Consideramos las dos primeras palabras como características y la tercera palabra como una etiqueta:*

Thou shalt not make a machine in the likeness of a human mind

Sliding window across running text

| thou | shalt | not | make | a | machine | in | the | … |
|---|---|---|---|---|---|---|---|---|

Dataset

| input 1 | input 2 | output |
|---|---|---|
| thou | shalt | not |

# Text Representation

## Word2Vec

## Training Language Modeling

*La predicción de la siguiente palabra.*

*A medida que esta ventana se desliza contra el texto, generamos (virtualmente) un conjunto de datos que usamos para entrenar un modelo.*

*Ahora hemos generado la primera muestra en el conjunto de datos que luego podemos usar para entrenar un modelo de lenguaje.*

Thou shalt not make a machine in the likeness of a human mind

Sliding window across running text

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| thou | shalt | not | make | a | machine | in | the | … |
| thou | shalt | not | make | a | machine | in | the | |

Dataset

| input 1 | input 2 | output |
|---|---|---|
| thou | shalt | not |
| shalt | not | make |

*Deslizando la ventana, a la siguiente posición.*

# Text Representation

## Word2Vec

### Training Language Modeling

*La predicción de la siguiente palabra.*



*A medida que esta ventana se desliza contra el texto, generamos (virtualmente) un conjunto de datos que usamos para entrenar un modelo.*

*Ahora hemos generado la primera muestra en el conjunto de datos que luego podemos usar para entrenar un modelo de lenguaje.*

Thou shalt not make a machine in the likeness of a human mind

Sliding window across running text

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| thou | shalt | not | make | a | machine | in | the | … |
| thou | shalt | not | make | a | machine | in | the | |
| thou | shalt | not | make | a | machine | in | the | |
| thou | shalt | not | make | a | machine | in | the | |
| thou | shalt | not | make | a | machine | in | the | |

Dataset

| input 1 | input 2 | output |
|---|---|---|
| thou | shalt | not |
| shalt | not | make |
| not | make | a |
| make | a | machine |
| a | machine | in |

*Deslizando la ventana, a la siguiente posición.*

# Text Representation

## Word2Vec

### Training
### Language Modeling

*La predicción de la siguiente palabra.*

*¿Mirando a ambos lados?*
*“Bus” y luego “red”.*

Jay was hit by a _____

Jay was hit by a _____ bus

# Text Representation

## Word2Vec

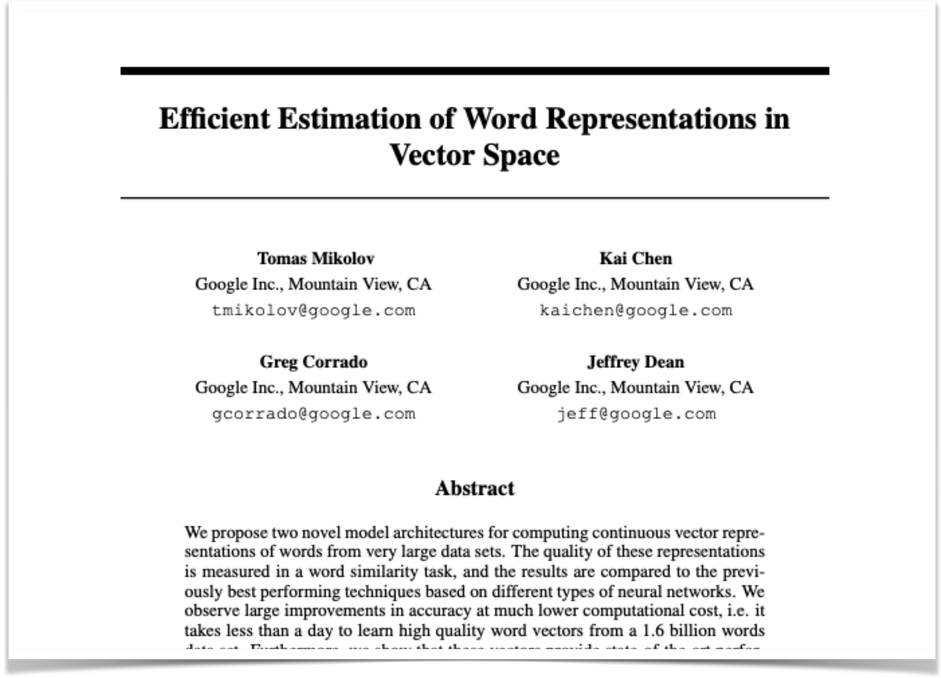
### CBOW:
### Language Modeling

*La predicción de la siguiente palabra.*

↓

*Mirando a ambos lados.*

*Efficient Estimation of Word Representations in Vector Space*



**Efficient Estimation of Word Representations in Vector Space**


**Tomas Mikolov**
Google Inc., Mountain View, CA
tmikolov@google.com

**Kai Chen**
Google Inc., Mountain View, CA
kaichen@google.com

**Greg Corrado**
Google Inc., Mountain View, CA
gcorrado@google.com

**Jeffrey Dean**
Google Inc., Mountain View, CA
jeff@google.com


**Abstract**


We propose two novel model architectures for computing continuous vector representations of words from very large data sets. The quality of these representations is measured in a word similarity task, and the results are compared to the previously best performing techniques based on different types of neural networks. We observe large improvements in accuracy at much lower computational cost, i.e. it takes less than a day to learn high quality word vectors from a 1.6 billion words

# Text Representation

## Word2Vec

### CBOW:
### Language Modeling

*La predicción de la siguiente palabra.*

↓

*Mirando a ambos lados.*

Jay was hit by a _____ bus in...

| by | a | red | bus | in |
|---|---|---|---|---|

| input 1 | input 2 | input 3 | input 4 | output |
|---|---|---|---|---|
| by | a | bus | in | red |

*CBOW: Continues Bag Of Words*

# Text Representation

## Word2Vec

### SkipGram: Language Modeling

*La predicción de la siguiente palabra.*

↓

*Pronóstico con base en la palabra actual.*

*No pronostica basándose en el contesto (palabras anteriores y posteriores)*


La palabra en la ranura verde sería la palabra de entrada, cada cuadro rosa sería una posible salida.

# Text Representation

## Word2Vec

### SkipGram: Language Modeling

*La predicción de la siguiente palabra.*

↓

*Pronóstico con base en la palabra actual.*

*No pronostica basándose en el contesto (palabras anteriores y posteriores)*

Jay was hit by a red bus in...

| by | a | red | bus | in |
|---|---|---|---|---|

| input | output |
|---|---|
| red | by |
| red | a |
| red | bus |
| red | in |

Esta ventana deslizante crea cuatro muestras separadas.

# Text Representation

## Word2Vec

### SkipGram: Language Modeling

*La predicción de la siguiente palabra.*

↓

*Pronóstico con base en la palabra actual.*

### Ventana deslizante:

Thou shalt not make a machine in the likeness of a human mind

| thou | shalt | not | make | a | machine | in | the | … |
|---|---|---|---|---|---|---|---|---|

| input word | target word |
|---|---|
| | |

Esta ventana deslizante crea cuatro muestras separadas.

