

# Procesamiento de Lenguaje Natural

---

## Representación de Texto

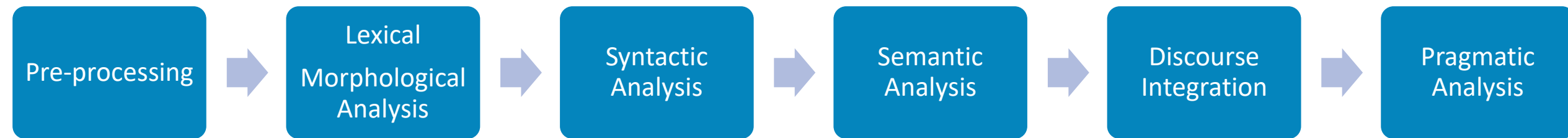
# Qué vamos a ver hoy?

## Representación de texto

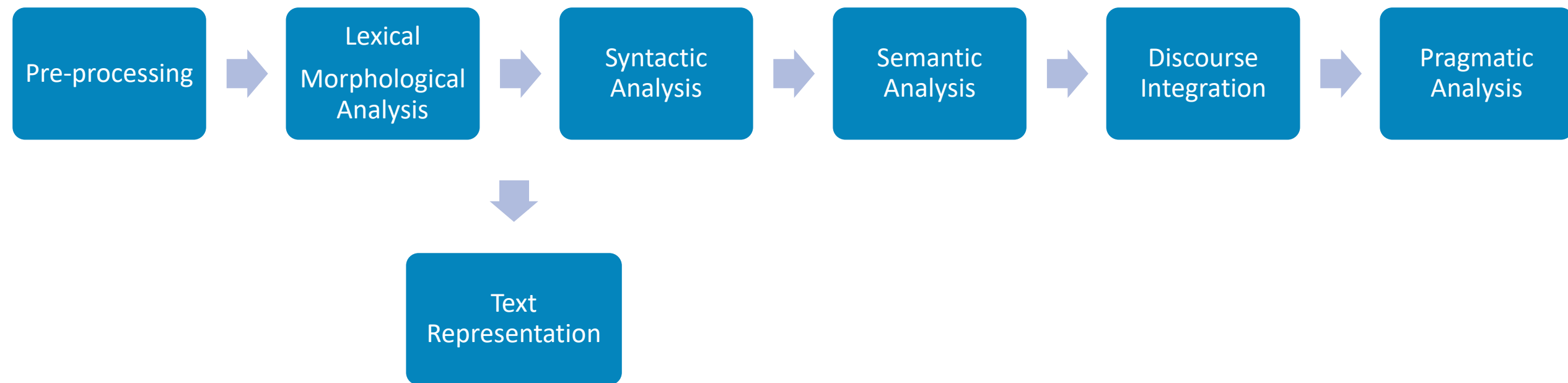
- Representación de texto.
- Representaciones tradicionales.
- Ponderación de características.
- Representaciones más avanzadas.
- Semejanza de texto.

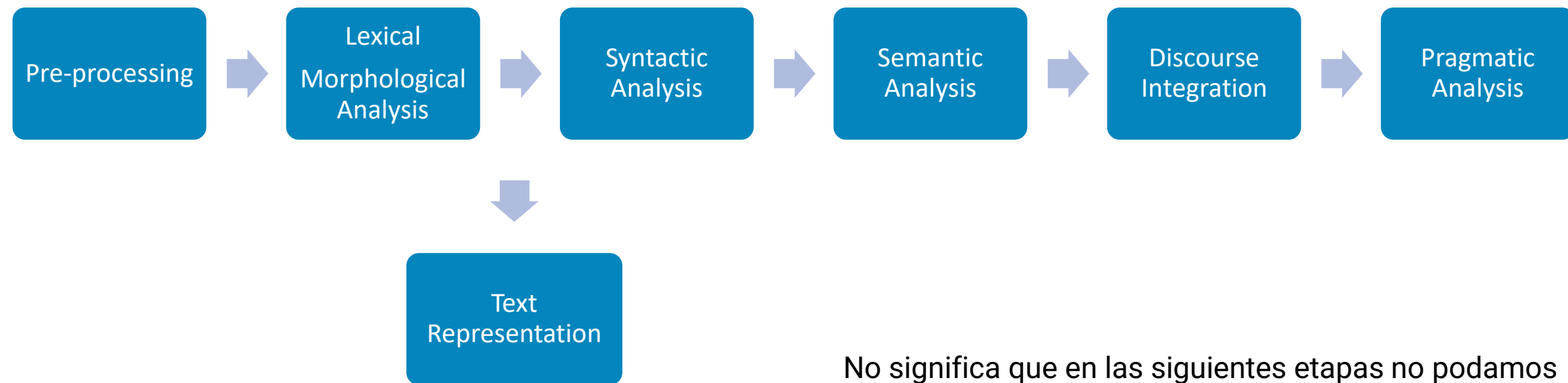


# Workflow NLP



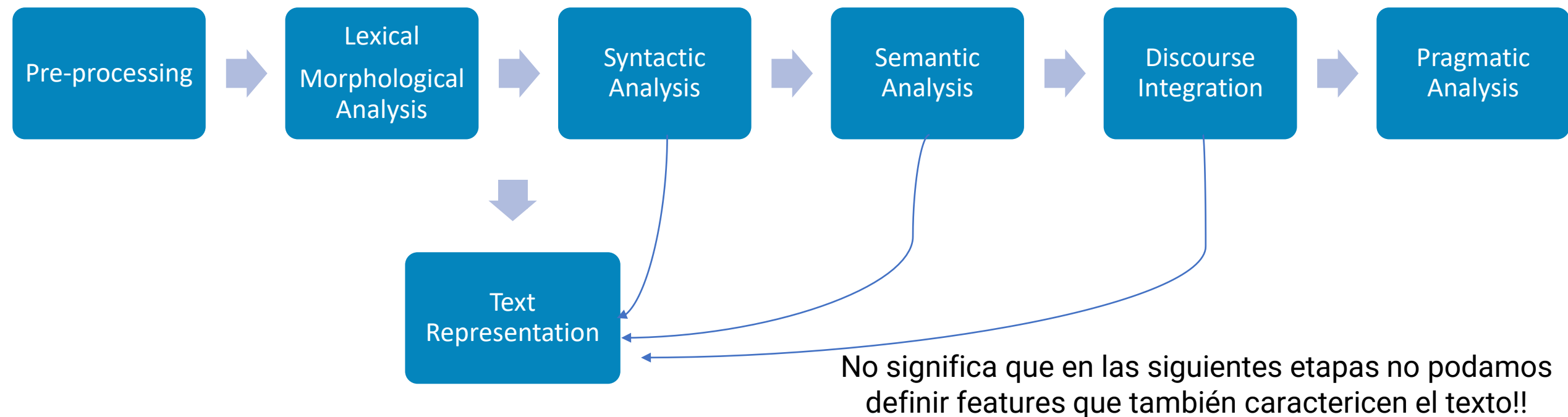
# Workflow NLP





No significa que en las siguientes etapas no podamos definir features que también caractericen el texto!!

# Workflow NLP

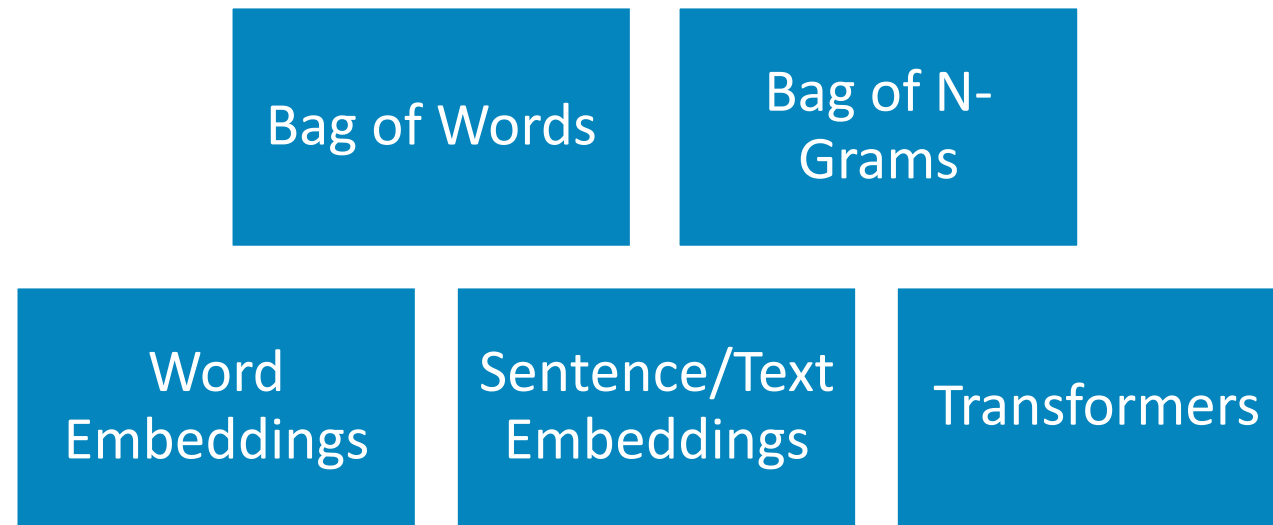




# Representación de texto

- Vimos como procesar y limpiar el texto.
- Sin embargo, no es suficiente. Los modelos de machine learning NO pueden comprender el texto de forma directa.
  - Hay que transformarlos a una representación numérica.

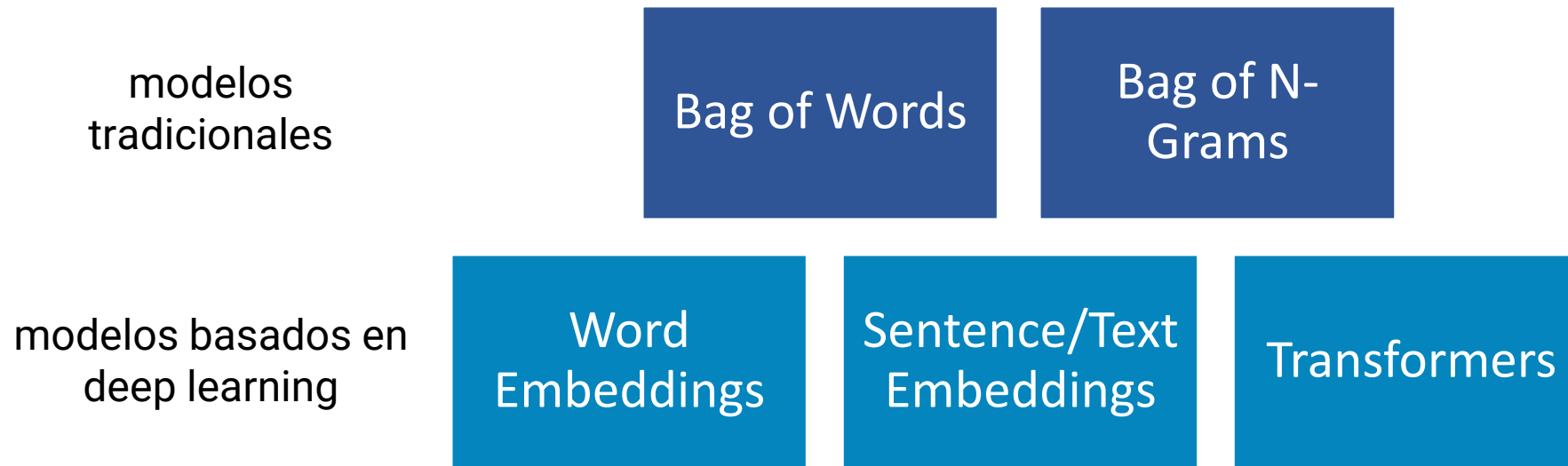
**Cómo representar el texto de forma que las técnicas y modelos de machine learning puedan comprenderlos?**



# Representación de texto

- Vimos como procesar y limpiar el texto.
- Sin embargo, no es suficiente. Los modelos de machine learning NO pueden comprender el texto de forma directa.
  - Hay que transformarlos a una representación numérica.

**Cómo representar el texto de forma que las técnicas y modelos de machine learning puedan comprenderlos?**





# Representación de texto

## Modelos Tradicionales Vs Deep Learning

### Modelos Tradicionales

- De basan en la parte léxica, es decir, en las palabras que componen el texto, su frecuencia y, ocasionalmente, algunas de las palabras inmediatamente a su alrededor.
- Al considerar aspectos solo léxicos, pierden aspectos relacionados al orden y secuencia.
- Propician el curse of dimensionality.
  - Tenemos muchas palabras que no aparecen en una porción significativa de los textos que estamos analizando.
- Útiles para muchas tareas, pero no son suficientes cuando aparecen cuestiones semánticas.

### Modelos de Deep Learning

- Transforman el texto original en una representación semántica de la palabra y su contexto.
  - Por ejemplo, con qué palabras se relaciona frecuentemente la palabra X?
- Capturan el significado, relaciones semánticas y los diferentes contextos en los que las palabras (y sus diferentes acepciones) son utilizadas.
- Básicamente transforman cada palabra en un vector numérico que captura su semántica.



Bag of  
Words

Bag of N-  
Grams



# Modelos Tradicionales

- Basados en estadísticas.
- “Bolsas” no estructuradas de palabras.
- Simples y efectivas.
- Pierden:
  - Semántica.
  - Estructura.
  - Contexto más allá de una o dos palabras cercanas.



# Modelos Tradicionales

- Basados en estadísticas.
- “Bolsas” no estructuradas de palabras.
- Simples y efectivas.
- Pierden:
  - Semántica.
  - Estructura.
  - Contexto más allá de una o dos palabras cercanas.

The sky is beautiful, and I love the blue sky.

Love this beautiful day.

The sky is very beautiful today.



# Modelos Tradicionales

- Basados en estadísticas.
  - “Bolsas” no estructuradas de palabras.
  - Simples y efectivas.
  - Pierden:
    - Semántica.
    - Estructura.
    - Contexto más allá de una o dos palabras cercanas.
1. [sky, beautiful, love, blue]
  2. [love, beautiful, day]
  3. [sky, very, beautiful, today]



- La representación de texto no estructurada más simple.
- Cada texto es literalmente una bolsa de sus palabras sin considerar orden, secuencia o gramática.
- Representa los textos como vectores numéricos donde cada dimensión representa una feature específica.
  - Feature == palabra.
- El valor para cada feature se puede corresponder con:
  - *Binaria*. Aparece la feature en el texto o no.
  - *Frecuencia*. Cuántas veces aparece en el texto.
  - *Ponderada*. Del total de palabras, qué porcentaje representan las apariciones?





# Modelos Tradicionales

## Bag of Words

- La representación de texto no estructurada más simple.
- Cada texto es literalmente una bolsa de sus palabras sin considerar orden, secuencia o gramática.
- Representa los textos como vectores numéricos donde cada dimensión representa una feature específica.
  - Feature == palabra.
- El valor para cada feature se puede corresponder con:
  - *Binaria*. Aparece la feature en el texto o no.

1. [sky, beautiful, love, blue]
2. [love, beautiful, day]
3. [sky, very, beautiful, today]

	sky	beautiful	blue	love	day	very	today
1	1	1	1	1			
2		1		1	1		
3	1	1				1	1



# Modelos Tradicionales

## Bag of Words

- La representación de texto no estructurada más simple.
- Cada texto es literalmente una bolsa de sus palabras sin considerar orden, secuencia o gramática.
- Representa los textos como vectores numéricos donde cada dimensión representa una feature específica.
  - Feature == palabra.
- El valor para cada feature se puede corresponder con:
  - *Frecuencia*. Cuántas veces aparece en el texto.

1. [sky, beautiful, love, blue]
2. [love, beautiful, day]
3. [sky, very, beautiful, today]

	sky	beautiful	blue	love	day	very	today
1	2	1	1	1			
2		1		1	1		
3	1	1				1	1



# Modelos Tradicionales

## Bag of Words

- La representación de texto no estructurada más simple.
- Cada texto es literalmente una bolsa de sus palabras sin considerar orden, secuencia o gramática.
- Representa los textos como vectores numéricos donde cada dimensión representa una feature específica.
  - Feature == palabra.
- El valor para cada feature se puede corresponder con:
  - *Frecuencia relativa*. Del total de palabras, qué porcentaje representan las apariciones?

1. [sky, beautiful, love, blue]
2. [love, beautiful, day]
3. [sky, very, beautiful, today]

	sky	beautiful	blue	love	day	very	today
1	2/5	1/5	1/5	1/5			
2		1/3		1/3	1/3		
3	1/4	1/4				1/4	1/4



## Bag of Words

- La representación de texto no estructurada más simple.
- Cada texto es literalmente una bolsa de sus palabras sin considerar orden, secuencia o gramática.
- Representa los textos como vectores numéricos donde cada dimensión representa una feature específica.
  - Feature == palabra.
- El valor para cada feature se puede corresponder con:
  - *Frecuencia relativa*. Del total de palabras, qué porcentaje representan las apariciones?

1. [sky, beautiful, love, blue]
2. [love, beautiful, day]
3. [sky, very, beautiful, today]

	sky	beautiful	blue	love	day	very	today
1	2/5	1/5	1/5	1/5			
2		1/3		1/3	1/3		
3	1/4	1/4				1/4	1/4

Vamos a ver  
otras  
alternativas!



## Bag of N-Grams

- Una palabra es un token simple, también llamado “unigram” o “1-gram”.
- Un N-gram es una colección de tokens que aparecen de forma continua en una oración.
  - Bi-gramas n-grams de orden 2.
  - Tri-gramas n-grams de orden 3.
  - ...
- Bag of N-grams es una extensión de Bag of Words.
- En este caso, las features representan la secuencia de n-grams.