# Text Representation

## Word2Vec

### SkipGram: Language Modeling

*La predicción de la siguiente palabra.*

↓

*Pronóstico con base en la palabra actual.*

### Ventana deslizante:

Thou shalt not make a machine in the likeness of a human mind

| thou | shalt | not | make | a | machine | in | the | … |
|---|---|---|---|---|---|---|---|---|

| input word | target word |
|---|---|
| not | thou |
| not | shalt |
| not | make |
| not | a |

Ahora a la siguiente posición…

# Text Representation

## Word2Vec

### SkipGram: Language Modeling

*La predicción de la siguiente palabra.*

↓

*Pronóstico con base en la palabra actual.*

### Ventana deslizante:

Thou shalt not make a machine in the likeness of a human mind

| thou | shalt | not | make | a | machine | in | the | … |
|---|---|---|---|---|---|---|---|---|

| thou | shalt | not | make | a | machine | in | the | … |
|---|---|---|---|---|---|---|---|---|

| input word | target word |
|---|---|
| not | thou |
| not | shalt |
| not | make |
| not | a |

Lo que genera cuatro registros más…

# Text Representation

## Word2Vec

### SkipGram: Language Modeling

*La predicción de la siguiente palabra.*

↓

*Pronóstico con base en la palabra actual.*

### Ventana deslizante:

Thou shalt not make a machine in the likeness of a human mind

| thou | shalt | not | make | a | machine | in | the | … |
|---|---|---|---|---|---|---|---|---|

| thou | shalt | not | make | a | machine | in | the | … |
|---|---|---|---|---|---|---|---|---|

| input word | target word |
|---|---|
| not | thou |
| not | shalt |
| not | make |
| not | a |
| make | shalt |
| make | not |
| make | a |
| make | machine |

Un par adelante veríamos algo como…

# Text Representation

## Word2Vec

### SkipGram: Language Modeling

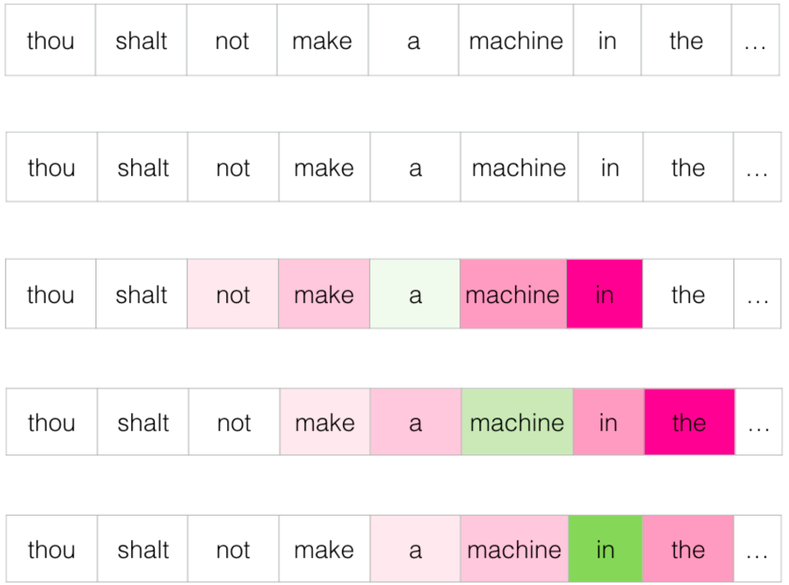
*La predicción de la siguiente palabra.*

↓

*Pronóstico con base en la palabra actual.*

### Ventana deslizante:

Thou shalt not make a machine in the likeness of a human mind

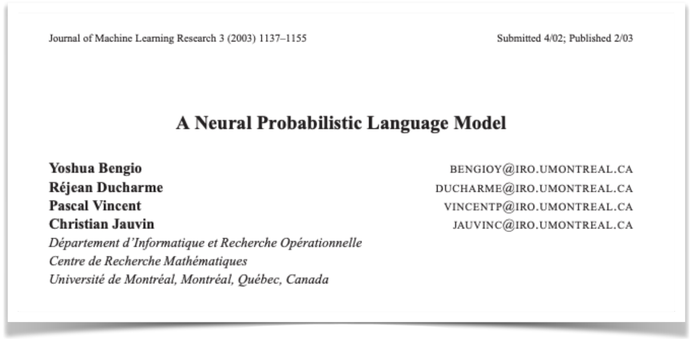



| input word | target word |
|---|---|
| not | thou |
| not | shalt |
| not | make |
| not | a |
| make | shalt |
| make | not |
| make | a |
| make | machine |
| a | not |
| a | make |
| a | machine |
| a | in |
| machine | make |
| machine | a |
| machine | in |
| machine | the |
| in | a |
| in | machine |
| in | the |
| in | likeness |
| | |

# Text Representation

## Word2Vec

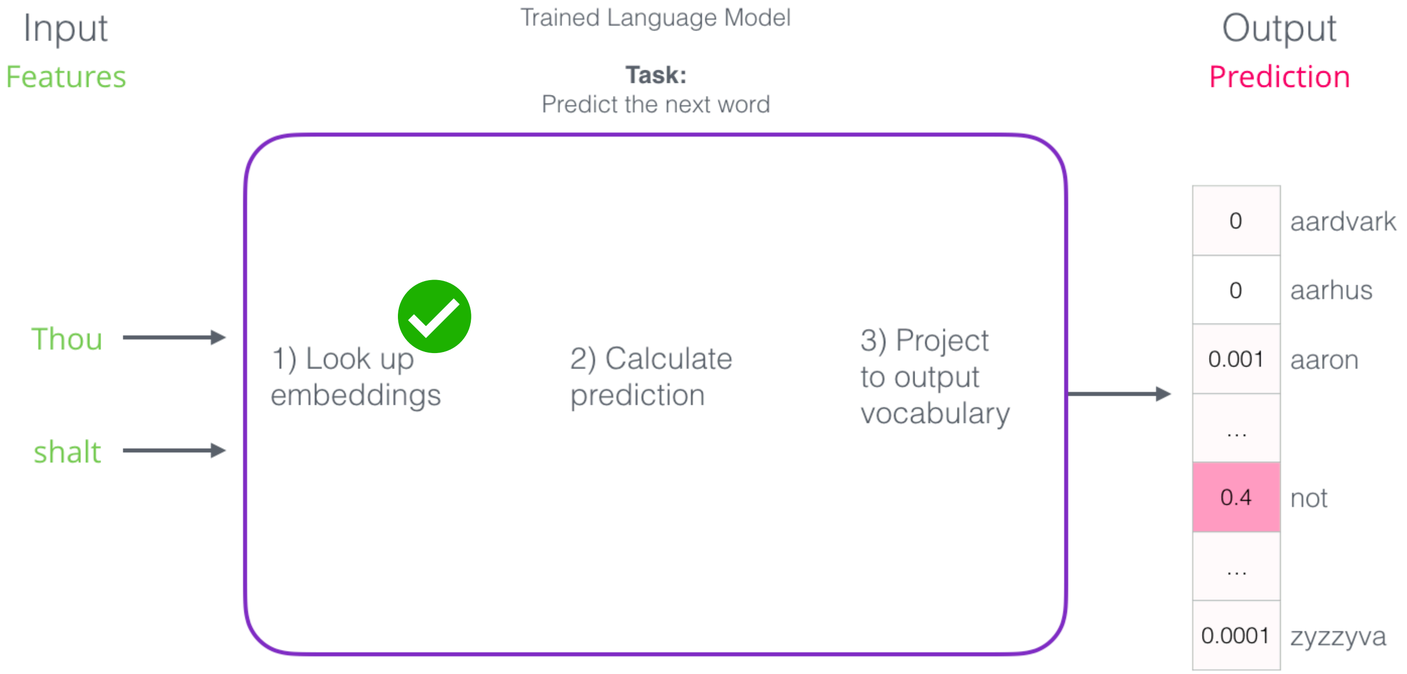
*Predicción en tres pasos, de acuerdo con Bengio 2003*



Journal of Machine Learning Research 3 (2003) 1137–1155 Submitted 4/02; Published 2/03

**A Neural Probabilistic Language Model**


**Yoshua Bengio** BENGIOY@IRO.UMONTREAL.CA
**Réjean Ducharme** DUCHARME@IRO.UMONTREAL.CA
**Pascal Vincent** VINCENTP@IRO.UMONTREAL.CA
**Christian Jauvin** JAUVINC@IRO.UMONTREAL.CA
*Département d'Informatique et Recherche Opérationnelle*
*Centre de Recherche Mathématiques*
*Université de Montréal, Montréal, Québec, Canada*

# Text Representation

## Word2Vec

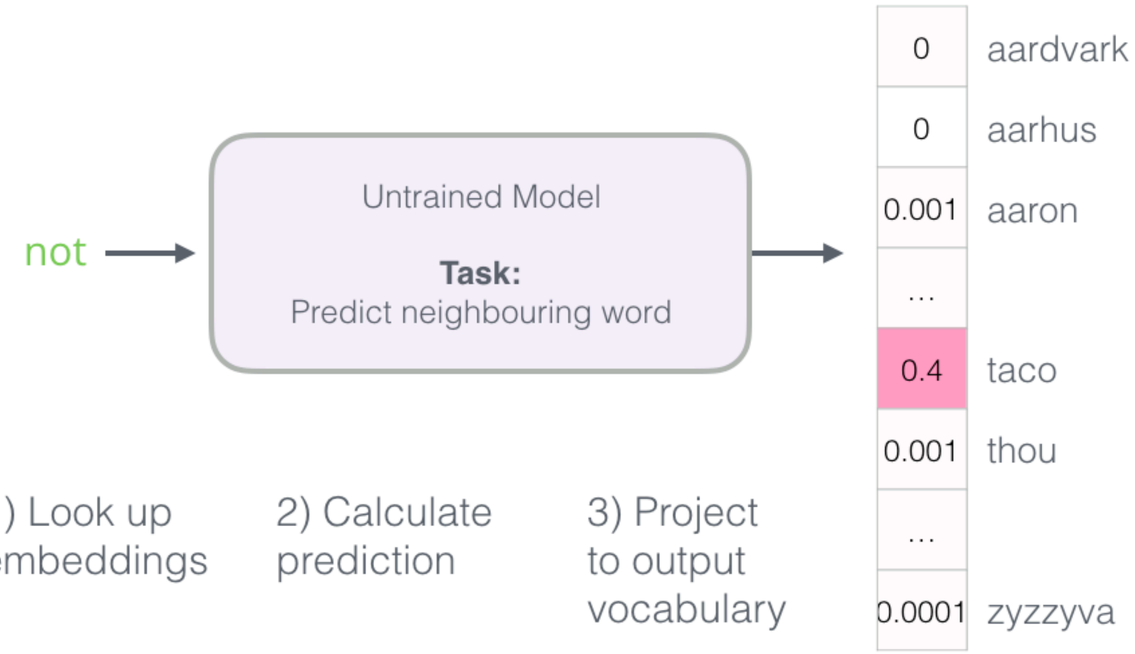
### SkipGram: Language Modeling

*La predicción de la siguiente palabra.*

↓

*Pronóstico con base en la palabra actual.*

**Entrenamiento:**



# Text Representation

## Word2Vec

### SkipGram: Language Modeling

*La predicción de la siguiente palabra.*

↓

*Pronóstico con base en la palabra actual.*

**Entrenamiento (iteración):**

*1. El modelo realiza los tres pasos y genera un vector de predicción (con una probabilidad asignada a cada palabra de su vocabulario).*

*2. Dado que el modelo no está entrenado, es seguro que su predicción será incorrecta en esta etapa.*

*3. Sabemos qué palabra debería haber adivinado: la etiqueta/celda de salida en la fila que estamos usando actualmente para entrenar el modelo*