- Una palabra es un token simple, también llamado “unigram” o “1-gram”.
  - Un N-gran es una colección de tokens que aparecen de forma continua en una oración.
    - Bi-gramas n-grams de orden 2.
    - Tri-gramas n-grams de orden 3.
    - ...
  - Bag of N-grams es una extensión de Bag of Words.
  - En este caso, las features representan la secuencia de n-grams.
1. [sky, beautiful, love, blue]

2. [love, beautiful, day]

3. [sky, very, beautiful, today]

	sky beautiful	beautiful love	love blue	blue sky	love beautiful	beautiful day	sky very	very beautiful	beautiful today
1	1	1	1	1					
2					1	1			
3							1	1	1



# Ponderación de características

- Binaria, frecuencia, frecuencia relativa son formas de ponderación (o pesado) de características.

Los métodos de ponderación intentan asignar pesos apropiados a las características (términos) de forma que las más “**importantes**” reciben mayores pesos en la representación



# Ponderación de características

- Binaria, frecuencia, frecuencia relativa son formas de ponderación (o pesado) de características.

Los métodos de ponderación intentan asignar pesos apropiados a las características (términos) de forma que las más “**importantes**” reciben mayores pesos en la representación

Intentar mejorar la performance de tareas como clasificación

Cómo definimos la importancia?



# Ponderación de características

- Binaria, frecuencia, frecuencia relativa son formas de ponderación (o pesado) de características.

Los métodos de ponderación intentan asignar pesos apropiados a las características (términos) de forma que las más **“importantes”** reciben mayores pesos en la representación

Intentar mejorar la performance de tareas como clasificación

Cómo definimos la importancia?

No supervisados

Supervisados



# Ponderación de características

- Binaria, frecuencia, frecuencia relativa son formas de ponderación (o pesado) de características.

Los métodos de ponderación intentan asignar pesos apropiados a las características (términos) de forma que las más **“importantes”** reciben mayores pesos en la representación

Intentar mejorar la performance de tareas como clasificación

Cómo definimos la importancia?

No supervisados

Independiente de si los textos pertenecen o no a una clase

Supervisados

Considera la clases para definir la ponderación.  
Una misma característica puede tener ponderaciones distintas de acuerdo a la clase a la que pertenece el texto.



# Ponderación de características

- Binaria, frecuencia, frecuencia relativa son formas de ponderación (o pesado) de características.

Los métodos de ponderación intentan asignar pesos apropiados a las características (términos) de forma que las más **“importantes”** reciben mayores pesos en la representación

Intentar mejorar la performance de tareas como clasificación

Suelen usarse también como técnicas de selección de características

Cómo definimos la importancia?

No supervisados

Independiente de si los textos pertenecen o no a una clase

Supervisados

Considera la clases para definir la ponderación.  
Una misma característica puede tener ponderaciones distintas de acuerdo a la clase a la que pertenece el texto.



# Ponderación de características

## No supervisado

- No requieren conocer la clase a la que pertenecen los textos.
- La mayoría de los métodos se basan en la frecuencia de aparición de los términos (TF).
  - Suelen funcionar adecuadamente con textos “tradicionales”.
  - No suelen ser adecuados con textos con longitud limitada, ralos y ruido, pero son comunmente utilizados.

Term  
Frequency

Document  
Frequency

TF-IDF

Term  
Variance

Term  
Strength





# Ponderación de características

## No supervisado

- No requieren conocer la clase a la que pertenecen los textos.
- La mayoría de los métodos se basan en la frecuencia de aparición de los términos (TF).
  - Suelen funcionar adecuadamente con textos “tradicionales”.
  - No suelen ser adecuados con textos con longitud limitada, ralos y ruido, pero son comunmente utilizados.

Term  
Frequency

Document  
Frequency

De DF deriva Inverse Document Frequency (IDF).  
Valores altos para términos poco frecuentes,  
bajos para términos muy frecuentes.

- Los más sencillos.
- Cuentan la cantidad de veces que el término aparece en el texto, o la cantidad de textos en el que aparece el término.
- Pueden calcularse sus variants relativas.
- Asume que los términos con frecuencias altas pueden ser irrelevantes.
- Términos con bajo DF son considerados relevantes.
- Simple, escala con grandes volúmenes de textos.
  - Performance similar a métodos más complejos.



# Ponderación de características

## No supervisado

- No requieren conocer la clase a la que pertenecen los textos.
- La mayoría de los métodos se basan en la frecuencia de aparición de los términos (TF).
  - Suelen funcionar adecuadamente con textos “tradicionales”.
  - No suelen ser adecuados con textos con longitud limitada, ralos y ruido, pero son comunmente utilizados.
- Asigna valores altos a términos con frecuencias bajas y que mantienen distribuciones no uniformes.
- Por el contrario, asigna valores bajos a términos que ocurren de manera uniforme en los textos.

Term  
Variance



# Ponderación de características

## No supervisado → Term Strength

- Estima la importancia de los términos considerando cuán comúnmente es que aparezcan en textos similares o relacionados.
- Se basa en clustering, asumiendo que los textos que comparten términos se encuentran relacionados y que los términos que se encuentran en ambos documentos son relevantes.
- Dos formulaciones:

$$TS(t) = P(t \in y \mid t \in x)$$

- Para pares de textos relacionados, la probabilidad de que aparezca en uno de los textos dado que apareció en el otro.
- Requiere definir cuándo dos textos se encuentran relacionados.
  - Cuál es el mínimo de términos compartidos para que dos textos estén relacionados?
  - Por ejemplo, como mínimo, el promedio de los overlaps.



# Ponderación de características

## No supervisado → Term Strength

- Estima la importancia de los términos considerando cuán comúnmente es que aparezcan en textos similares o relacionados.
- Se basa en clustering, asumiendo que los textos que comparten términos se encuentran relacionados y que los términos que se encuentran en ambos documentos son relevantes.
- Dos formulaciones:

$$TS(t) = P(t \in y \mid t \in x)$$

$$TS(t) = \frac{\# \text{ pares donde } t \text{ aparece en } x \text{ e } y}{\# \text{ pares donde } t \text{ aparece solo en uno de los textos}}$$



# Ponderación de características

No supervisado → TF-IDF

- Las ponderaciones TF están basadas en estadísticas calculadas sobre cada uno de los textos de forma individual.
  - Puede haber términos que aparecen en todos los documentos.
  - Esos términos pueden parecer más importantes que otros que aparecen en pocos documentos.
- Palabras que no ocurren en todos los documentos pueden ser relevantes para posteriores tareas.
  - Por ejemplo, clasificación.

$$TF - IDF = TF * IDF$$

*Term Frequency*  
Frecuencia de la  
palabra en el texto

*Inverse Document Frequency*  
Inverso de la "cantidad de  
documentos en los que aparece  
la palabra" con scaling  
logaritmico



# Ponderación de características

No supervisado → TF-IDF

- Las ponderaciones TF están basadas en estadísticas calculadas sobre cada uno de los textos de forma individual.
  - Puede haber términos que aparecen en todos los documentos.
  - Esos términos pueden parecer más importantes que otros que aparecen en pocos documentos.
- Palabras que no ocurren en todos los documentos pueden ser relevantes para posteriores tareas.
  - Por ejemplo, clasificación.

$$TF - IDF = TF * IDF$$

Al igual que las otras  
features numéricas,  
esta también se  
puede normalizar.

*Term Frequency*  
Frecuencia de la  
palabra en el texto

*Inverse Document Frequency*  
Inverso de la “cantidad de  
documentos en los que aparece  
la palabra” con scaling  
logaritmico





# Ponderación de características

No supervisado → TF-IDF

- Las ponderaciones TF están basadas en estadísticas calculadas sobre cada uno de los textos de forma individual.
  - Puede haber términos que aparecen en todos los documentos.
  - Esos términos pueden parecer más importantes que otros que aparecen en pocos documentos.
- Palabras que no ocurren en todos los documentos pueden ser relevantes para posteriores tareas.
  - Por ejemplo, clasificación.

$$TF - IDF = \frac{TF * IDF}{||TF * IDF||}$$

Al igual que las otras  
features numéricas,  
esta también se  
puede normalizar.



# Ponderación de características

No supervisado → TF-IDF

- Las ponderaciones TF están basadas en estadísticas calculadas sobre cada uno de los textos de forma individual.
  - Puede haber términos que aparecen en todos los documentos.
  - Esos términos pueden parecer más importantes que otros que aparecen en pocos documentos.
- Palabras que no ocurren en todos los documentos pueden ser relevantes para posteriores tareas.
  - Por ejemplo, clasificación.

$$TF - IDF = \frac{TF * IDF}{||TF * IDF||}$$

Al igual que las otras  
features numéricas,  
esta también se  
puede normalizar.

También se pueden aplicar correcciones para  
evitar divisiones por 0. Generalmente, se le  
suma 1 a las frecuencias.



# Ponderación de características

No supervisado → TF-IDF

- Las ponderaciones TF están basadas en estadísticas calculadas sobre cada uno de los textos de forma individual.
  - Puede haber términos que aparecen en todos los documentos.
  - Esos términos pueden parecer más importantes que otros que aparecen en pocos documentos.
- Palabras que no ocurren en todos los documentos pueden ser relevantes para posteriores tareas.
  - Por ejemplo, clasificación.

	sky	beautiful	blue	love	day	very	today
1	0.74	0.29	0.49	0.37			
2		0.43		0.55	0.72		
3	0.44	0.35				0.58	0.58

1. [sky, beautiful, love, blue]
2. [love, beautiful, day]
3. [sky, very, beautiful, today]



# Ponderación de características

## Supervisado

- Requieren conocer la clase a la que pertenecen los textos.
- La mayoría de los métodos se basan en combinaciones de TF con métricas supervisadas.  $TF * Metrica$
- Las metricas suelen considerar cálculos estadísticos.
- Pueden calcularse a partir de la relevancia dada por los modelos de clasificación.

Chi  
Square

Odds  
Ratio

Mutual  
Information

S. Short  
Text W.

Gini  
Index

Information  
Gain



# Ponderación de características

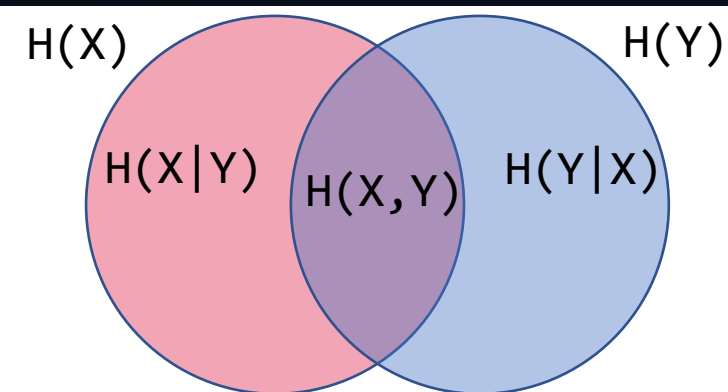
## Supervisado

### Information Gain

- Mide la información obtenida a partir de conocer la información de la presencia o ausencia de un término en un texto.
  - Requiere el cálculo de propiedades condicionales y entropía.
  - Su valor es afectado por incremento de la dependencia entre los términos y la entropía.
  - Términos con baja entropía reciben valores bajos, aún cuando se encuentren relacionados con la clase.
- 
- Gain Ratio (GR) es la version normalizada de IG.
    - Al normalizar por la entropía disminuye el bias por los términos frecuentes.

$$IG(t) = -\sum_{i=1}^m P(c_i) \log P(c_i) + P(t) \sum_{i=1}^m P(c_i|t) \log P(c_i|t) + P(\bar{t}) \sum_{i=1}^m P(c_i|\bar{t}) \log P(c_i|\bar{t})$$

$$GR(t) = \frac{IG(t)}{-\sum_{g \in \{t, \bar{t}\}} P(g) \log P(g)}$$



La entropía puede ser considerada como una medida de la incertidumbre y de la información necesaria para, en cualquier proceso, poder acotar, reducir o eliminar la incertidumbre.



# Ponderación de características

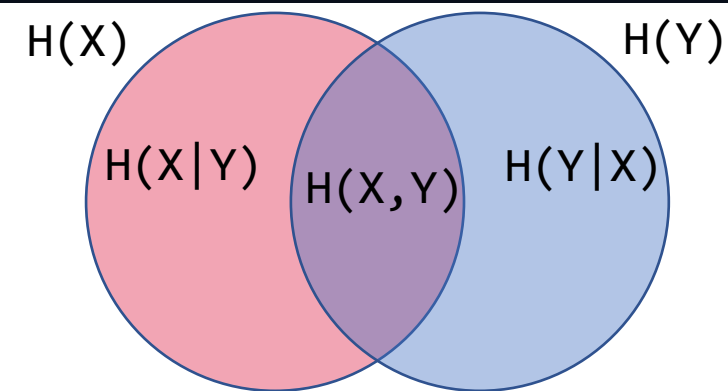
## Supervisado

### Information Gain

- Mide la información obtenida a partir de conocer la información de la presencia o ausencia de un término en un texto.
  - Requiere el cálculo de propiedades condicionales y entropía.
  - Su valor es afectado por incremento de la dependencia entre los términos y la entropía.
  - Términos con baja entropía reciben valores bajos, aún cuando se encuentren relacionados con la clase.
- 
- Gain Ratio (GR) es la version normalizada de IG.
    - Al normalizar por la entropía disminuye el bias por los términos frecuentes.

$$IG(t) = \frac{-\sum_{i=1}^m P(c_i) \log P(c_i)}{+P(t) \sum_{i=1}^m P(c_i|t) \log P(c_i|t) + P(\bar{t}) \sum_{i=1}^m P(c_i|\bar{t}) \log P(c_i|\bar{t})}$$

$$GR(t) = \frac{IG(t)}{-\sum_{g \in \{t, \bar{t}\}} P(g) \log P(g)}$$



La entropía puede ser considerada como una medida de la incertidumbre y de la información necesaria para, en cualquier proceso, poder acotar, reducir o eliminar la incertidumbre.

### Mutual Information

- En algunas fuentes la definen de la misma que Information Gain.
- La diferencia es la simetría.
  - Mientras que Mutual Information solo considera los términos positivos, Information Gain considera todos.

$$MI(t, c) = \log \frac{P(t, c)}{P(t)P(c)}$$





# Ponderación de características

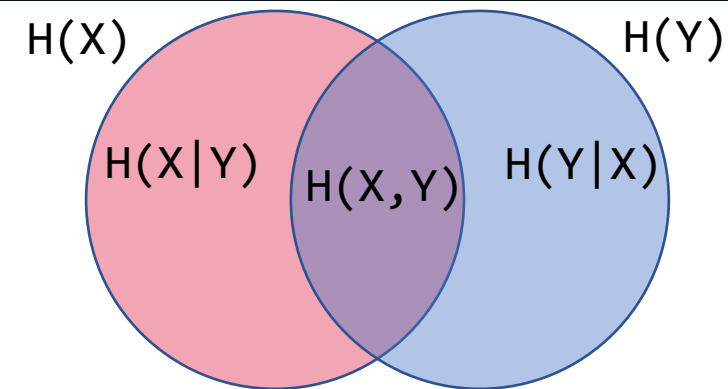
## Supervisado

### Information Gain

- Mide la información obtenida a partir de conocer la información de la presencia o ausencia de un término en un texto.
  - Requiere el cálculo de propiedades condicionales y entropía.
  - Su valor es afectado por incremento de la dependencia entre los términos y la entropía.
  - Términos con baja entropía reciben valores bajos, aún cuando se encuentren relacionados con la clase.
- 
- Gain Ratio (GR) es la version normalizada de IG.
    - Al normalizar por la entropía disminuye el bias por los términos frecuentes.

$$IG(t) = \frac{-\sum_{i=1}^m P(c_i) \log P(c_i)}{+P(t) \sum_{i=1}^m P(c_i|t) \log P(c_i|t) + P(\bar{t}) \sum_{i=1}^m P(c_i|\bar{t}) \log P(c_i|\bar{t})}$$

$$GR(t) = \frac{IG(t)}{-\sum_{g \in \{t, \bar{t}\}} P(g) \log P(g)}$$



La entropía puede ser considerada como una medida de la incertidumbre y de la información necesaria para, en cualquier proceso, poder acotar, reducir o eliminar la incertidumbre.

### Mutual Information

- En algunas fuentes la definen de la misma que Information Gain.
- La diferencia es la simetría.
  - Mientras que Mutual Information solo considera los términos positivos, Information Gain considera todos.

$$MI(t, c) = \log \frac{A \times N}{(A + C) \times (A + B)}$$





### Chi Square

- Mide la dependencia de un término respecto a una clase.
- Normalizada.
- Puede no ser adecuada en textos con vocabularios ralos o con pocos textos por clase.

$$\chi^2(t, c) = \frac{N \times (AD - CB)^2}{(A + C) \times (B + D) \times (A + B) \times (C + D)}$$

- $A \rightarrow$  cantidad de veces que el término  $t$  aparece en un texto que es de clase  $c$ .
- $B \rightarrow$  cantidad de veces que el término  $t$  aparece en un texto que no es de clase  $c$ .
- $C \rightarrow$  cantidad de textos de clase  $c$  en los que no aparece  $t$ .
- $D \rightarrow$  cantidad de textos donde no aparece el término  $t$  ni son de clase  $c$ .
- $N \rightarrow$  cantidad total de textos.

### Chi Square

- Mide la dependencia de un término respecto a una clase.
- Normalizada.
- Puede no ser adecuada en textos con vocabularios ralos o con pocos textos por clase.

$$\chi^2(t, c) = \frac{N \times (AD - CB)^2}{(A + C) \times (B + D) \times (A + B) \times (C + D)}$$

- $A \rightarrow$  cantidad de veces que el término  $t$  aparece en un texto que es de clase  $c$ .
- $B \rightarrow$  cantidad de veces que el término  $t$  aparece en un texto que no es de clase  $c$ .
- $C \rightarrow$  cantidad de textos de clase  $c$  en los que no aparece  $t$ .
- $D \rightarrow$  cantidad de textos donde no aparece el término  $t$  ni son de clase  $c$ .
- $N \rightarrow$  cantidad total de textos.