# Text Representation

## Word2Vec

### SkipGram: Language Modeling

*La predicción de la siguiente palabra.*

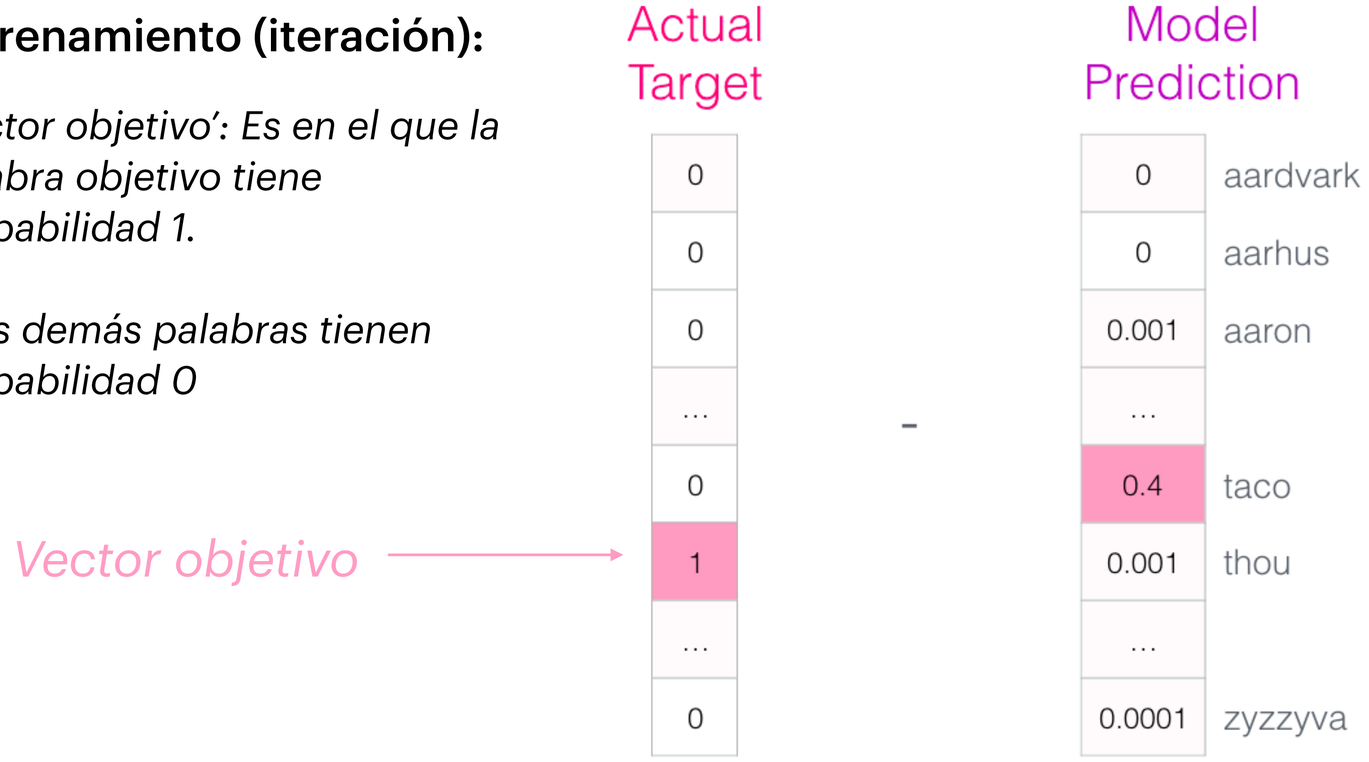
↓

*Pronóstico con base en la palabra actual.*

**Entrenamiento (iteración):**

*'Vector objetivo': Es en el que la palabra objetivo tiene probabilidad 1.*

*Y las demás palabras tienen probabilidad 0*



# Text Representation

## Word2Vec

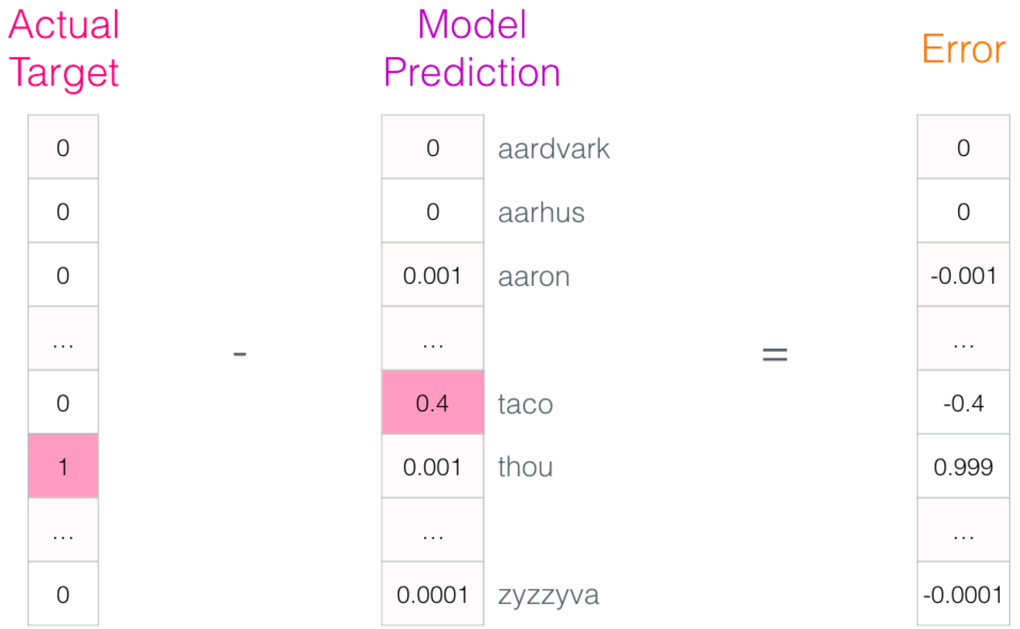
### SkipGram: Language Modeling

*La predicción de la siguiente palabra.*

↓

*Pronóstico con base en la palabra actual.*

**Entrenamiento (iteración):** *¿A que distancia estaba el modelo?*



# Text Representation

## Word2Vec

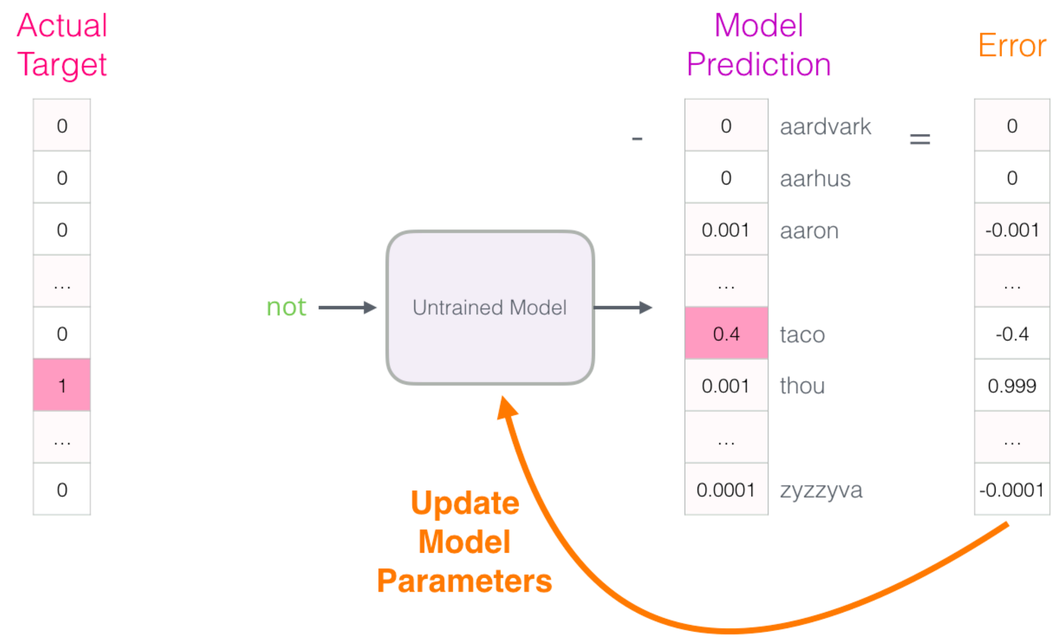
### SkipGram: Language Modeling

*La predicción de la siguiente palabra.*

↓

*Pronóstico con base en la palabra actual.*

**Entrenamiento (iteración):**

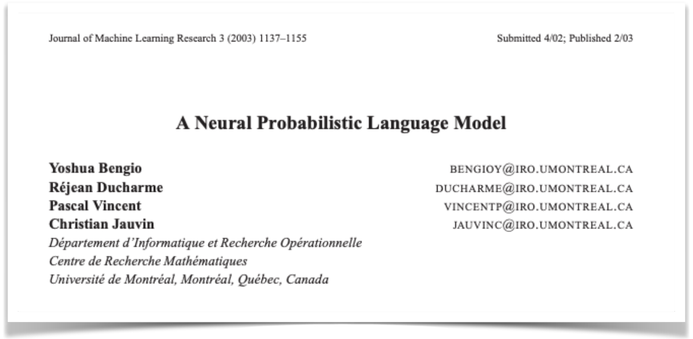



*Actualizamos el modelo con el vector de error.*

# Text Representation

## Word2Vec

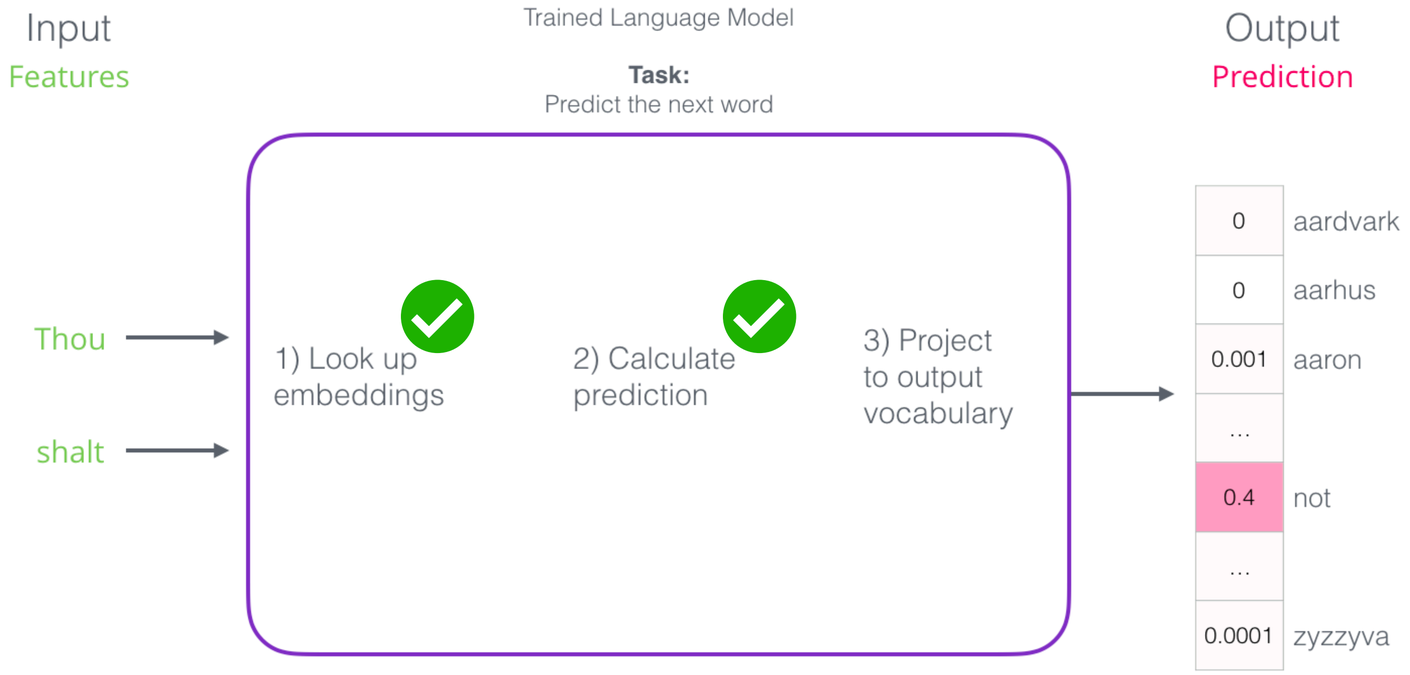
*Predicción en tres pasos, de acuerdo con Bengio 2003*



Journal of Machine Learning Research 3 (2003) 1137–1155 Submitted 4/02; Published 2/03

**A Neural Probabilistic Language Model**


**Yoshua Bengio** BENGIOY@IRO.UMONTREAL.CA
**Réjean Ducharme** DUCHARME@IRO.UMONTREAL.CA
**Pascal Vincent** VINCENTP@IRO.UMONTREAL.CA
**Christian Jauvin** JAUVINC@IRO.UMONTREAL.CA
*Département d'Informatique et Recherche Opérationnelle*
*Centre de Recherche Mathématiques*
*Université de Montréal, Montréal, Québec, Canada*

# Text Representation

## Word2Vec

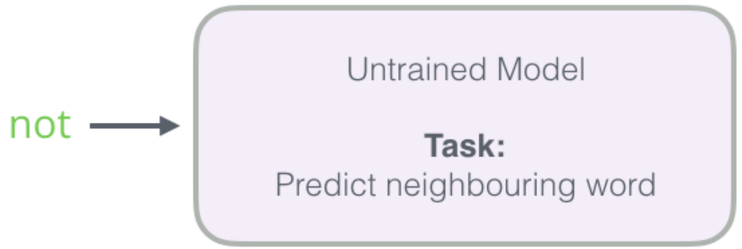
### SkipGram: Language Modeling

*La predicción de la siguiente palabra.*

↓

*Pronóstico con base en la palabra actual.*

**Project vocabulary:**




1) Look up embeddings

2) Calculate prediction

**3) Project to output vocabulary**

**[Computationally Intensive]**

# Text Representation

## Word2Vec

### SkipGram: Language Modeling

*La predicción de la siguiente palabra.*

↓

*Pronóstico con base en la palabra actual.*

**Project vocabulary:**

*Costoso computacionalmente*

*Una vez por cada muestra de entrenamiento en el conjunto de datos.*

*Decenas de millones de veces.*

# Text Representation

## Word2Vec

### SkipGram: Language Modeling

*La predicción de la siguiente palabra.*

↓

*Pronóstico con base en la palabra actual.*

**Project vocabulary:**

*1. Generar embeddings de palabras de alta calidad* → *No preocuparse por la predicción del siguiente pronóstico.*

*2. Utilizar estas incorporaciones de alta calidad para entrenar un modelo de lenguaje* → *Realizar predicciones de la siguiente palabra.*

# Text Representation

## Word2Vec

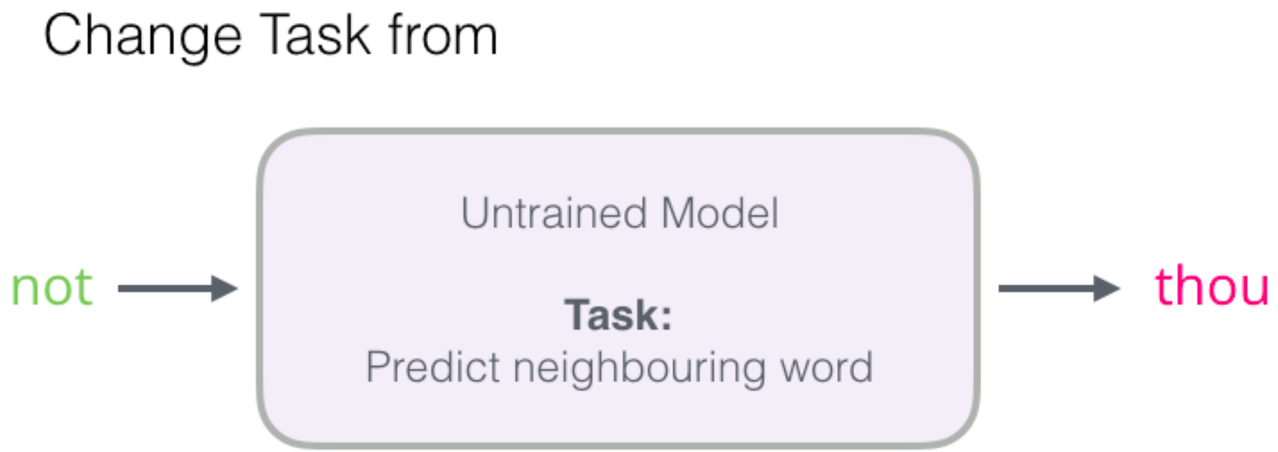
### SkipGram: Language Modeling

*La predicción de la siguiente palabra.*

↓

*Pronóstico con base en la palabra actual.*

**Project vocabulary:** *Cambiar la tarea del modelo de predecir la palabra vecina*



# Text Representation

## Word2Vec

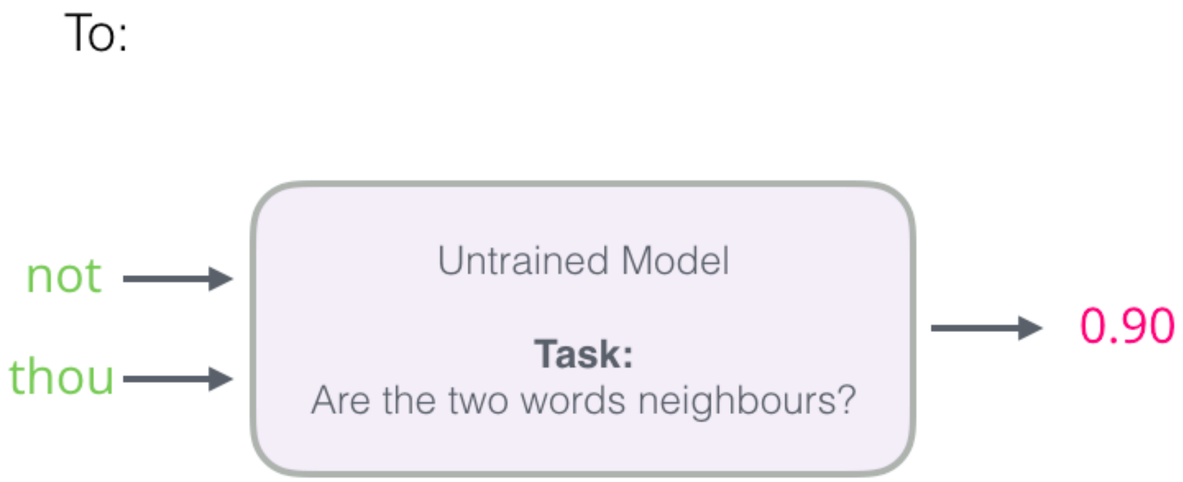
### SkipGram: Language Modeling

*La predicción de la siguiente palabra.*

↓

*Pronóstico con base en la palabra actual.*

**Project vocabulary:** *Nueva tarea, decir si son vecinos o no. 1 para "vecinos".*



# Text Representation

## Word2Vec

### SkipGram: Language Modeling

*La predicción de la siguiente palabra.*

↓

*Pronóstico con base en la palabra actual.*

**Project vocabulary:** *Cambio en nuestro conjunto de datos. Nueva etiqueta con valores 0 y 1.*

| input word | target word |
|---|---|
| not | thou |
| not | shalt |
| not | make |
| not | a |
| make | shalt |
| make | not |
| make | a |
| make | machine |
| | |

| input word | output word | target |
|---|---|---|
| not | thou | **1** |
| not | shalt | **1** |
| not | make | **1** |
| not | a | **1** |
| make | shalt | **1** |
| make | not | **1** |
| make | a | **1** |
| make | machine | **1** |
| | | |