### Odds Ratio

- Refleja la probabilidad de los términos apareciendo en la clase de interés, normalizada por la probabilidad en el resto de las clases.
- Asume que los términos que aparecen frecuentemente en una clase no son relevantes si también aparecen en la misma frecuencia en las otras clases.
- Se suele introducir un factor de corrección para no dividir por cero.
- El logaritmo es opcional.

$$OR(t, c) = \log \frac{A \times (1 - B)}{B \times (1 - A)}$$

### Gini Index

- Mide dispersión o la “impuridad” de los términos.
- Normalizada.
- A mayor valor, mayor cantidad de clases en las que el término se encuentra equitativamente distribuido.
- A menor valor, el término presenta más diferencias en las apariciones en las distintas clases.
- Hay varias formulaciones.

$$GiniIndex(t, c) = \frac{P(c|t)^2}{|\log_2 P(t|c)^2|}$$

$$GiniIndex(t) = \frac{\sum_{c \in C} P(c|t)^2}{\sum_{c \in C} |\log_2 P(t|c)^2|}$$

Versión extendida a todas las clases. En este caso, un término recibe siempre el mismo valor, independientemente de la clase a la que pertenezca el texto

# Ponderación de características

## Supervisado

### Gini Index

- Mide dispersión o la “impuridad” de los términos.
- Normalizada.
- A mayor valor, mayor cantidad de clases en las que el término se encuentra equitativamente distribuido.
- A menor valor, el término presenta más diferencias en las apariciones en las distintas clases.
- Hay varias formulaciones.

$$GiniIndex(t, c) = \frac{P(c|t)^2}{|\log_2 P(t|c)^2|}$$

$$GiniIndex(t) = \frac{\sum_{c \in C} P(c|t)^2}{\sum_{c \in C} |\log_2 P(t|c)^2|}$$

Versión extendida a todas las clases. En este caso, un término recibe siempre el mismo valor, independientemente de la clase a la que pertenezca el texto

### S. Short Text W.

- Propone una combinación de la relevancia de un término con un factor de su distribución en los diferentes textos aplicado a textos cortos.
- Estima la distribución de los términos en cada una de las clases y en el conjunto total de textos.
- Valores altos para aquellos términos que aparecen frecuentemente en una única clase y poco en otras clases.

$$SSTW(t, d) = \frac{TF(t, d) + 1}{\sum_{i=1}^{|t|} TF(t, d) + |T|} \times \log \left( 1 + \frac{A}{B + C + 1} \right)$$

cantidad de términos en el texto d



# Extra: Modelos basados en grafos

## Graph-of-Word

- Los grafos han sido utilizados en IR (information retrieval) para encontrar relaciones y proponer esquemas de pesos significativos.
  - Ej. PageRank.
- Desafían la independencia de los tokens y los esquemas de ponderado por frecuencias que toman en cuenta la independencia de palabras, orden y distancia.
- Usos:
  - Desambiguación.
  - Entailment.
  - Summarization.
  - Relation extraction.
  - Narrative fluency

[Graph-of-word and TW-IDF: New Approach to Ad Hoc IR](#)



# Extra: Modelos basados en grafos

## Graph-of-Word

- Los grafos han sido utilizados en IR (information retrieval) para encontrar relaciones y proponer esquemas de pesos significativos.
  - Ej. PageRank.
- Desafían la independencia de los tokens y los esquemas de ponderado por frecuencias que toman en cuenta la independencia de palabras, orden y distancia.
  - Desambiguación.
  - Entailment.
  - Summarization.
  - Relation extraction.

information retrieval is the activity of  
obtaining information resources relevant  
to an information need from a collection  
of information resources

[Graph-of-word and TW-IDF: New Approach to Ad Hoc IR](#)

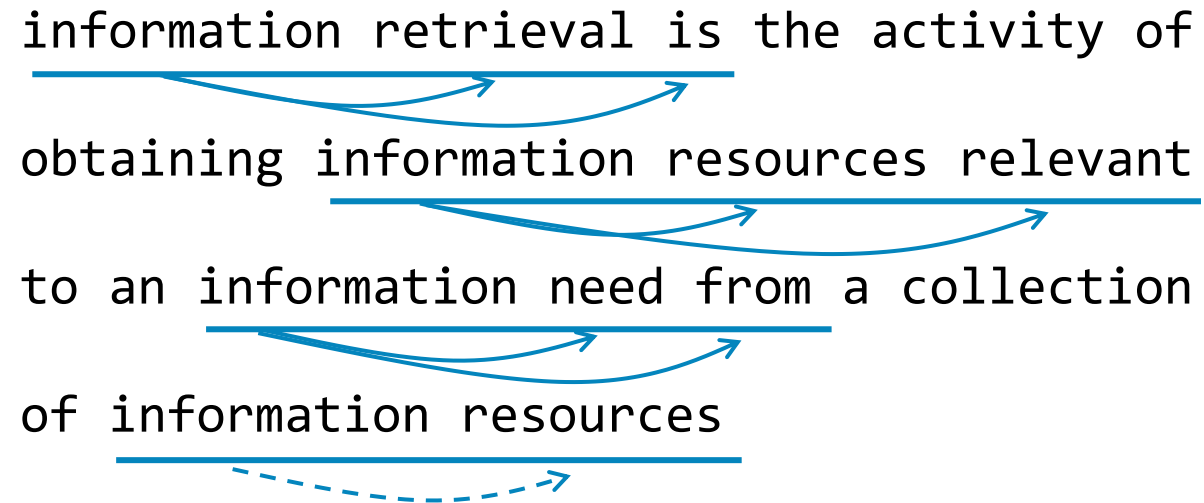


# Extra: Modelos basados en grafos

## Graph-of-Word

- Los grafos han sido utilizados en IR (information retrieval) para encontrar relaciones y proponer esquemas de pesos significativos.
  - Ej. PageRank.
- Desafían la independencia de los tokens y los esquemas de ponderado por frecuencias que toman en cuenta la independencia de palabras, orden y distancia.
  - Desambiguación.
  - Entailment.
  - Summarization.
  - Relation extraction.

information retrieval is the activity of  
obtaining information resources relevant  
to an information need from a collection  
of information resources



[Graph-of-word and TW-IDF: New Approach to Ad Hoc IR](#)



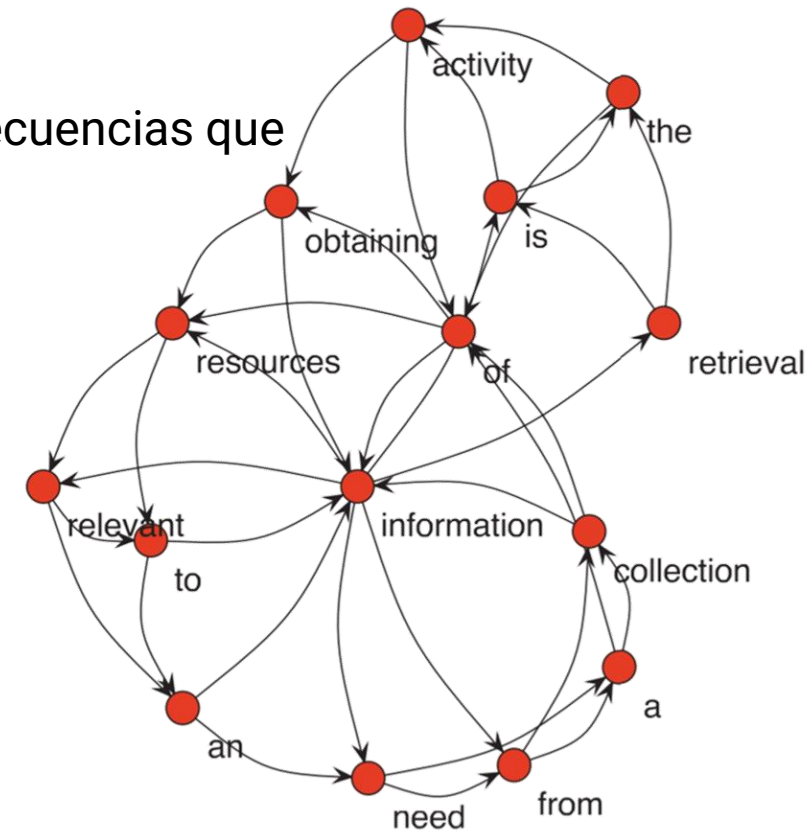


# Extra: Modelos basados en grafos

## Graph-of-Word

- Los grafos han sido utilizados en IR (information retrieval) para encontrar relaciones y proponer esquemas de pesos significativos.
  - Ej. PageRank.
- Desafían la independencia de los tokens y los esquemas de ponderado por frecuencias que toman en cuenta la independencia de palabras, orden y distancia.
  - Desambiguación.
  - Entailment.
  - Summarization.
  - Relation extraction.

information retrieval is the activity of  
obtaining information resources relevant  
to an information need from a collection  
of information resources



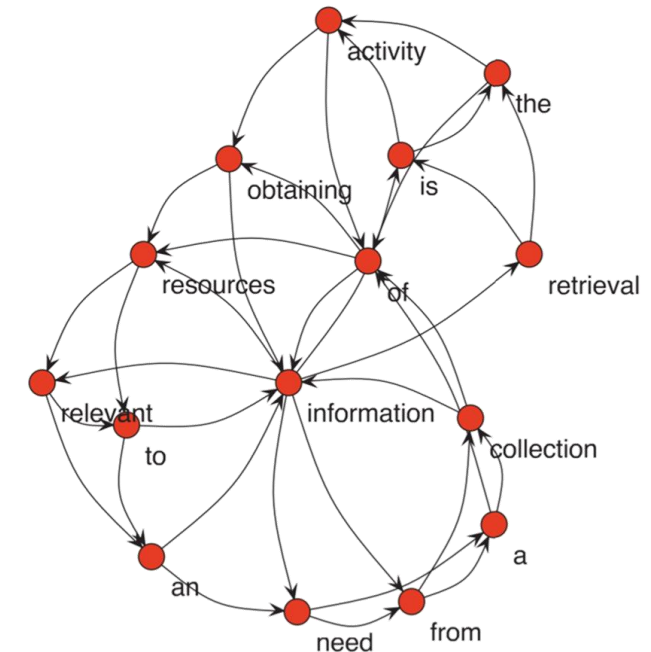
[Graph-of-word and TW-IDF: New Approach to Ad Hoc IR](#)



# Extra: Modelos basados en grafos

## Graph-of-Word

- Tenemos el  $G = (V, E)$ .
- Los **nodos** se pueden corresponder con:
  - Párrafos.
  - Oraciones.
  - Frases.
  - Palabras.
  - Sílabas.
  - Chars.
- Los **arcos** pueden capturar distintos tipos de relaciones:
  - Co-ocurrencias de palabras en una determinada ventana.
    - La dirección del arco indica el orden de las palabras.
  - Relaciones sintácticas.
    - Análisis sintáctico.
  - Relaciones semánticas.
    - Por ejemplo, sinónimos, antónimos, ...



[Graph-of-word and TW-IDF: New Approach to Ad Hoc IR](#)



# Extra: Modelos basados en grafos

## Graph-of-Word

- **Grafo dirigido vs. no dirigido**
  - En grafos no dirigidos se representa la relación de forma “bilateral”.
  - Grafos dirigidos permiten preservar el flujo del texto.
- **Grafo ponderado vs. sin ponderar**
  - Todas las relaciones valen lo mismo.
  - El peso de la relación varia de acuerdo con la cantidad de veces que la relación ocurrió.
  - Considerar que muchas de las técnicas standard de grafos suelen estar diseñadas para grafos no pesados.
- **Tamaño de la ventana**
  - Se pueden considerar ventanas de diferentes tamaños.
  - En algunos estudios recomiendan entre 6 y 30.
  - A mayor ventana, mayor densidad en el grafo.
  - Modifica el impacto de la dependencia de término.
  - Tener en cuenta que el tamaño de la Ventana afecta de modo lineal la complejidad temporal.

[Graph-of-word and TW-IDF: New Approach to Ad Hoc IR](#)



# Extra: Modelos basados en grafos

## Graph-of-Word

- Se pueden utilizar otras alternativas de ponderado.
- Por ejemplo, reemplazar el TF por un Term Weight derivado de la centralidad de los nodos.

Degree-  
based

Eigenvector

Betweenness

Closeness



# Extra: Modelos basados en grafos

## Graph-of-Word

- Se pueden utilizar otras alternativas de ponderado.
- Por ejemplo, reemplazar el TF por un Term Weight derivado de la centralidad de los nodos.

Degree-  
based

Eigenvector

Betweenness

Closeness

- **Degree-based.**
  - Categoría más simple.
  - Definida en función de la cantidad y peso de las conexiones entre los nodos y sus vecinos.
- **Eigenvector-centralities.**
  - No solo consideran la cantidad de vecinos, sino también la importancia de cada vecino.
  - Ej. PageRank



# Extra: Modelos basados en grafos

## Graph-of-Word

- Se pueden utilizar otras alternativas de ponderado.
- Por ejemplo, reemplazar el TF por un Term Weight derivado de la centralidad de los nodos.

Degree-  
based

Eigenvector

Betweenness

Closeness

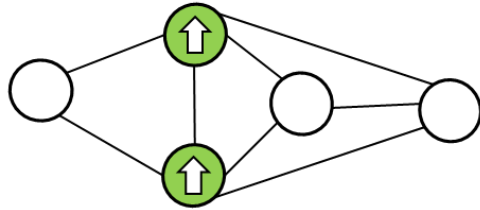
- **Closeness.**
  - Basado en estructura general de la red.
  - Longitudes de los caminos más cortos entre un nodo y el resto en la red.
- **Betweenness centrality.**
  - Basado en estructura general de la red.
  - Basado en cuán frecuente es que un nodo sea parte de los caminos más cortos entre todos los otros pares.





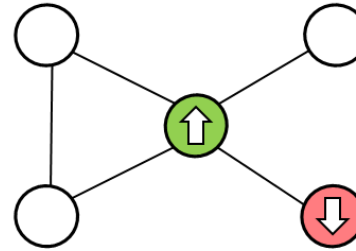
# Extra: Modelos basados en grafos

## Graph-of-Word



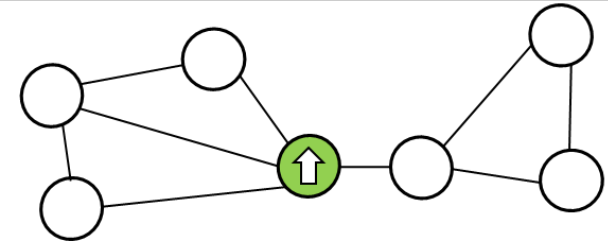
### Page Rank

Mide las conexiones de los nodos y las conexiones de sus conexiones teniendo en cuenta la dirección y peso.



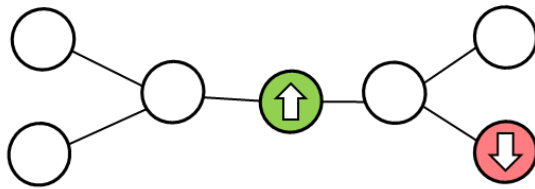
### Clustering Coefficient

Mide cuan conectados están los nodos entre sí.



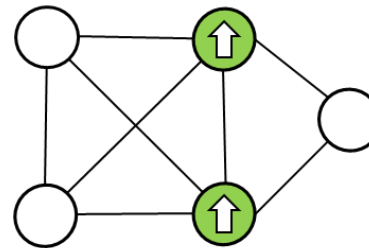
### Eigenvector Centrality

Mide la influencia de un nodo basado en la cantidad de link que tiene a otros nodos en el grafo.



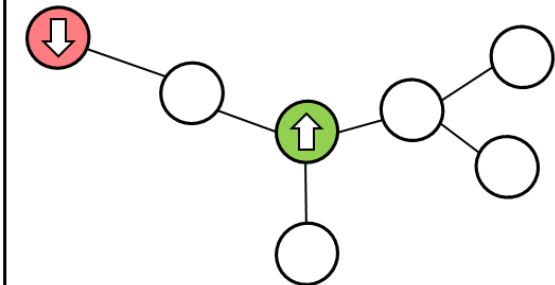
### Betweenness Centrality

Mide la cantidad de veces que un nodo se encuentra en el camino más corto de dos nodos.



### Katz Centrality

Mide la cantidad de nodos adyacentes y todos los otros nodos en el grafo a los que se puede alcanzar mediante los nodos adyacente.



### Closeness Centrality

Mide la cercanía de un nodo con todo el resto.



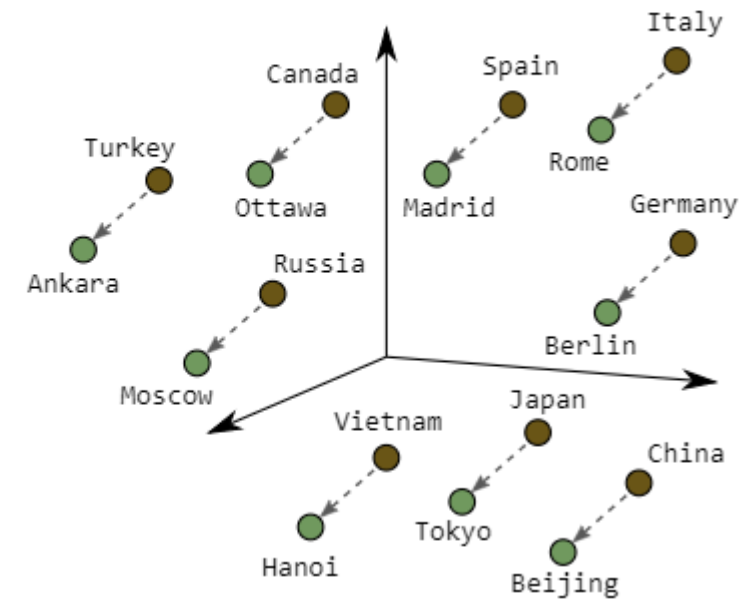
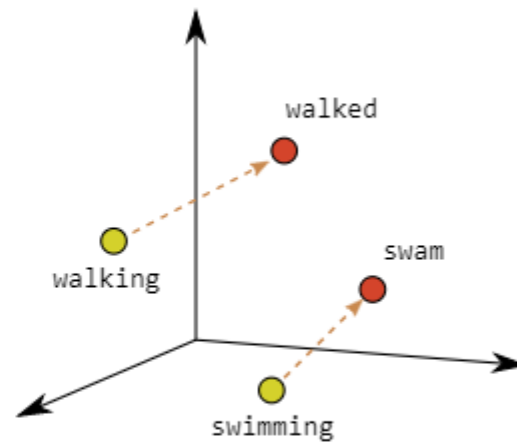
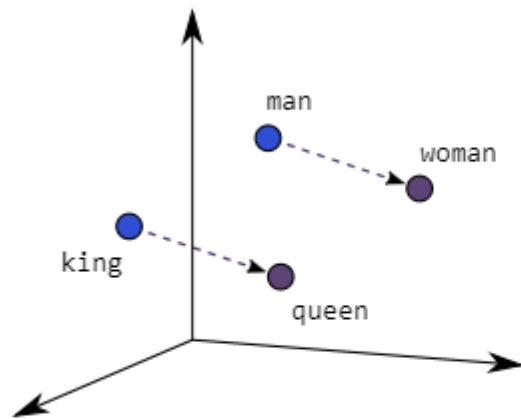


# Modelos basados en Deep Learning

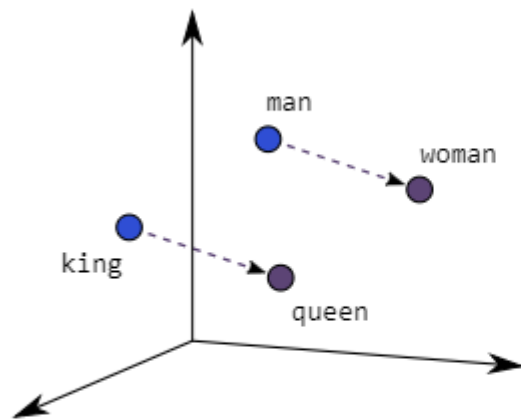
- Si bien las técnicas anteriores son efectivas para la extracción de características, pierden información como:
  - Semántica.
  - Estructura.
  - Secuencia, contexto.
- Las representaciones tradicionales requieren gran cantidad de datos.
  - Sin datos los modelos pueden no capturar lo esencial/relevante.
  - Curse of dimensionality.
  - Overfitting.
- Estas representaciones se basan en inferencia de semántica.
  - Palabras que ocurren y son utilizadas en el mismo contexto son semánticamente y contextualmente similares.
  - “una palabra se caracteriza por la compañía que tiene”.



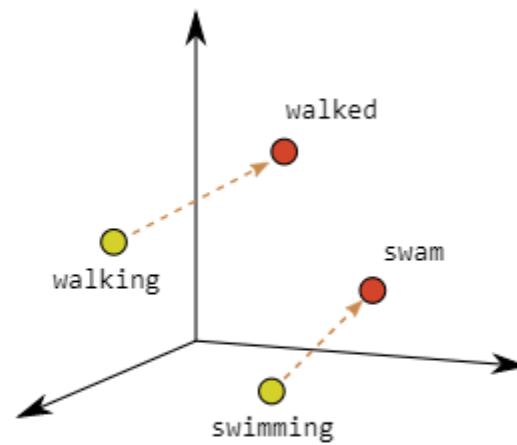
# Modelos basados en Deep Learning



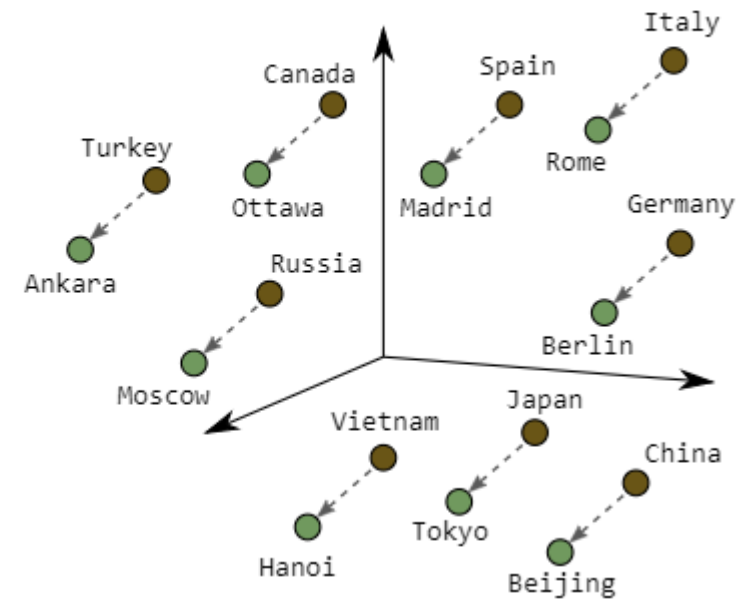
# Modelos basados en Deep Learning



Género  
(útil para tareas  
de eliminar bias)



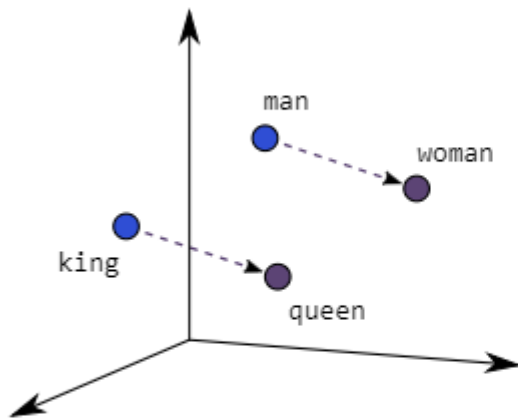
Tiempos  
verbales



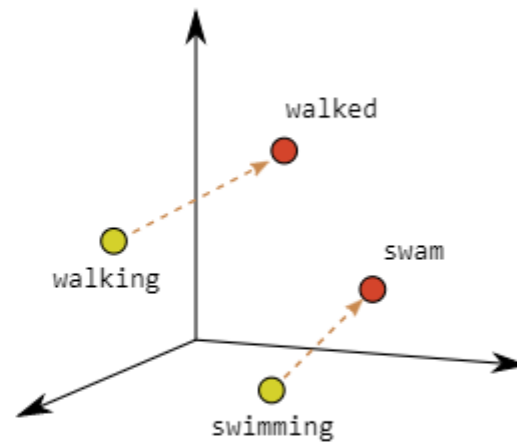
País - Capital



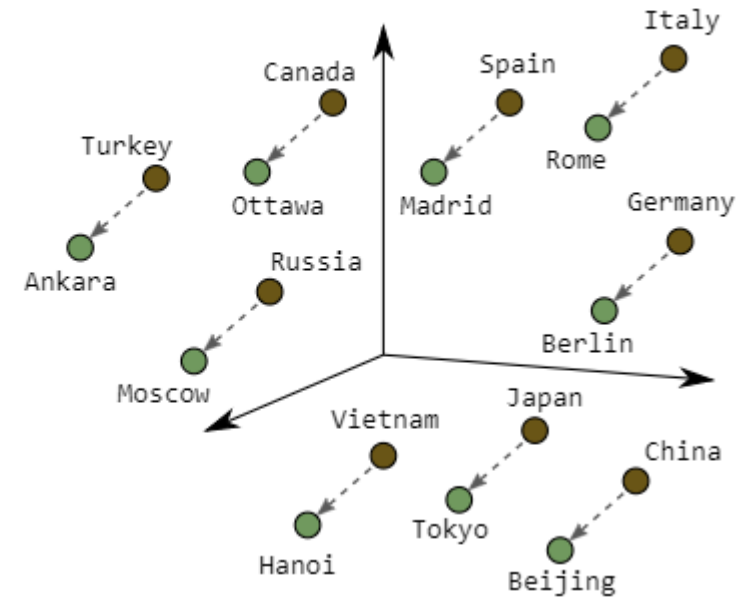
# Modelos basados en Deep Learning



Género  
(útil para tareas  
de eliminar bias)



Tiempos  
verbales



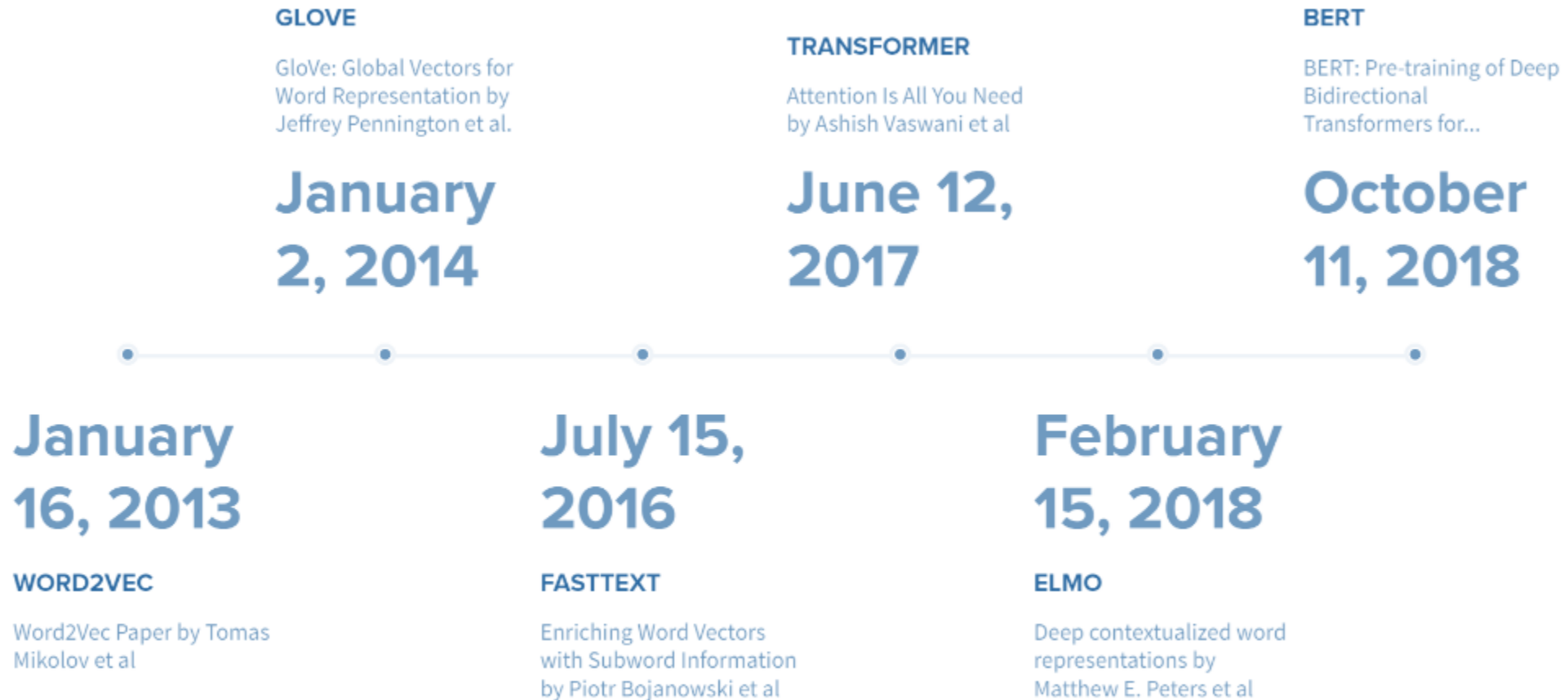
País - Capital

Analogías: X es a Y, como X' es a Y'



# Modelos basados en Deep Learning

## Evolución



# Modelos basados en Deep Learning

- Word embeddings es el nombre por defecto que se le da a estas representaciones, aunque también se las puede encontrar como:
  - Distributional semantic model.
  - Distributed representation.
  - Semantic vector space.
  - Word space.

Nota. No vamos a ver detalles de la implementación. Eso en el próximo curso.



# Modelos basados en Deep Learning

- Word embeddings es el nombre por defecto que se le da a estas representaciones, aunque también se las puede encontrar como:
  - Distributional semantic model.
  - Distributed representation.
  - Semantic vector space.
  - Word space.
- De qué hablamos cuando hablamos de embeddings?
  - Representaciones densas de elementos en la forma de vectores en un espacio de dimensionalidad reducida.

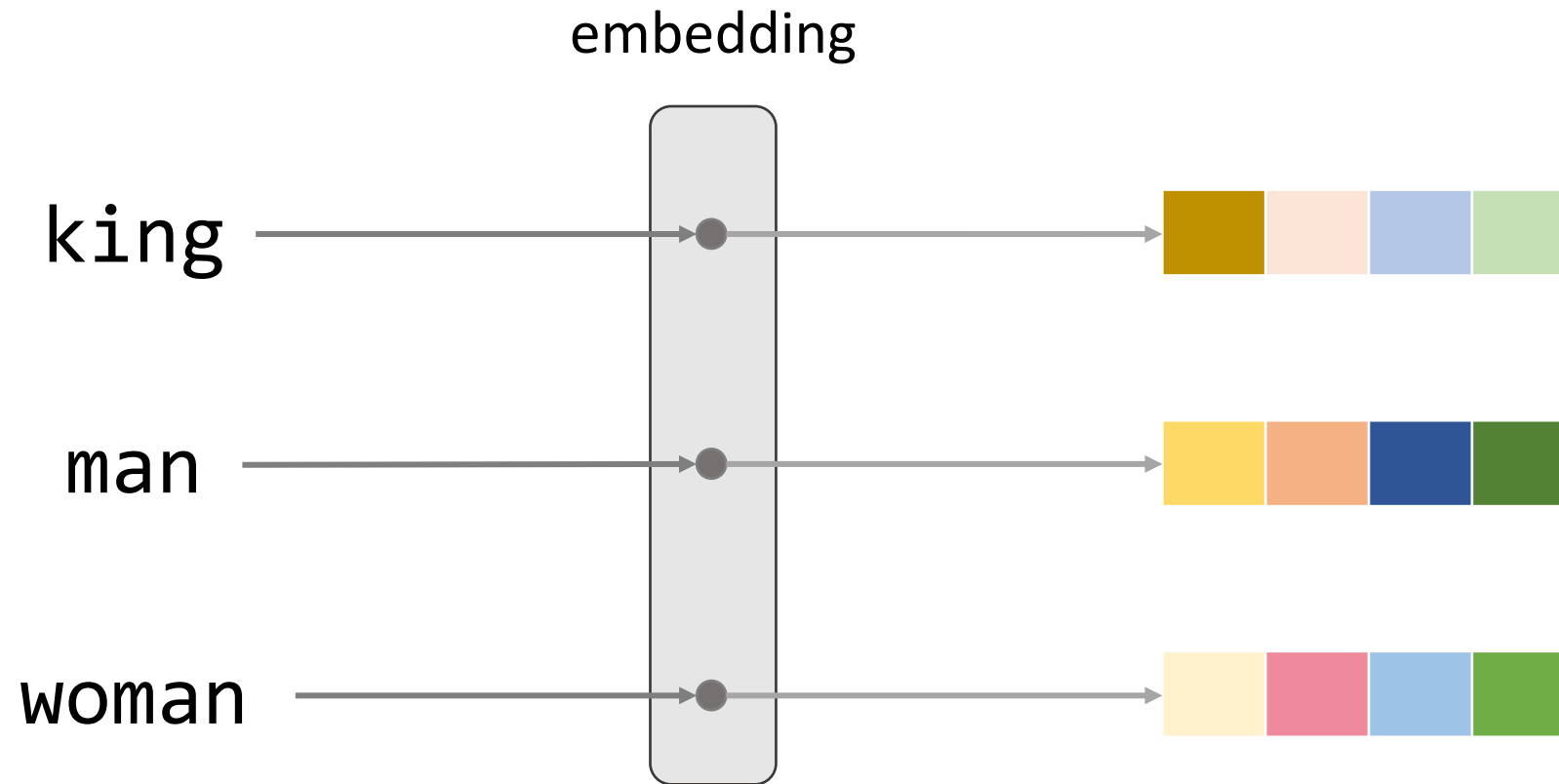
Nota. No vamos a ver detalles de la implementación. Eso en el próximo curso.





# Modelos basados en Deep Learning

Arranquemos por un ejemplo...