# Text Representation

## Word2Vec

### SkipGram: Language Modeling

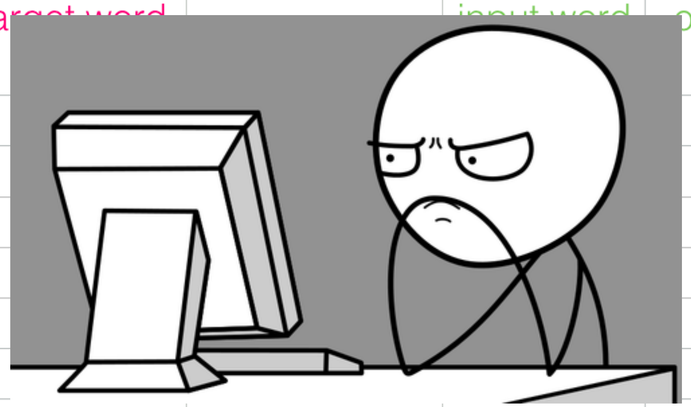
*La predicción de la siguiente palabra.*

↓

*Pronóstico con base en la palabra actual.*

**Project vocabulary:** *Cambio en nuestro conjunto de datos. Nueva etiqueta con valores 0 y 1.*

| input word | target word | input word | output word | target |
|---|---|---|---|---|
| not | | | thou | 1 |
| not | | | shalt | 1 |
| not | | | make | 1 |
| not | | | a | 1 |
| make | | | shalt | 1 |
| make | | | not | 1 |
| make | a | make | a | 1 |
| ma | | | | 1 |
| | | | | |



**¿Qué falla tiene el conjunto de datos?**

# Text Representation

## Word2Vec

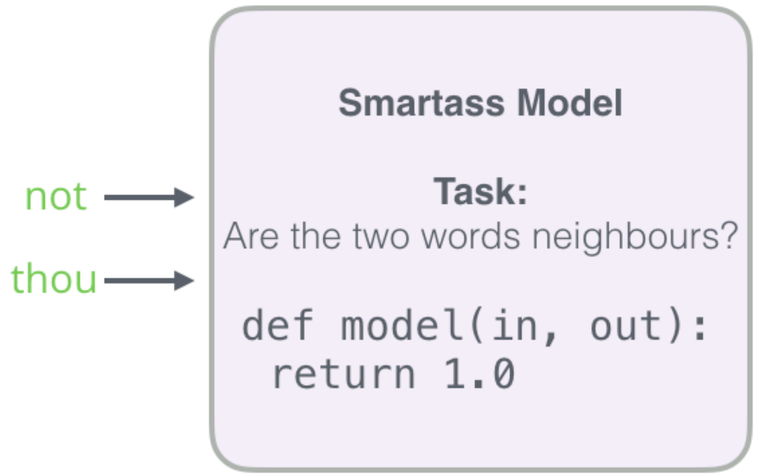
### SkipGram: Language Modeling

*La predicción de la siguiente palabra.*

↓

*Pronóstico con base en la palabra actual.*

**Project vocabulary:** *Smartass Model Problem*




*Todos nuestros objetivos son 1. El modelo puede devolver siempre 1, logrando una precisión del 100%. Pero no aprendió nada y genera embeddings basura.*

# Text Representation

## Word2Vec

### SkipGram: Language Modeling

*La predicción de la siguiente palabra.*



*Pronóstico con base en la palabra actual.*

**Project vocabulary:** *Introducir muestras negativas en el conjunto de datos*

| input word | output word | target |
|---|---|---|
| not | thou | **1** |
| not | | **0** |
| not | | **0** |
| not | shalt | **1** |
| | | |
| | | |
| not | make | **1** |
| | | |
| | | |

Negative examples

Palabras que no son vecinas.

# Text Representation

## Word2Vec

### SkipGram: Language Modeling

*La predicción de la siguiente palabra.*

↓

*Pronóstico con base en la palabra actual.*

**Project vocabulary:** *Introducir muestras negativas en el conjunto de datos*

Pick randomly from vocabulary (random sampling)

| input word | output word | target |
|---|---|---|
| not | thou | 1 |
| not | aaron | 0 |
| not | taco | 0 |
| not | shalt | 1 |
| | | |
| | | |
| not | make | 1 |
| | | |
| | | |

| Word | Count | Probability |
|---|---|---|
| aardvark | | |
| aarhus | | |
| aaron | | |
| | | |
| | | |
| taco | | |
| thou | | |
| | | |
| zyzzyva | | |

# Text Representation

## Word2Vec

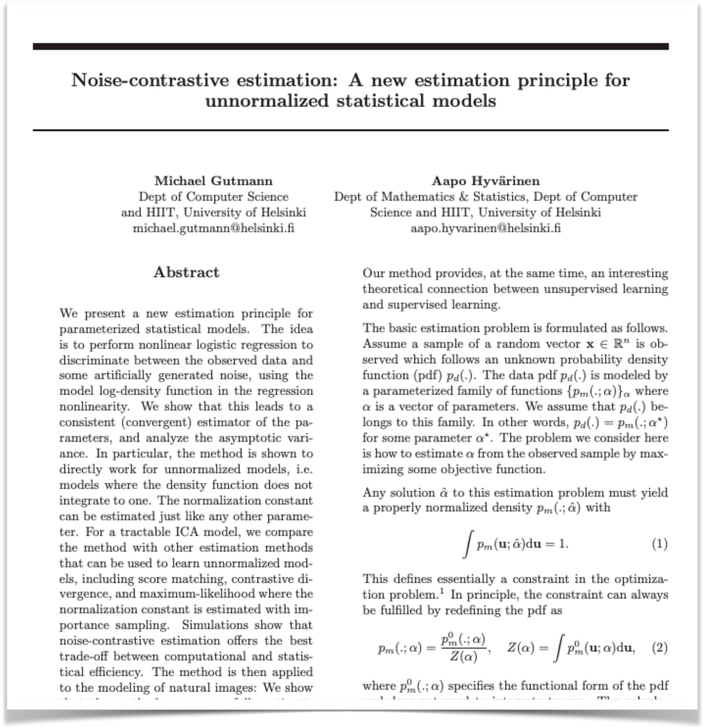
### SkipGram Negative Sampling: Language Modeling

*La predicción de la siguiente palabra.*

↓

*Pronóstico con base en la palabra actual.*

Noise-contrastive estimation: A new estimation principle for unnormalized statistical models



**Noise-contrastive estimation: A new estimation principle for unnormalized statistical models**

**Michael Gutmann**
Dept of Computer Science
and HIIT, University of Helsinki
michael.gutmann@helsinki.fi

**Aapo Hyvärinen**
Dept of Mathematics & Statistics, Dept of Computer
Science and HIIT, University of Helsinki
aapo.hyvarinen@helsinki.fi

**Abstract**

We present a new estimation principle for parameterized statistical models. The idea is to perform nonlinear logistic regression to discriminate between the observed data and some artificially generated noise, using the model log-density function in the regression nonlinearity. We show that this leads to a consistent (convergent) estimator of the parameters, and analyze the asymptotic variance. In particular, the method is shown to directly work for unnormalized models, i.e. models where the density function does not integrate to one. The normalization constant can be estimated just like any other parameter. For a tractable ICA model, we compare the method with other estimation methods that can be used to learn unnormalized models, including score matching, contrastive divergence, and maximum-likelihood where the normalization constant is estimated with importance sampling. Simulations show that noise-contrastive estimation offers the best trade-off between computational and statistical efficiency. The method is then applied to the modeling of natural images: We show

Our method provides, at the same time, an interesting theoretical connection between unsupervised learning and supervised learning.

The basic estimation problem is formulated as follows. Assume a sample of a random vector $\mathbf{x} \in \mathbb{R}^n$ is observed which follows an unknown probability density function (pdf) $p_d(.)$. The data pdf $p_d(.)$ is modeled by a parameterized family of functions $\{p_m(.;\alpha)\}_\alpha$ where $\alpha$ is a vector of parameters. We assume that $p_d(.)$ belongs to this family. In other words, $p_d(.) = p_m(.;\alpha^*)$ for some parameter $\alpha^*$. The problem we consider here is how to estimate $\alpha$ from the observed sample by maximizing some objective function.

Any solution $\hat{\alpha}$ to this estimation problem must yield a properly normalized density $p_m(.;\hat{\alpha})$ with

$$\int p_m(\mathbf{u};\hat{\alpha})\mathrm{d}\mathbf{u} = 1. \quad (1)$$

This defines essentially a constraint in the optimization problem.[1] In principle, the constraint can always be fulfilled by redefining the pdf as

$$p_m(.;\alpha) = \frac{p^0_m(.;\alpha)}{Z(\alpha)}, \quad Z(\alpha) = \int p^0_m(\mathbf{u};\alpha)\mathrm{d}\mathbf{u}, \quad (2)$$

where $p^0_m(.;\alpha)$ specifies the functional form of the pdf

# Text Representation

## Word2Vec

**SkipGram Negative Sampling: Language Modeling**

*La predicción de la siguiente palabra.*

↓

*Pronóstico con base en la palabra actual.*

Skipgram

| shalt | not | make | a | machine |
|---|---|---|---|---|

| input | output |
|---|---|
| make | shalt |
| make | not |
| make | a |
| make | machine |

Negative Sampling

| input word | output word | target |
|---|---|---|
| make | shalt | 1 |
| make | aaron | 0 |
| make | taco | 0 |

# Text Representation

## Word2Vec

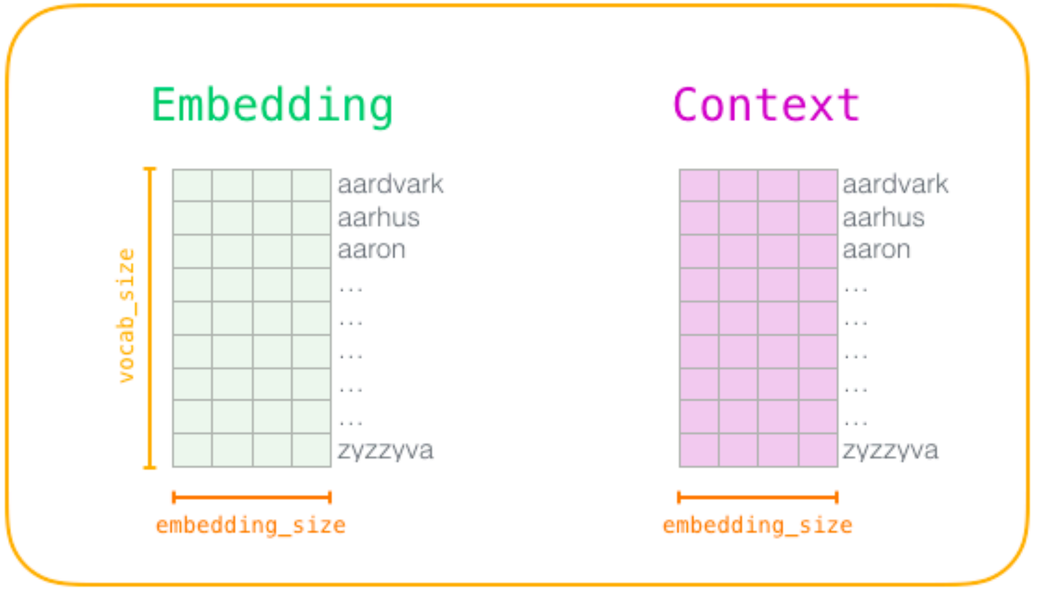
### SkipGram Negative Sampling: Language Modeling

*La predicción de la siguiente palabra.*

↓

*Pronóstico con base en la palabra actual.*

**Training:** *Definición de matrices con tamaño de vocabulario y tamaño de embedding.*




*Tamaño de vocabulario: 10.000 palabras*

*Tamaño de embedding: 300 dimensiones*

# Text Representation

## Word2Vec

### SkipGram Negative Sampling: Language Modeling

*La predicción de la siguiente palabra.*

↓

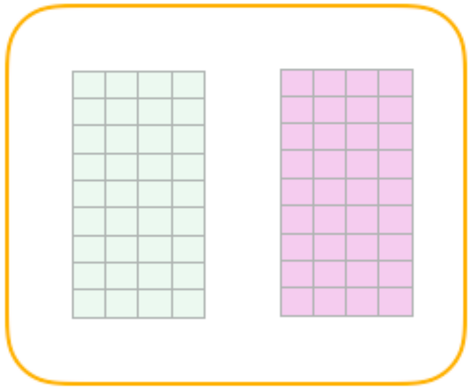
*Pronóstico con base en la palabra actual.*

**Training:** *Inicializar las matrices con valores aleatorios. Luego, se inicia el entrenamiento.*

dataset

| input word | output word | target |
|---|---|---|
| not | thou | 1 |
| not | aaron | 0 |
| not | taco | 0 |
| not | shalt | 1 |
| not | mango | 0 |
| not | finglonger | 0 |
| not | make | 1 |
| not | plumbus | 0 |
| ... | ... | ... |

model



# Text Representation

## Word2Vec

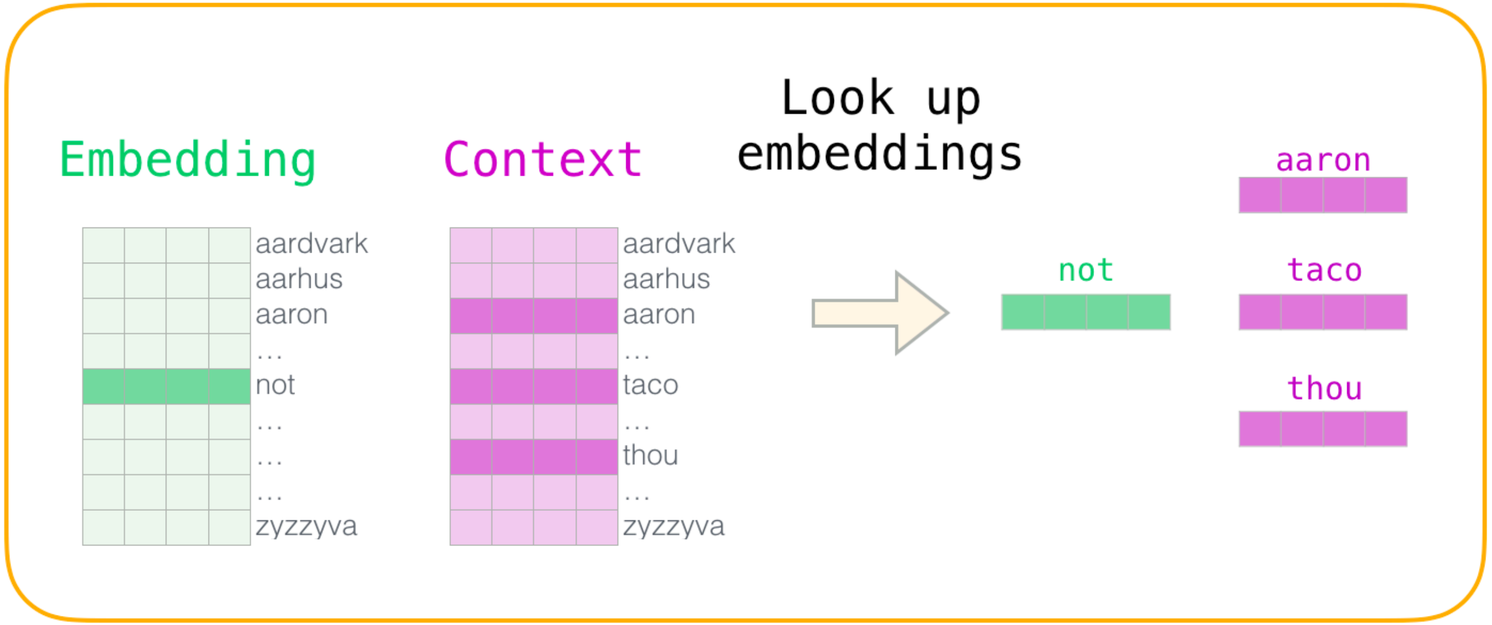
### SkipGram Negative Sampling: Language Modeling