# Modelos basados en Deep Learning

Arranquemos por un ejemplo...

```
[ 0.50451 , 0.68607 , -0.59517 , -0.022801, 0.60046 , -0.13498 , -0.08813 , 0.47377 , -0.61798 ,  
-0.31012 , -0.076666, 1.493 , -0.034189, -0.98173 , 0.68229 , 0.81722 , -0.51874 , -0.31503 ,  
-0.55809 , 0.66421 , 0.1961 , -0.13495 , -0.11476 , -0.30344 , 0.41177 , -2.223 , -1.0756 , -  
1.0783 , -0.34354 , 0.33505 , 1.9927 , -0.04234 , -0.64319 , 0.71125 , 0.49159 , 0.16754 ,  
0.34344 , -0.25663 , -0.8523 , 0.1661 , 0.40102 , 1.1685 , -1.0137 , -0.21585 , -0.15155 ,  
0.78321 , -0.91241 , -1.6106 , -0.64426 , -0.51042 ]
```

max  
0  
min

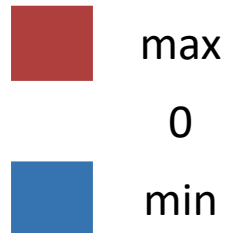


# Modelos basados en Deep Learning

Arranquemos por un ejemplo...

```
[ 0.50451 , 0.68607 , -0.59517 , -0.022801, 0.60046 , -0.13498 , -0.08813 , 0.47377 , -0.61798 ,  
-0.31012 , -0.076666, 1.493 , -0.034189, -0.98173 , 0.68229 , 0.81722 , -0.51874 , -0.31503 ,  
-0.55809 , 0.66421 , 0.1961 , -0.13495 , -0.11476 , -0.30344 , 0.41177 , -2.223 , -1.0756 , -  
1.0783 , -0.34354 , 0.33505 , 1.9927 , -0.04234 , -0.64319 , 0.71125 , 0.49159 , 0.16754 ,  
0.34344 , -0.25663 , -0.8523 , 0.1661 , 0.40102 , 1.1685 , -1.0137 , -0.21585 , -0.15155 ,  
0.78321 , -0.91241 , -1.6106 , -0.64426 , -0.51042 ]
```

king



# Modelos basados en Deep Learning

Arranquemos por un ejemplo...

```
[ 0.50451 , 0.68607 , -0.59517 , -0.022801, 0.60046 , -0.13498 , -0.08813 , 0.47377 , -0.61798 ,  
-0.31012 , -0.076666, 1.493 , -0.034189, -0.98173 , 0.68229 , 0.81722 , -0.51874 , -0.31503 ,  
-0.55809 , 0.66421 , 0.1961 , -0.13495 , -0.11476 , -0.30344 , 0.41177 , -2.223 , -1.0756 , -  
1.0783 , -0.34354 , 0.33505 , 1.9927 , -0.04234 , -0.64319 , 0.71125 , 0.49159 , 0.16754 ,  
0.34344 , -0.25663 , -0.8523 , 0.1661 , 0.40102 , 1.1685 , -1.0137 , -0.21585 , -0.15155 ,  
0.78321 , -0.91241 , -1.6106 , -0.64426 , -0.51042 ]
```

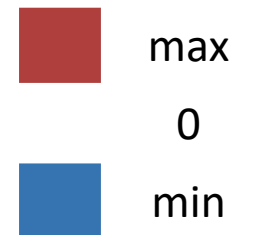
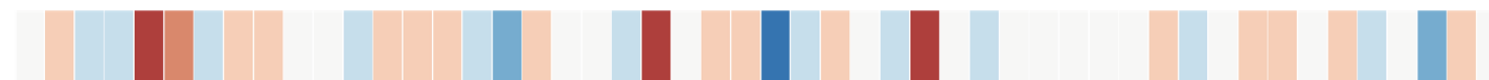
king



man




woman




# Modelos basados en Deep Learning

Arranquemos por un ejemplo...

king 

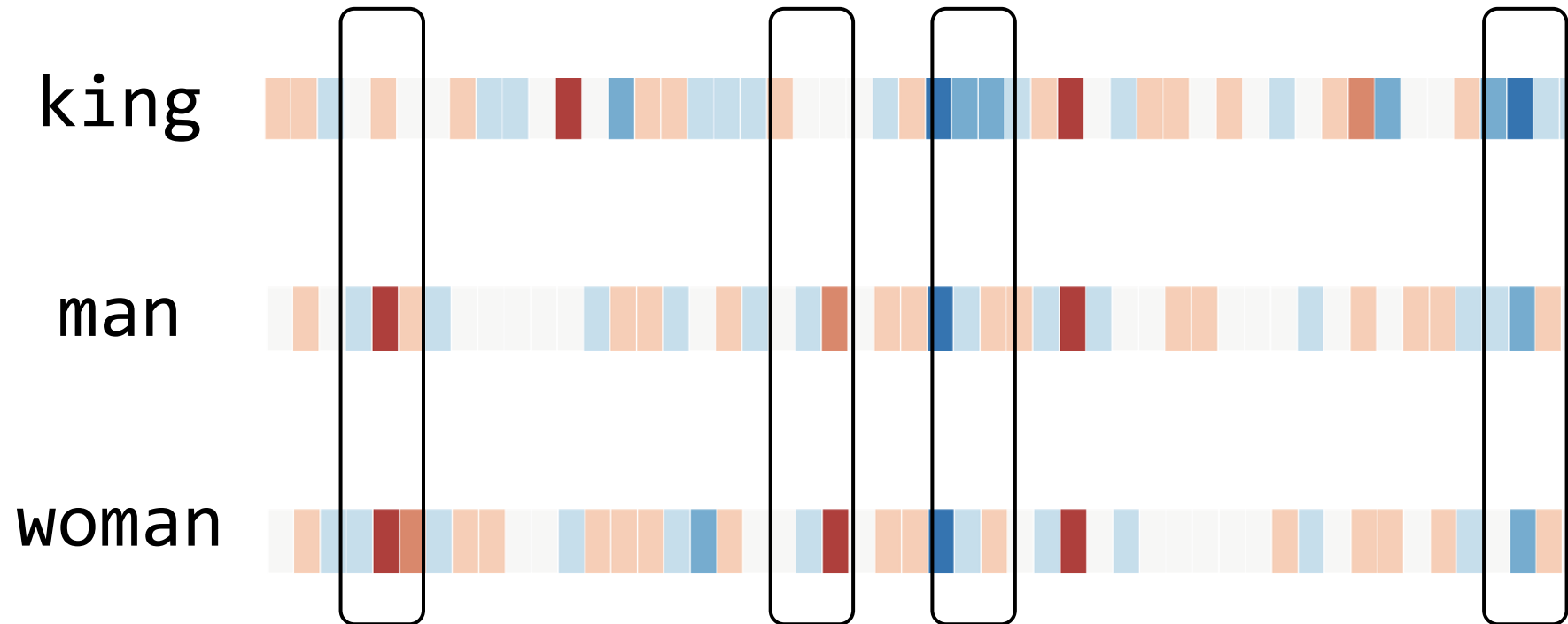
man 

woman 



# Modelos basados en Deep Learning

Arranquemos por un ejemplo...



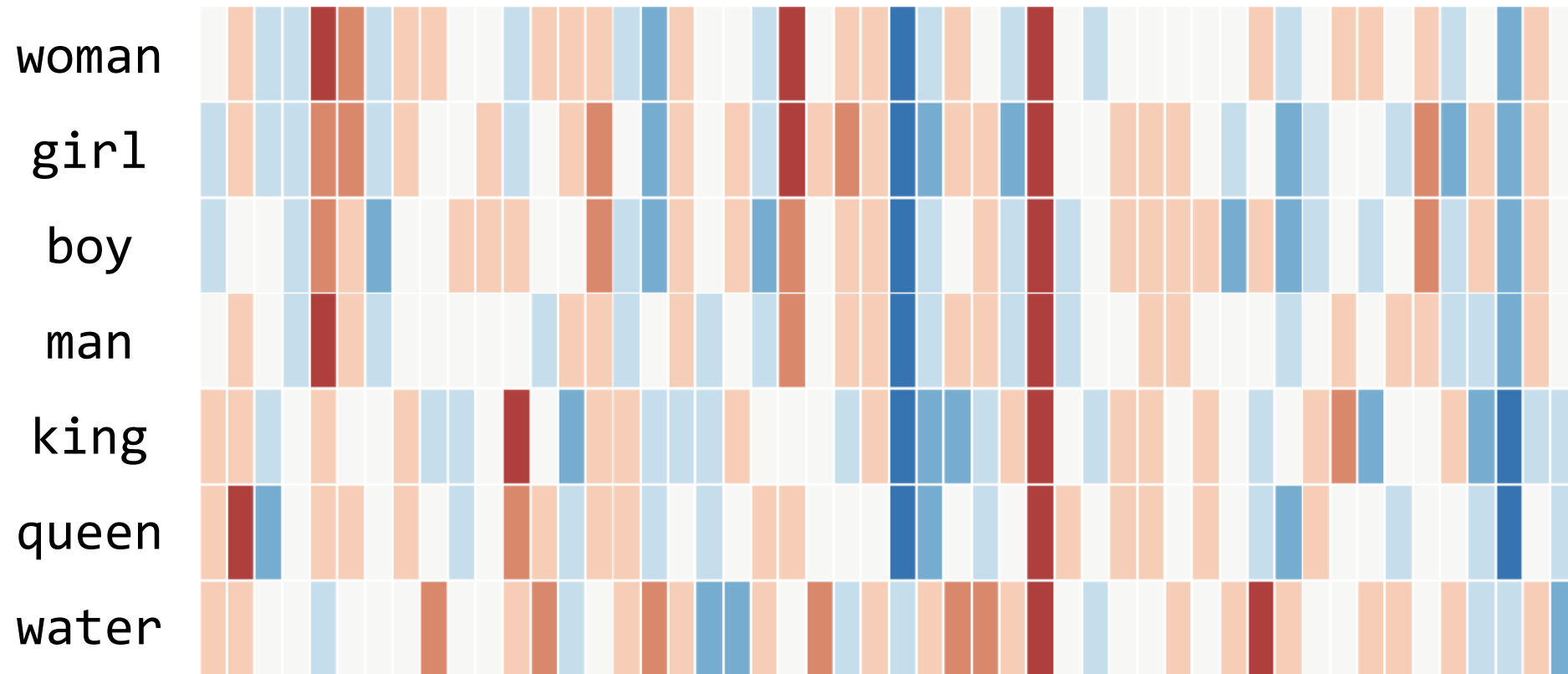
hay más semejanzas entre man y women que con king

Estas representaciones capturan el significado e incluso asociaciones entre las palabras



# Modelos basados en Deep Learning

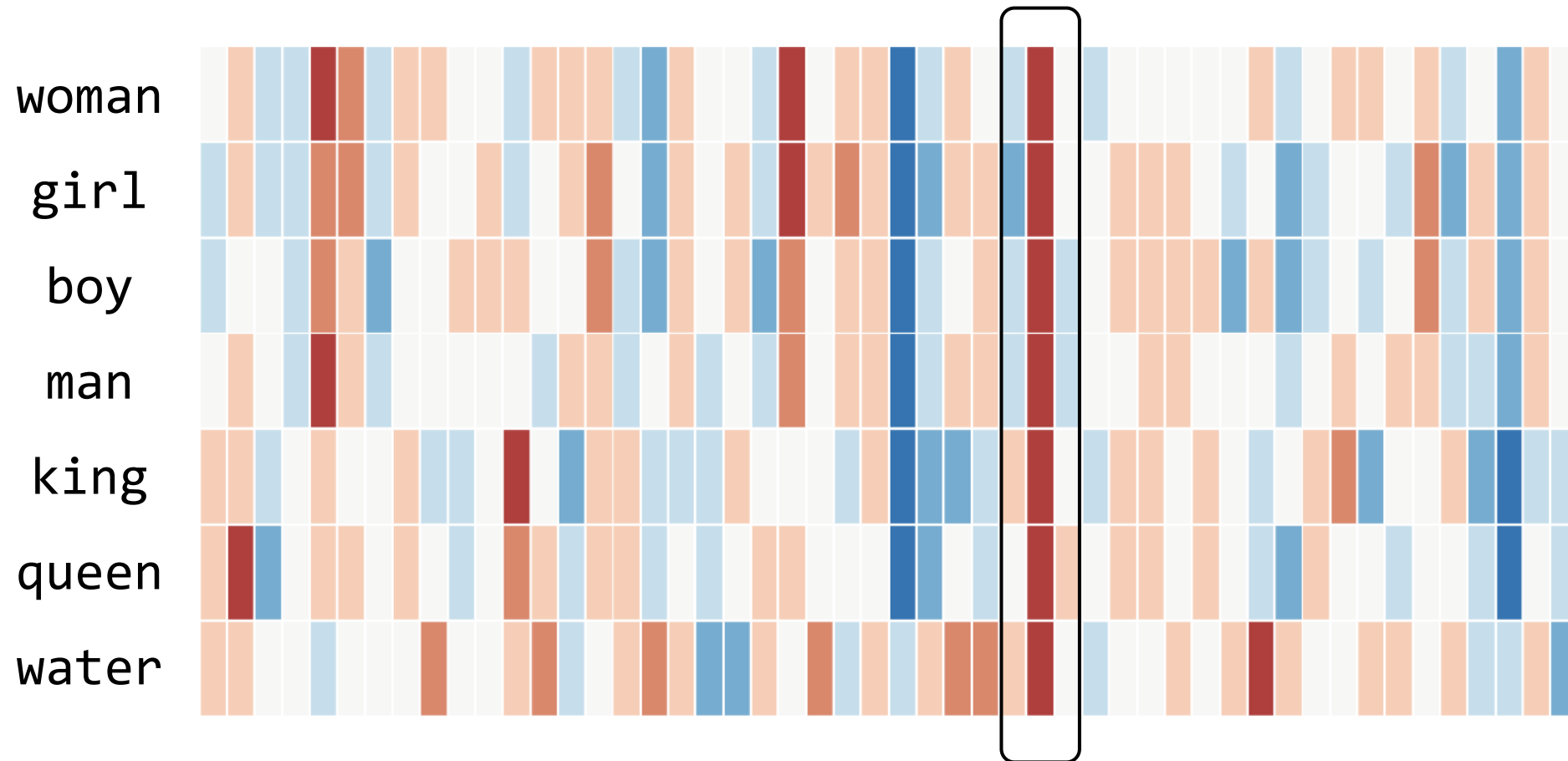
Arranquemos por un ejemplo...





# Modelos basados en Deep Learning

Arranquemos por un ejemplo...

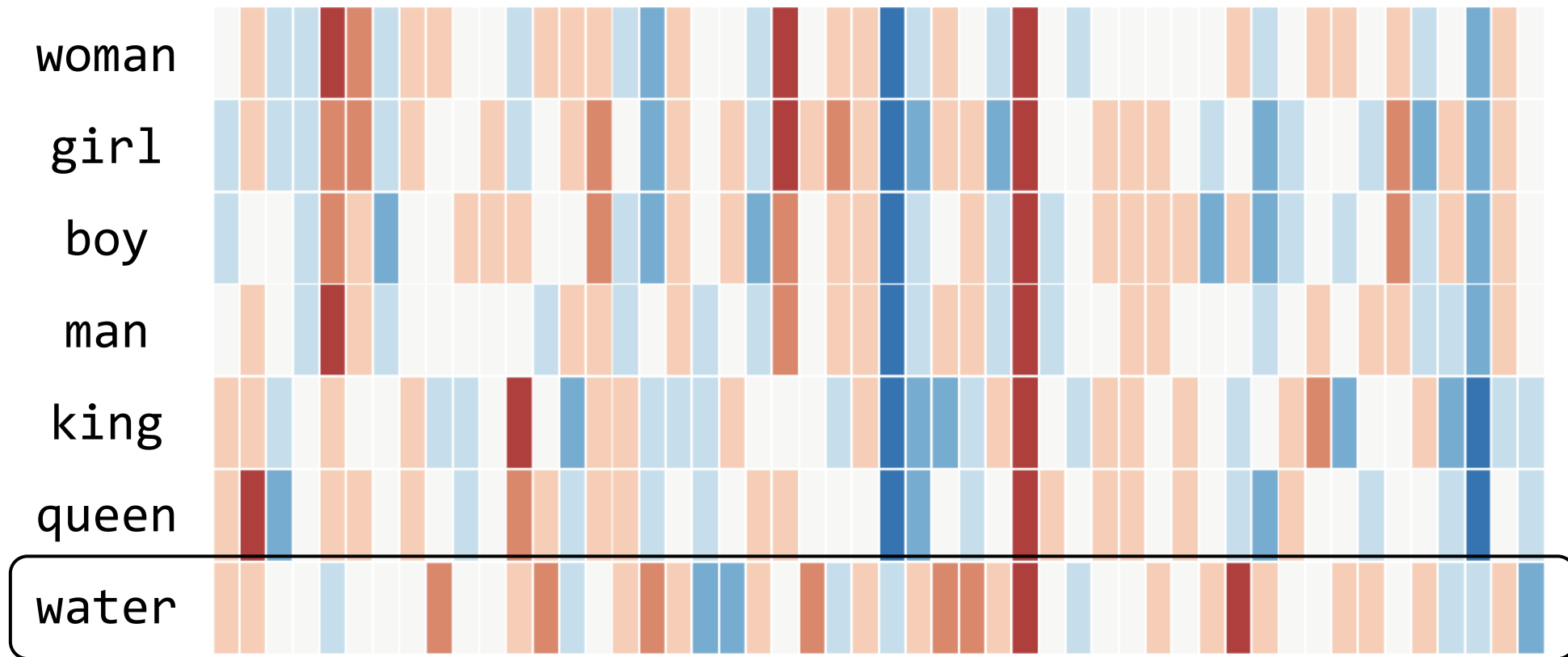


No sabemos qué significa la dimension, pero si que todas las palabras son similares



# Modelos basados en Deep Learning

Arranquemos por un ejemplo...

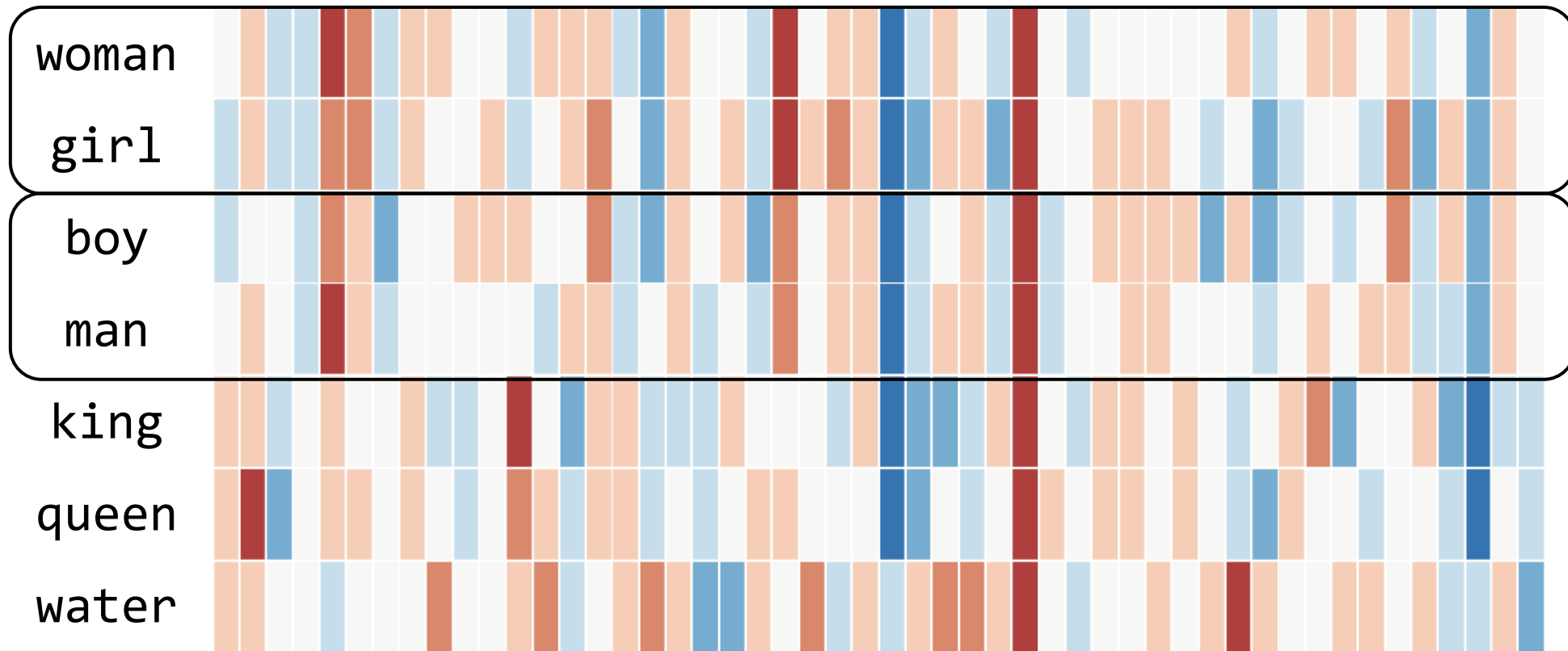


Todas las palabras menos esta representan personas.  
Hay una columna que es común a todas las otras, menos esta palabra. Distingue categorías.