*La predicción de la siguiente palabra.*

↓

*Pronóstico con base en la palabra actual.*

**Training:** *Buscar los embeddings en la matriz de embeddings y en la matriz de contexto.*



# Text Representation

## Word2Vec

### SkipGram Negative Sampling: Language Modeling

*La predicción de la siguiente palabra.*

↓

*Pronóstico con base en la palabra actual.*

**Training:** *Producto escalar de los embeddings de entrada con cada embedding de contexto.*

| input word | output word | target | input • output |
|---|---|---|---|
| not | thou | 1 | 0.2 |
| not | aaron | 0 | -1.11 |
| not | taco | 0 | 0.74 |

*Ahora debemos convertir estas puntuaciones en probabilidades. Positivas entre 0 y 1. Aplicamos una función sigmoide.*

| input word | output word | target | input • output | sigmoid() |
|---|---|---|---|---|
| not | thou | 1 | 0.2 | 0.55 |
| not | aaron | 0 | -1.11 | 0.25 |
| not | taco | 0 | 0.74 | 0.68 |

# Text Representation

## Word2Vec

### SkipGram Negative Sampling: Language Modeling

*La predicción de la siguiente palabra.*

↓

*Pronóstico con base en la palabra actual.*

**Training:** *Restamos las puntuaciones sigmoideas de las etiquetas objetivo/ target.*

| input word | output word | target | input • output | sigmoid() | Error |
|---|---|---|---|---|---|
| not | thou | 1 | 0.2 | 0.55 | 0.45 |
| not | aaron | 0 | -1.11 | 0.25 | -0.25 |
| not | taco | 0 | 0.74 | 0.68 | -0.68 |

# Text Representation

## Word2Vec

### SkipGram Negative Sampling: Language Modeling

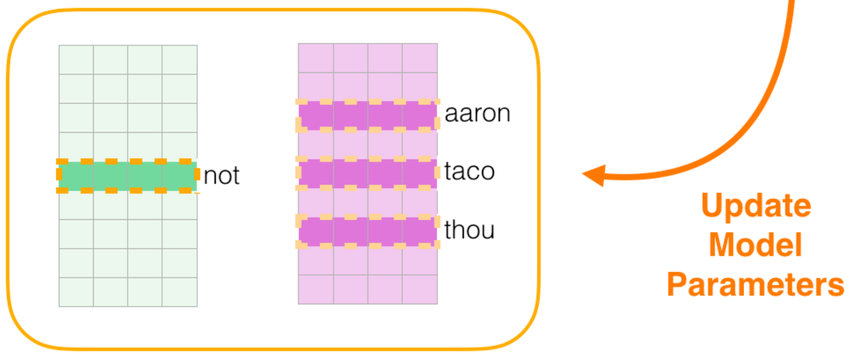
*La predicción de la siguiente palabra.*

↓

*Pronóstico con base en la palabra actual.*

**Training:** *Actualizamos con los errores el modelo, proceso de "aprendizaje" en machine learning.*

| input word | output word | target | input • output | sigmoid() | Error |
|---|---|---|---|---|---|
| not | thou | 1 | 0.2 | 0.55 | 0.45 |
| not | aaron | 0 | -1.11 | 0.25 | -0.25 |
| not | taco | 0 | 0.74 | 0.68 | -0.68 |



# Text Representation

## Word2Vec

### SkipGram Negative Sampling: Language Modeling

*La predicción de la siguiente palabra.*

↓

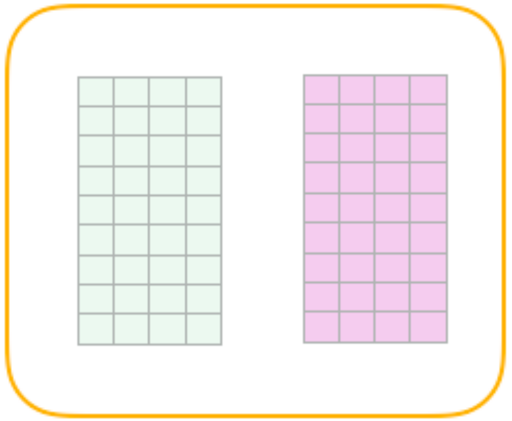
*Pronóstico con base en la palabra actual.*

**Training:** *Continuamos con la siguiente muestra de entrenamiento del conjunto de datos. Así, sucesivamente...*

dataset

| input word | output word | target |
|---|---|---|
| not | thou | 1 |
| not | aaron | 0 |
| not | taco | 0 |
| not | shalt | 1 |
| not | mango | 0 |
| not | finglonger | 0 |
| not | make | 1 |
| not | plumbus | 0 |
| ... | ... | ... |

model



# Text Representation

## Word2Vec

### SkipGram Negative Sampling: Language Modeling

*La predicción de la siguiente palabra.*



*Pronóstico con base en la palabra actual.*

### Finalizando:

*1. Las incorporaciones continúan mejorándose mientras recorremos todo nuestro conjunto de datos varias veces.*

*2. Al finalizar entrenamiento, debemos descartar la matriz de contexto y usar la matriz de embeddings (embeddings pre-entrenados)*

# Text Representation

## Word2Vec

**Hiperparámetros importantes**

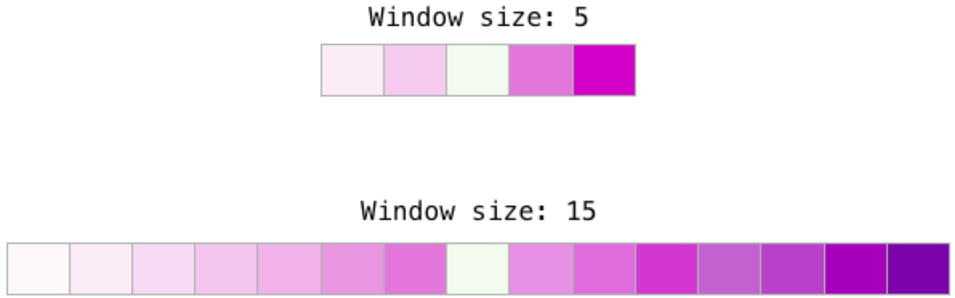
### SkipGram Negative Sampling: Language Modeling

*La predicción de la siguiente palabra.*

↓

*Pronóstico con base en la palabra actual.*

### Tamaño de ventana:




*1. Tamaño pequeños (2-15): score altos de similitud entre dos embeddings indica que las las palabras son intercambiables (antónimos: bueno y malo en contexto similares).*

*2. Tamaño más grande (15-50+): conducen a embeddings donde la similitud es más indicación de relación de las palabras.*

# Text Representation

## Word2Vec

**Hiperparámetros importantes**

### SkipGram Negative Sampling: Language Modeling

*La predicción de la siguiente palabra.*

↓

*Pronóstico con base en la palabra actual.*

### Número de muestras negativas:

Negative samples: 2

| input word | output word | target |
|---|---|---|
| make | shalt | 1 |
| make | aaron | 0 |
| make | taco | 0 |

Negative samples: 5

| input word | output word | target |
|---|---|---|
| make | shalt | 1 |
| make | aaron | 0 |
| make | taco | 0 |
| make | finglonger | 0 |
| make | plumbus | 0 |
| make | mango | 0 |

*1. El artículo original prescribe que entre 5 y 20 son un buen número de muestras negativas.*

*2. También indica que 2-5 parece ser suficiente cuando se tiene un conjunto de datos lo suficientemente grande.*