# Modelos basados en Deep Learning

Arranquemos por un ejemplo...

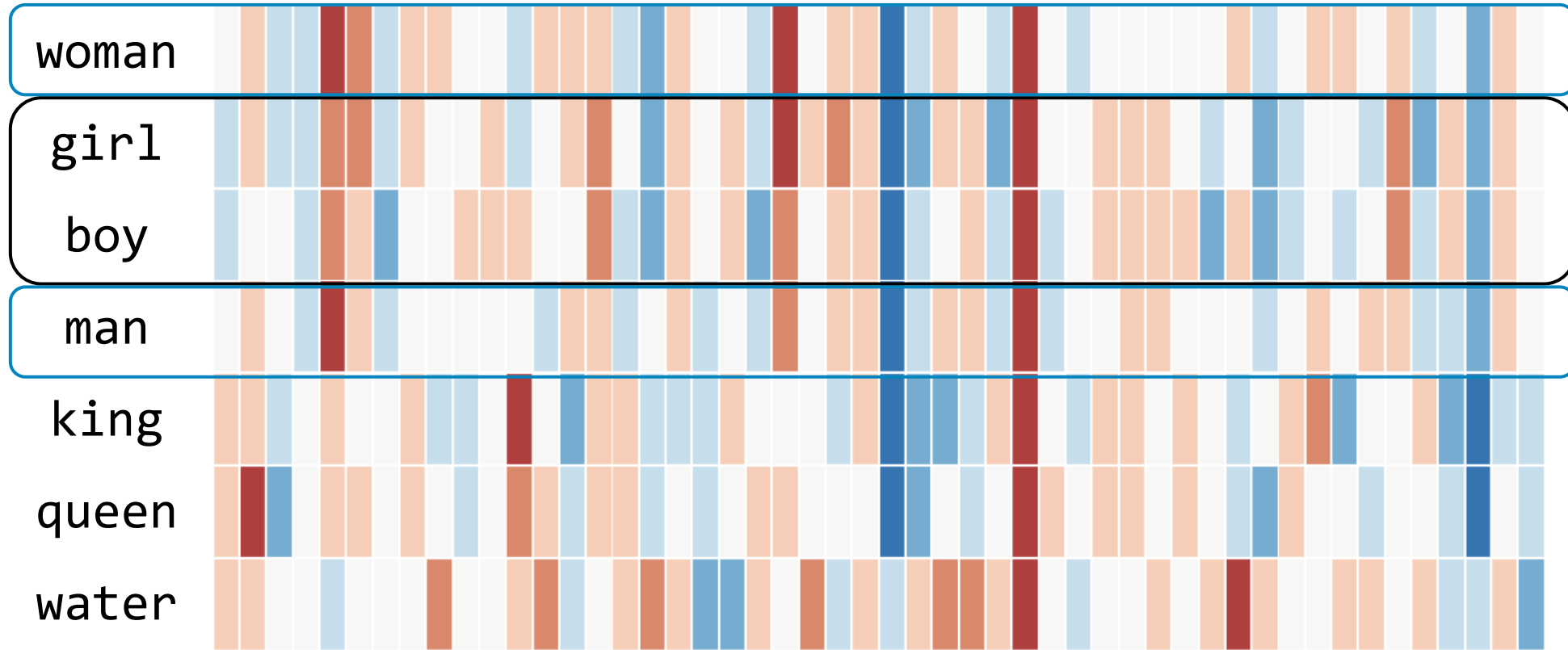


woman y girl tienen muchas semejanzas, lo mismo que entre boy y man.



# Modelos basados en Deep Learning

Arranquemos por un ejemplo...

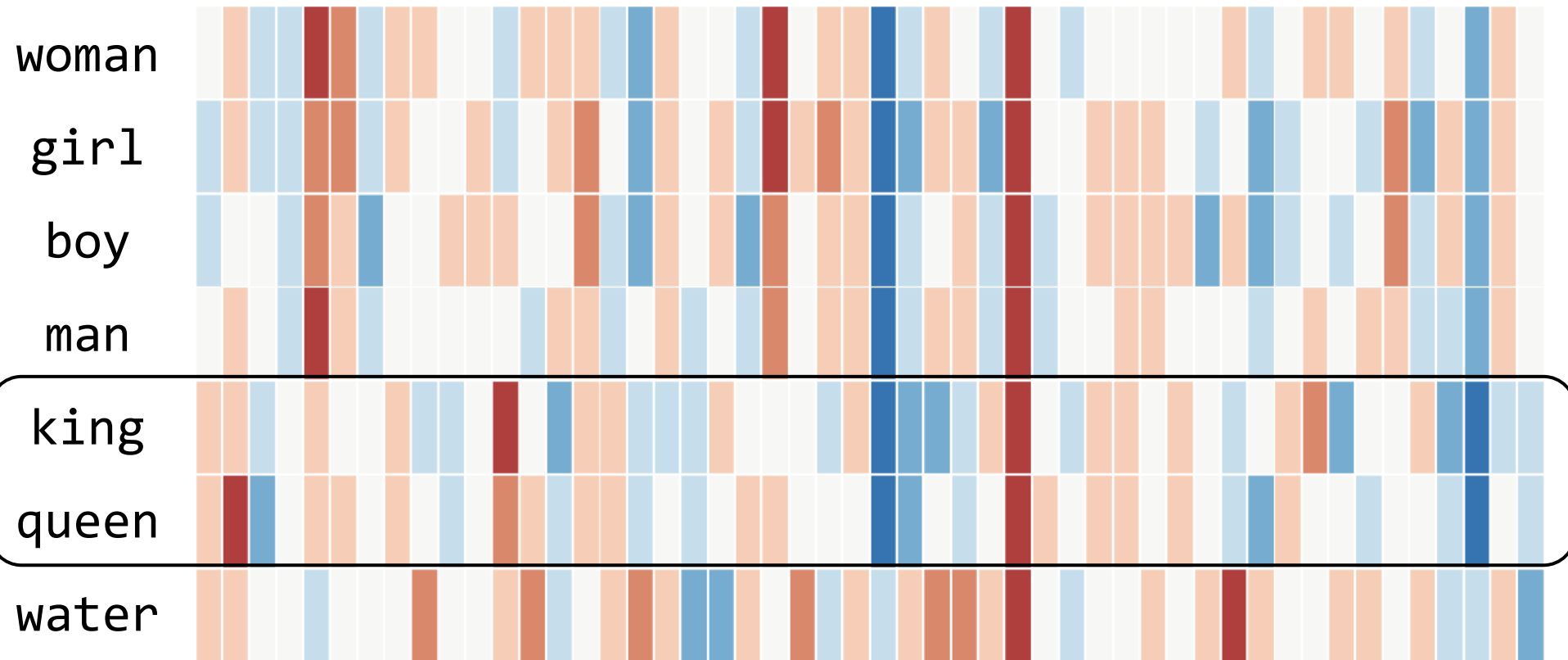


boy y girl tienen muchas semejanzas, y diferencias con woman y man.  
Algún encoding para “juventud”?



# Modelos basados en Deep Learning

Arranquemos por un ejemplo...

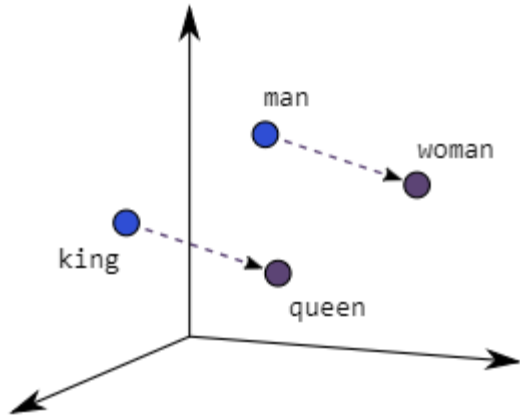


king y queen tienen muchas semejanzas, y diferencias con todo el resto.  
Algún encoding para “royalty”?



# Modelos basados en Deep Learning

Arranquemos por un ejemplo...

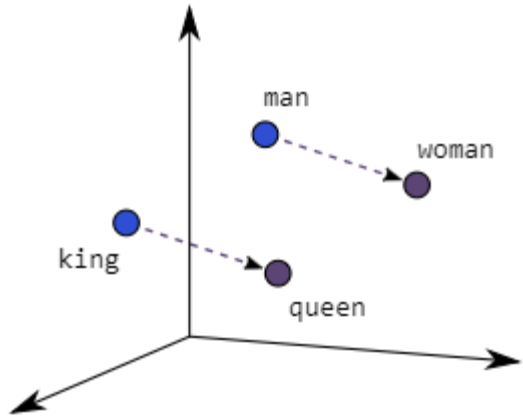


$$\text{king} - \text{man} + \text{woman} \sim \text{queen}$$

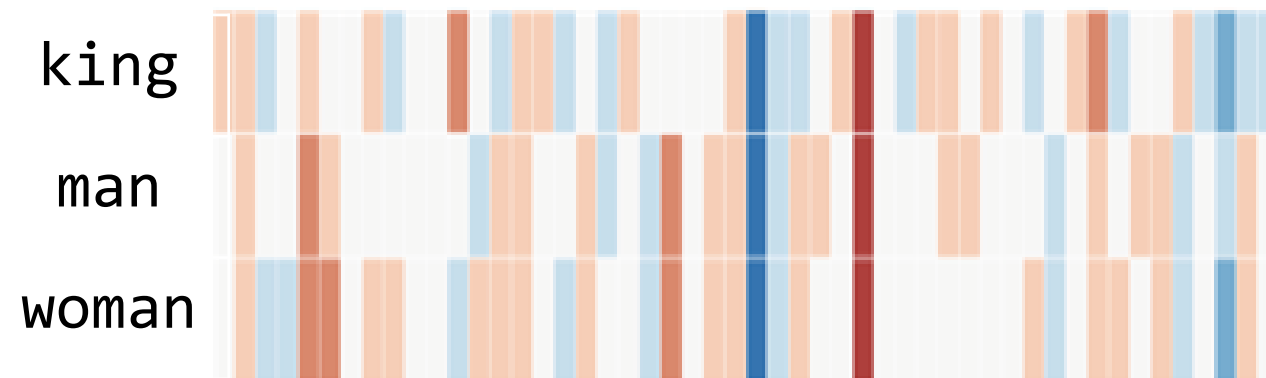


# Modelos basados en Deep Learning

Arranquemos por un ejemplo...



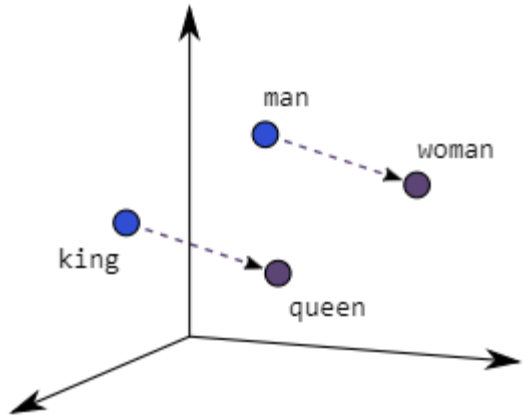
$$\text{king} - \text{man} + \text{woman} \sim \text{queen}$$





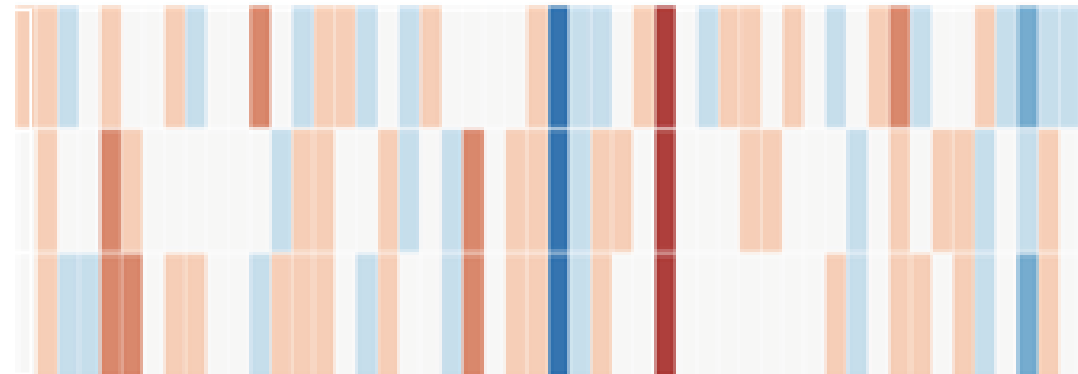
# Modelos basados en Deep Learning

Arranquemos por un ejemplo...

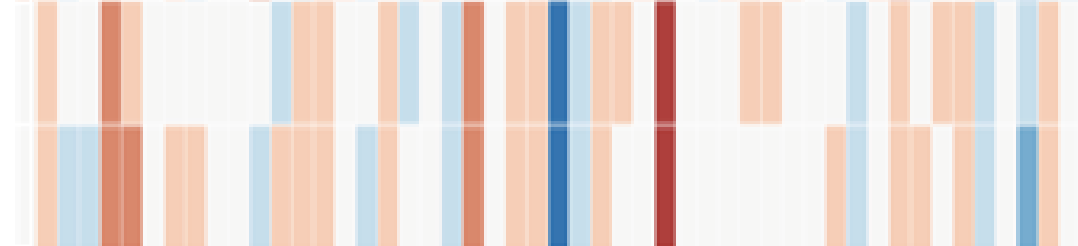


$$\text{king} - \text{man} + \text{woman} \sim \text{queen}$$

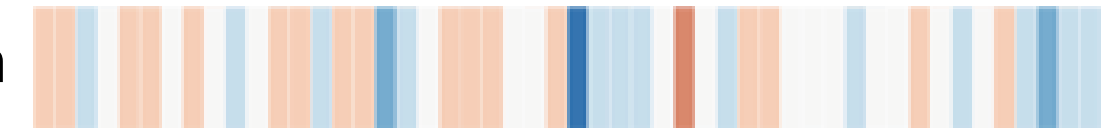
king



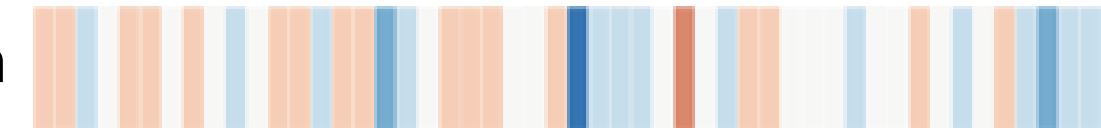
man



woman

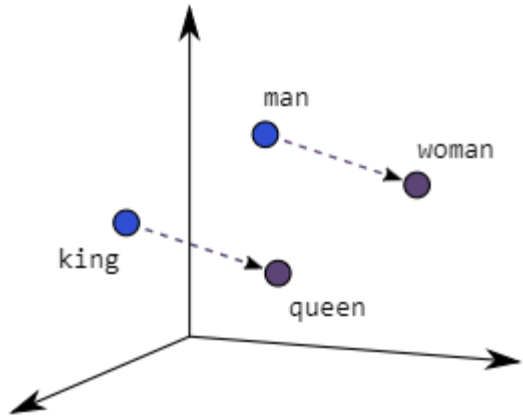


king - man + woman



# Modelos basados en Deep Learning

Arranquemos por un ejemplo...



$$\text{king} - \text{man} + \text{woman} \sim \text{queen}$$

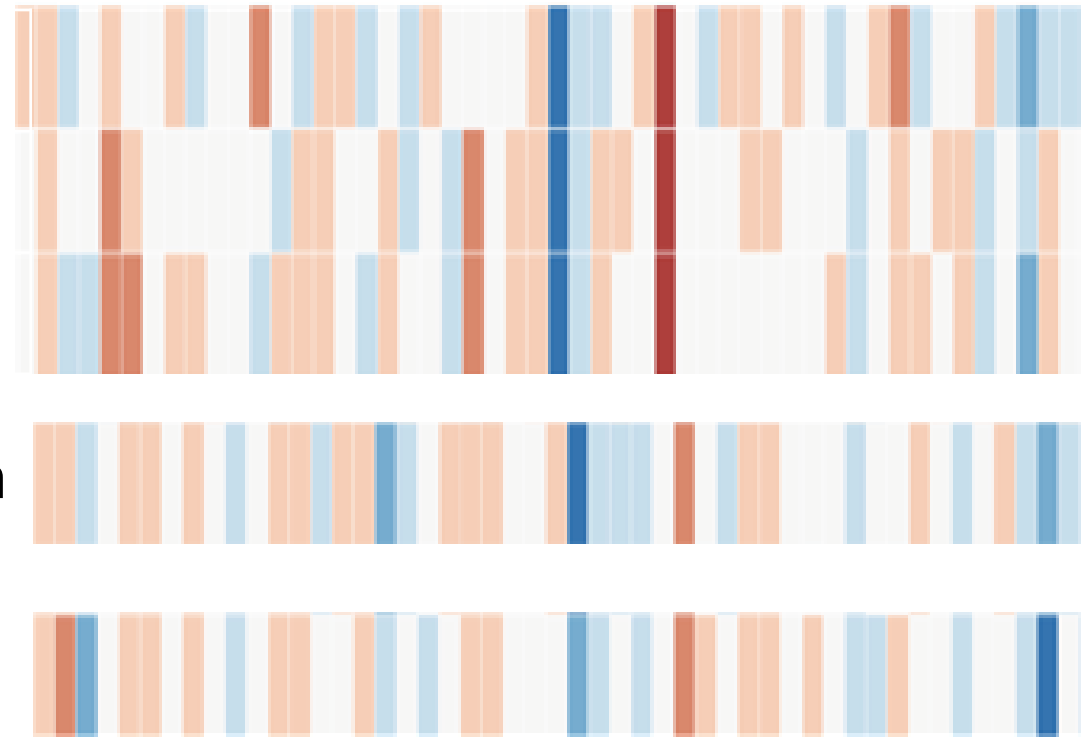
king

man

woman

king - man + woman

queen



# Modelos basados en Deep Learning

- Word embeddings es el nombre por defecto que se le da a estas representaciones, aunque también se las puede encontrar como:
  - Distributional semantic model.
  - Distributed representation.
  - Semantic vector space.
  - Word space.
- De qué hablamos cuando hablamos de embeddings?
  - Representaciones densas de elementos en la forma de vectores en un espacio de dimensionalidad reducida.

Word  
Embeddings

Sentence/Text  
Embeddings

Transformers

Nota. No vamos a ver detalles de la implementación. Eso en el próximo curso.



# Modelos basados en Deep Learning

Word  
Embeddings

Sentence/Text  
Embeddings

Transformers

Word2Vec

GloVe

FastText

ELMo

Nota. No vamos a ver detalles de la implementación. Eso en el próximo curso.



# Modelos basados en Deep Learning

## Word2vec

- Desarrollado por Google en 2013.
- Modelos no supervisados que toman grandes cantidades de textos, crean un vocabulario de posibles palabras y generan representaciones densas para cada palabra.
- Se puede especificar el tamaño de la representación.
  - Es decir, cuantas features van a representar cada una de las palabras de los textos.
  - Reduce la dimensionalidad respecto a las representaciones de bag-of-words.
- Considera las palabras como una unidad ignorando su morfología.
  - Limitación cuando los idiomas tienen vocabularios muy extensos con muchas palabras poco frecuentes.
- Dos arquitecturas:

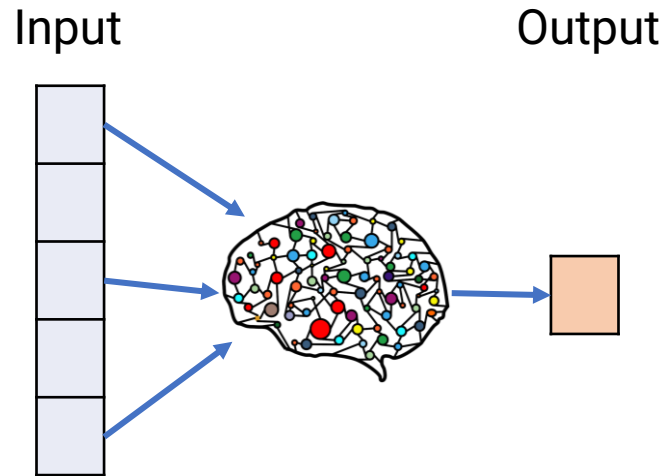
Continuous  
Bag of Words

Skip-Gram



# Modelos basados en Deep Learning

## Word2vec



CBOW trata de predecir la palabra central basado en las palabras del contexto.

Entrenamiento rápido.

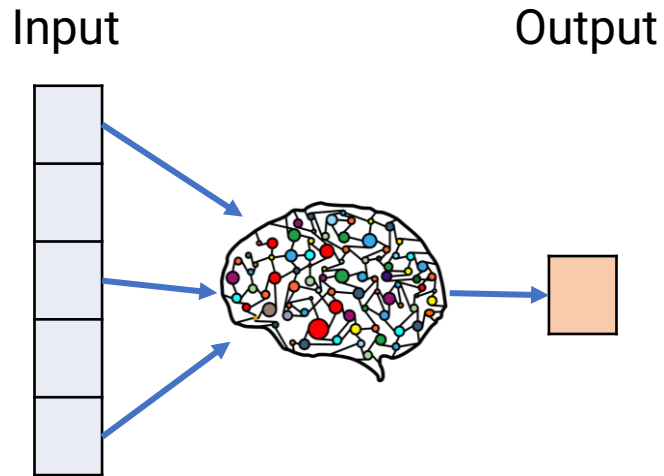
Buena representación para palabras frecuentes.

Más simple.



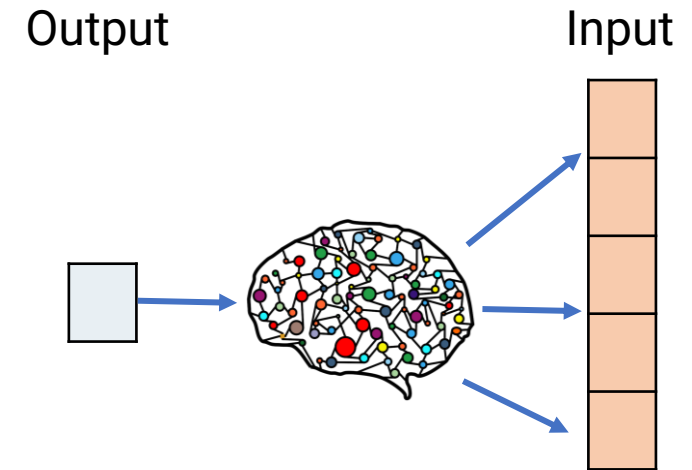
# Modelos basados en Deep Learning

## Word2vec



**CBOW** trata de predecir la palabra central basado en las palabras del contexto.

Entrenamiento rápido.  
Buena representación para palabras frecuentes.  
Más simple.



**Skip Gram** trata de predecir el contexto de una palabra central.

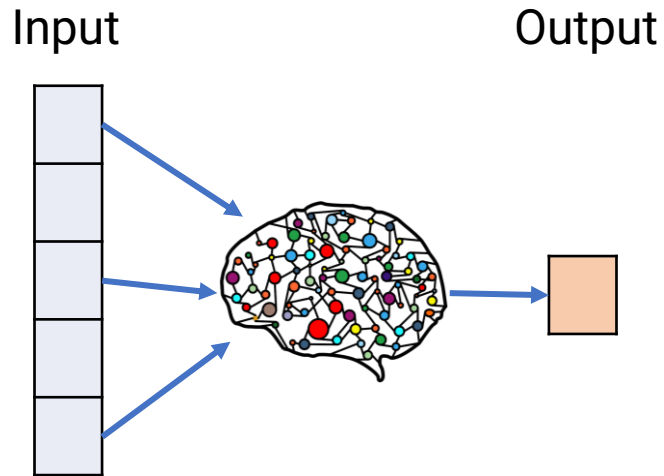
Funciona bien con poca cantidad de texto.  
Buena representación para palabras raras.



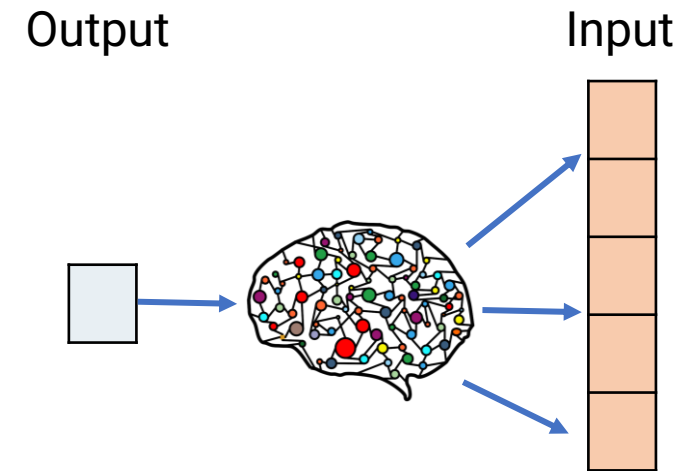


# Modelos basados en Deep Learning

## Word2vec



**CBOW** trata de predecir la palabra central basado en las palabras del contexto.



**Skip Gram** trata de predecir el contexto de una palabra central.

([quick, fox], brown)  
([the, brown], quick)  
([the, dog], lazy)

"the quick brown fox jumps  
over the lazy dog"

(brown,[quick, fox])  
(quick,[the, brown])  
(lazy,[the, dog])



# Modelos basados en Deep Learning

## Word2Vec v2

En sus definición original, resultaba en una red grande, con un proceso de entrenamiento lento y propenso al overfitting.

Se agregaron modificaciones para:

- Aprendizaje sobre frases o grupos de palabras para reducir el tamaño del vocabulario. Por ejemplo, New York es tratado como una única palabra.
- Subsampleo de palabras frecuentes para disminuir el impacto de las palabras muy frecuentes. Por ejemplo, stopwords que aparecen frecuentemente y no aportan a la creación de los word embedding.
  - Por cada palabra que encontramos en el corpus, hay una posibilidad de que sea eliminada.
  - La probabilidad está relacionada con la frecuencia de dicha palabra y el parámetro que indica cuánto subsampling se realiza.
- Negative sampling por el cual cada instancia de entrenamiento solo modifica una pequeña fracción de los pesos en lugar de todos.
  - Se selecciona una pequeña cantidad de palabras “negativas” (palabra para la cual se espera que la red de un output 0) para actualizar los pesos.
  - Palabras muy frecuentes tienen más probabilidad de ser elegidas como negative samplings.
  - Entre 5-20 palabras para datasets pequeños. Entre 2-5 para datasets grandes.



# Modelos basados en Deep Learning

## Word2Vec: Otras consideraciones

- **Tamaño de la ventana**
  - Depende de la tarea a realizar.
  - Tamaños pequeños de ventana (entre 2 y 15) dan lugar a embeddings donde una alta semejanza indica que las palabras serían “intercambiables”.
    - Tener en cuenta que, si se considera solo el contexto, los antónimos también serían intercambiables.
  - Tamaños más grandes de ventana (entre 15, 50 y más) dan lugar a embeddings donde la semejanza resulta más indicativa de la relación entre las palabras.



# Modelos basados en Deep Learning

## Word2Vec: Otras consideraciones

- **Tamaño de la ventana**
  - Depende de la tarea a realizar.
  - Tamaños pequeños de ventana (entre 2 y 15) dan lugar a embeddings donde una alta semejanza indica que las palabras serían “intercambiables”.
    - Tener en cuenta que, si se considera solo el contexto, los antónimos también serían intercambiables.
  - Tamaños más grandes de ventana (entre 15, 50 y más) dan lugar a embeddings donde la semejanza resulta más indicativa de la relación entre las palabras.
- **Modelo**
  - Si se tiene un dataset muy grande, con muchas dimensiones, el skip-gram permite alcanzar alto accuracy y buenas relaciones de semejanza semántica.
  - CBOW puede alcanzar resultados parecidos con mayor complejidad computacional.
  - Duplicar la cantidad de datos suele generar el mismo incremento en complejidad que duplicar la cantidad de dimensiones.
    - Ambos pueden mejorar el accuracy.



# Modelos basados en Deep Learning

## Word2Vec: Pre-procesamiento?

### Aplicar pre-procesamiento o no aplicarlo?

- Pre-procesamiento mínimo.
- Depende (todo depende) de qué se vaya a hacer con los vectores.
- Objetivo?
  - Reducir el tamaño del vocabulario sin eliminar contenido importante.
  - A menor tamaño de vocabulario, menor es la complejidad.
  - Más robustos los parámetros.



# Modelos basados en Deep Learning

## Word2Vec: Pre-procesamiento?

### Aplicar pre-procesamiento o no aplicarlo?

- Pre-procesamiento mínimo.
- Depende (todo depende) de qué se vaya a hacer con los vectores.
- Objetivo?
  - Reducir el tamaño del vocabulario sin eliminar contenido importante.
  - A menor tamaño de vocabulario, menor es la complejidad.
  - Más robustos los parámetros.
- Qué **SI** hacer?
  - Eliminar puntuación.
  - Convertir a lower case (discutible).
  - Reemplazar valores numéricos (por encima de un threshold) con un token único.
- Training + Test mismo pre-procesamiento!



# Modelos basados en Deep Learning

## Word2Vec: Pre-procesamiento?

### Aplicar pre-procesamiento o no aplicarlo?

- Pre-procesamiento mínimo.
- Depende (todo depende) de qué se vaya a hacer con los vectores.
- Objetivo?
  - Reducir el tamaño del vocabulario sin eliminar contenido importante.
  - A menor tamaño de vocabulario, menor es la complejidad.
  - Más robustos los parámetros.
- Qué **SI** hacer?
  - Eliminar puntuación.
  - Convertir a lower case (discutible).
  - Reemplazar valores numéricos (por encima de un threshold) con un token único.
- Training + Test mismo pre-procesamiento!

“Bush” y “bush” no son lo mismo, pero “Another” y “another” si.





# Modelos basados en Deep Learning

## GloVe

- GloVe == Global Vector.
- Desarrollado en Stanford.
- Similar a Word2Vec.
- Método no supervisado que se basa en:
  - Factorización de matrices de co-ocurrencias para disminuir dimensionalidad.
  - CBOW o Skip-gram para encontrar las representaciones.
- A diferencia de Word2Vec que trata de optimizar la tarea de encontrar el contexto o la palabra central, GloVe intenta directamente optimizar la representación.



# Modelos basados en Deep Learning

## FastText



- Desarrollado por Facebook en 2016 como una extensión y mejora de Word2Vec.
- Soporta aproximadamente 150 idiomas entrenados con Wikipedia.
- Provee modelos para identificación de idioma.
- No considera las palabras de forma individual, sino que en conjunto con la representación n-gram de sus caracteres.



# Modelos basados en Deep Learning

## FastText



- Desarrollado por Facebook en 2016 como una extensión y mejora de Word2Vec.
- Soporta aproximadamente 150 idiomas entrenados con Wikipedia.
- Provee modelos para identificación de idioma.
- No considera las palabras de forma individual, sino que en conjunto con la representación n-gram de sus caracteres.

apple



# Modelos basados en Deep Learning

## FastText



- Desarrollado por Facebook en 2016 como una extensión y mejora de Word2Vec.
- Soporta aproximadamente 150 idiomas entrenados con Wikipedia.
- Provee modelos para identificación de idioma.
- No considera las palabras de forma individual, sino que en conjunto con la representación n-gram de sus caracteres.

Si elegimos 3-grams:

apple



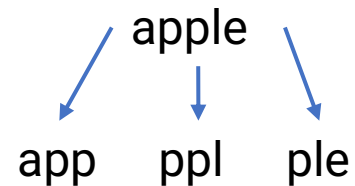
# Modelos basados en Deep Learning

## FastText



- Desarrollado por Facebook en 2016 como una extensión y mejora de Word2Vec.
- Soporta aproximadamente 150 idiomas entrenados con Wikipedia.
- Provee modelos para identificación de idioma.
- No considera las palabras de forma individual, sino que en conjunto con la representación n-gram de sus caracteres.

Si elegimos 3-grams:



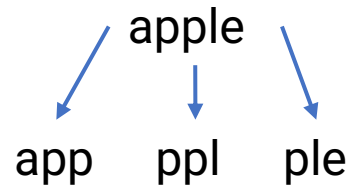
# Modelos basados en Deep Learning

## FastText



- Desarrollado por Facebook en 2016 como una extensión y mejora de Word2Vec.
- Soporta aproximadamente 150 idiomas entrenados con Wikipedia.
- Provee modelos para identificación de idioma.
- No considera las palabras de forma individual, sino que en conjunto con la representación n-gram de sus caracteres.

Si elegimos 3-grams:



El embedding de apple va a ser la suma de los emeddings de app, ppl, ple



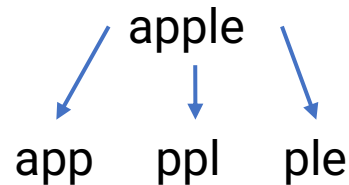
# Modelos basados en Deep Learning

## FastText



- Desarrollado por Facebook en 2016 como una extensión y mejora de Word2Vec.
- Soporta aproximadamente 150 idiomas entrenados con Wikipedia.
- Provee modelos para identificación de idioma.
- No considera las palabras de forma individual, sino que en conjunto con la representación n-gram de sus caracteres.

Si elegimos 3-grams:



El embedding de apple va a ser la suma de los emeddings de app, ppl, ple

- Las palabras poco frecuentes podrán ser representadas adecuadamente dado que es probable que sus n-grams también aparezcan como parte de otras palabras.





# Representaciones basadas en Deep Learning

## Agregando contexto

“The bank on the other end of the street was robbed”

"We had a picnic on the bank of the river"



# Representaciones basadas en Deep Learning

## Agregando contexto

“The **bank** on the other end of the street was robbed”

"We had a picnic on the **bank** of the river"

- En ambas oraciones se utiliza la palabra “bank”, pero no con el mismo significado.
  - Polisemia → dos palabras idénticas cambian el significado de acuerdo al contexto en el que se encuentran.
- Los embeddings que vimos no se adaptan a la polisemia dado que hay una única forma de representar cada palabra, independiente del contexto.



# Representaciones basadas en Deep Learning

## Agregando contexto

“The **bank** on the other end of the street was robbed”

"We had a picnic on the **bank** of the river"

- En ambas oraciones se utiliza la palabra “bank”, pero no con el mismo significado.
  - Polisemia → dos palabras idénticas cambian el significado de acuerdo al contexto en el que se encuentran.
- Los embeddings que vimos no se adaptan a la polisemia dado que hay una única forma de representar cada palabra, independiente del contexto.
- Para adaptarse, el **vector representando a cada palabra debería cambiar de acuerdo a las palabras que lo rodean.**



# Modelos basados en Deep Learning

## ELMo: Embeddings from Language Models

AllenNLP

- Objetivos:
  - Modelar las características del uso de las palabras (sintaxis y semántica).
  - Modelar como los usos varían de acuerdo a los contextos.
- En lugar de utilizar un embedding fijo para cada palabra, ELMo analiza la oración completa antes de asignar cada palabra a un embedding específico.



[Deep contextualized word representations](#)



diplomatura universitaria en  
**inteligencia artificial**



FACULTAD DE CIENCIAS  
**EXACTAS**  
UNIVERSIDAD NACIONAL DEL CENTRO  
DE LA PROVINCIA DE BUENOS AIRES

# Modelos basados en Deep Learning

## ELMo: Embeddings from Language Models

AllenNLP

- Objetivos:
  - Modelar las características del uso de las palabras (sintaxis y semántica).
  - Modelar como los usos varían de acuerdo a los contextos.
- En lugar de utilizar un embedding fijo para cada palabra, ELMo analiza la oración completa antes de asignar cada palabra a un embedding específico.
- Se basa en una LSTM bi-direccional y extrae el hidden state de cada layer para secuencia de entrada de palabras.
- Calcula la weighted sum de esos estados para obtener el embedding de cada palabra.
- Los pesos para cada hidden state son dependientes de la tarea y aprendidos.
  - Aprende a predecir cuál será la siguiente palabra en la secuencia → Language Model.

[Deep contextualized word representations](#)



diplomatura universitaria en  
**inteligencia artificial**



FACULTAD DE CIENCIAS  
**EXACTAS**  
UNIVERSIDAD NACIONAL DEL CENTRO  
DE LA PROVINCIA DE BUENOS AIRES

# Modelos basados en Deep Learning

## ELMo: Embeddings from Language Models

AllenNLP

- Objetivos:
  - Modelar las características del uso de las palabras (sintaxis y semántica).
  - Modelar como los usos varían de acuerdo a los contextos.
- En lugar de utilizar un embedding fijo para cada palabra, ELMo analiza la oración completa antes de asignar cada palabra a un embedding específico.
- Se basa en una LSTM bi-direccional y extrae el hidden state de cada layer para secuencia de entrada de palabras.
- Calcula la weighted sum de esos estados para obtener el embedding de cada palabra.
- Los pesos para cada hidden state son dependientes de la tarea y aprendidos.
  - Aprende a predecir cuál será la siguiente palabra en la secuencia → Language Model.
- Fácil de integrar con otros modelos.
- Útil para:
  - Question answering.
  - Semantic role labeling.
  - Named entity recognition.
  - Textual entailment.
  - Coreference resolution.
  - Sentiment analysis.

determinar si un enunciado es verdadero dada una premisa



[Deep contextualized word representations](#)

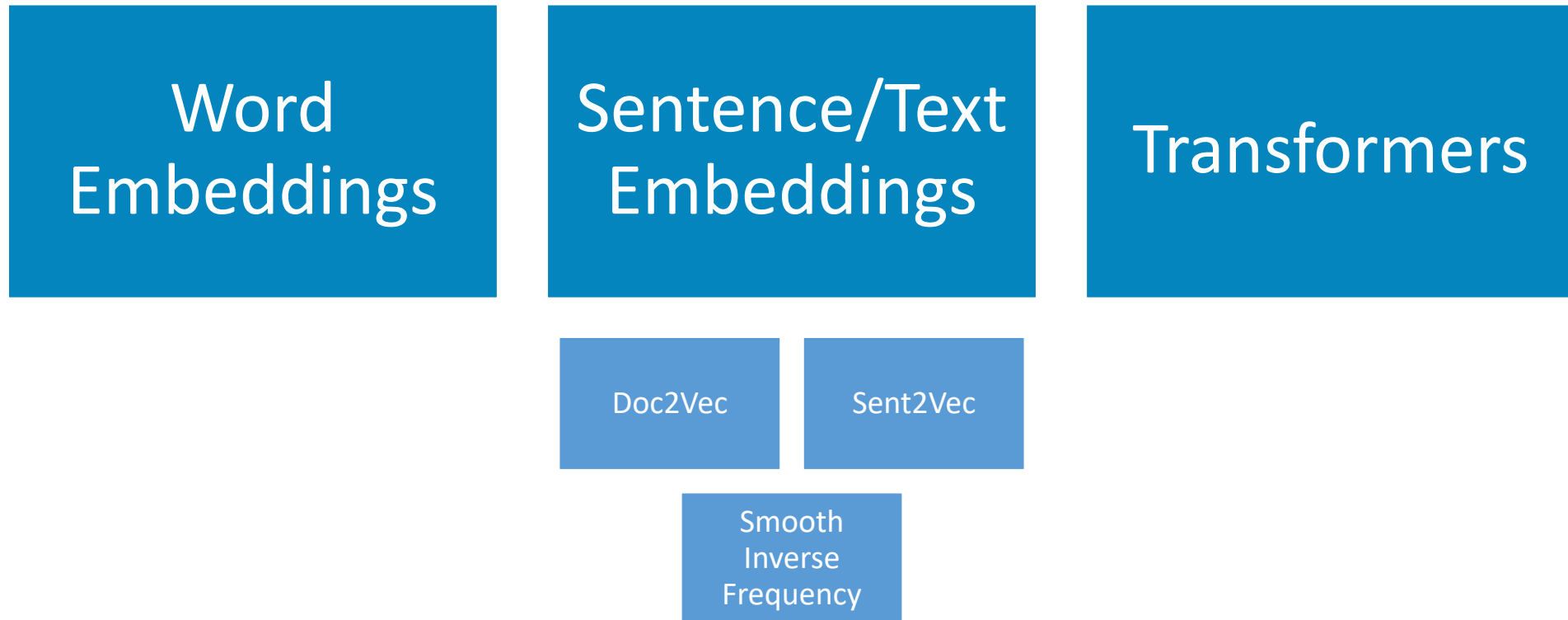


diplomatura universitaria en  
**inteligencia artificial**



FACULTAD DE CIENCIAS  
**EXACTAS**  
UNIVERSIDAD NACIONAL DEL CENTRO  
DE LA PROVINCIA DE BUENOS AIRES

# Modelos basados en Deep Learning



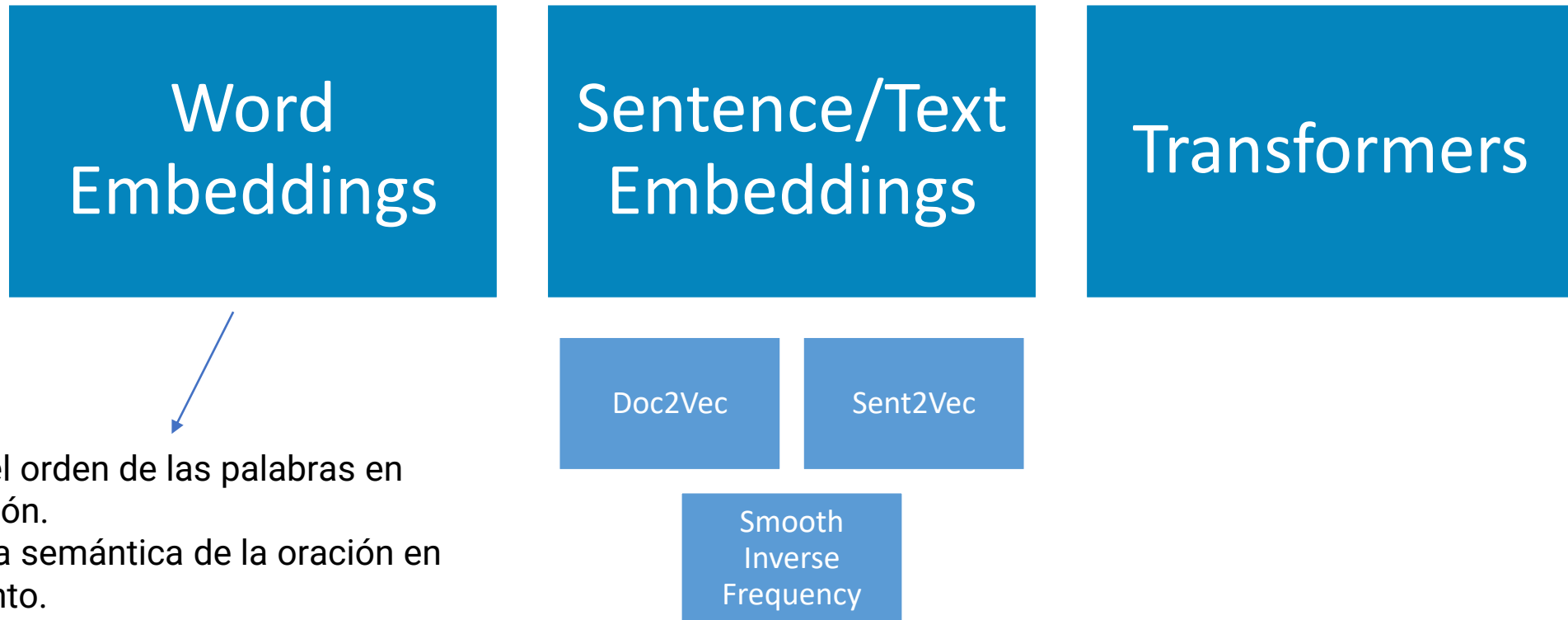
Nota. No vamos a ver detalles de la implementación. Eso en el próximo curso.





# Modelos basados en Deep Learning

Cómo capturar las relaciones de las palabras de un texto en un único vector?



Nota. No vamos a ver detalles de la implementación. Eso en el próximo curso.



# Modelos basados en Deep Learning

## Sent2vec

- Extensión de CBOW.
- No supervisado.
- El embedding de las oraciones es definido por el promedio de los embeddings correspondientes a:
  - Las palabras en la oración.
  - Los n-grams presentados en la oración.
- Se supone que de buena generalización.
- Baja complejidad computacional.
  - Permite entrenar con corpus de gran volumen.
- Útil para tareas varias de clasificación
  - Por ejemplo, clasificación de subjetividad.

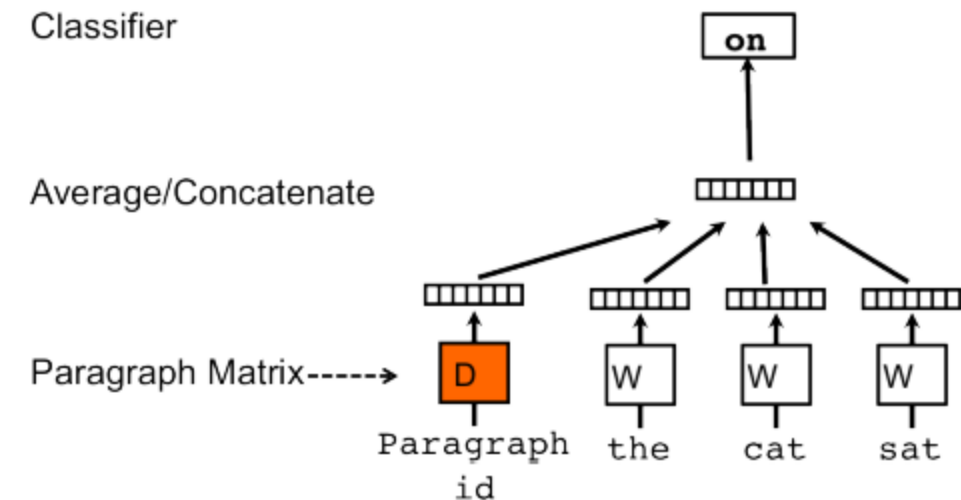
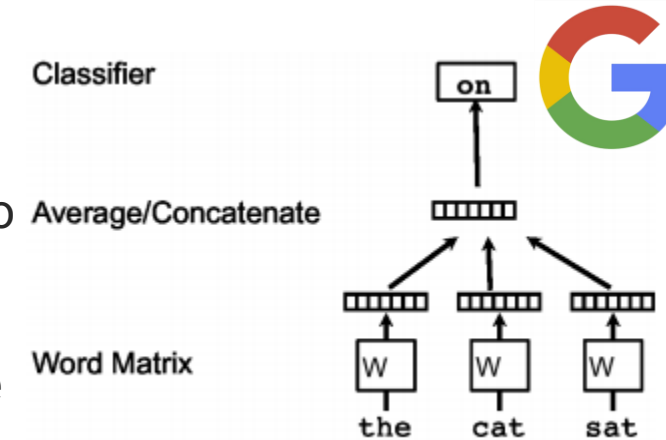
[Unsupervised Learning of Sentence Embeddings using Compositional n-Gram Features](#)



# Modelos basados en Deep Learning

## Doc2Vec

- El objetivo es crear una representación de un documento independientemente de su longitud.
- Es una extensión de word2vec CBOW al que le agregaron otro vector, el cual es único para cada document.
- Luego, cuando se entrenan los vectores de palabras, también se entrena el vector de documentos, el cual al final tiene una representación numérica del documento.
- En este modelo, la concatenación o promedio del vector con el contexto (en este caso de 3 palabras), es utilizado para predecir la siguiente.
  - El vector de documento representa la información faltante del contexto actual y puede funcionar como una memoria del contenido del documento
- Mientras los vectores de palabra representan el concepto de una palabra, el vector de document intenta representar el concepto del documento.



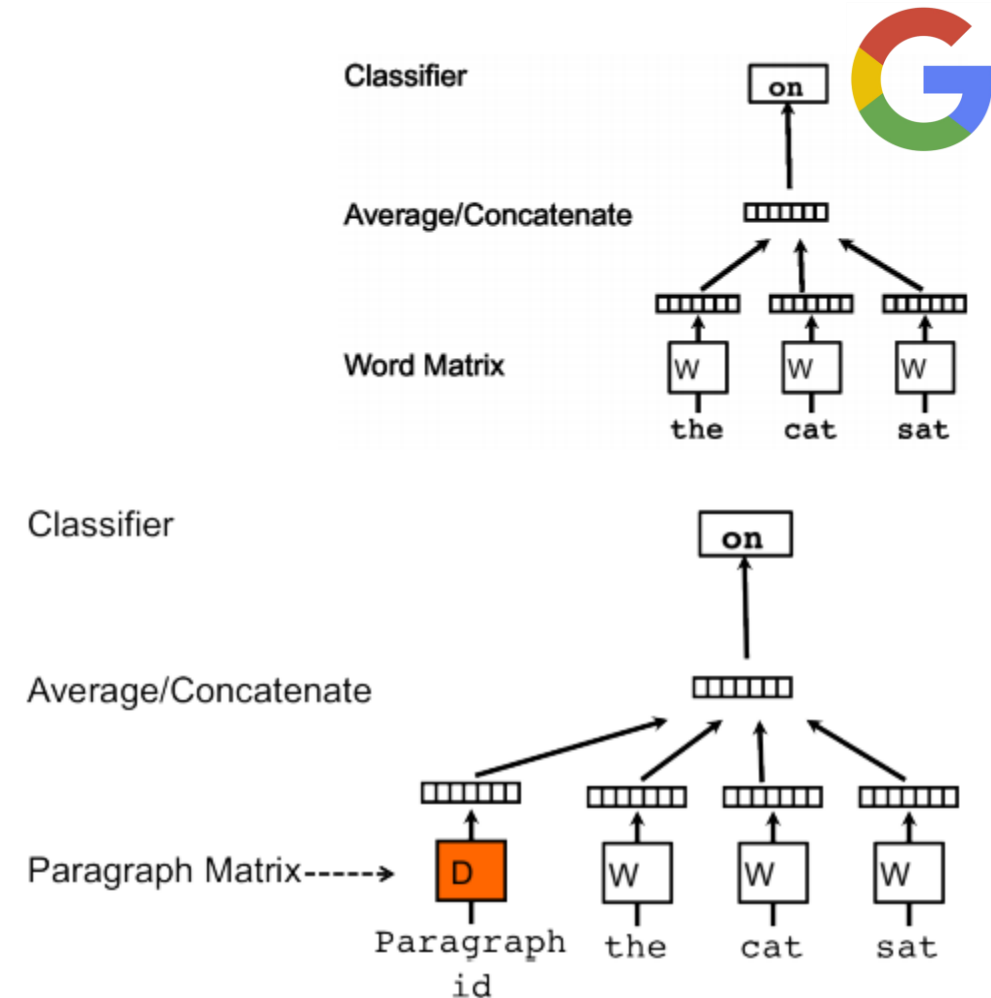
[Distributed Representations of Sentences and Documents](#)



# Modelos basados en Deep Learning

## Doc2Vec

- Son aprendidos a partir de datos no etiquetados, con lo que pueden ser utilizados para tareas que no tienen suficientes datos etiquetados.
- Heredan una característica importante de los vectores de palabras: la semántica de las palabras.
- Incorporan nociones de orden de las palabras, en el mismo sentido que lo hace un n-grams (con un n grande).
- A diferencia de los n-grams, incluyen la información en un espacio más pequeño, permitiendo una mejor escalabilidad y generalización.



[Distributed Representations of Sentences and Documents](#)



# Modelos basados en Deep Learning

## Smooth Inverse Frequency

- Parte de que promediar los vectores de palabras para obtener una representación de los documentos tiende a dar mucho peso a las palabras que son muy frecuentes pero semánticamente irrelevantes.
- Para resolver esta situación:
  - Toma un promedio ponderado de los word embeddings en una oración considerando la frecuencia estimada de las palabras en un corpus de referencia (y un parámetro a definir).
  - Calcula el componente principal (PCA) de los embeddings resultants para un conjunto de oraciones. Luego les resta las proyecciones del primer componente principal. Esto remueve las variaciones relacionadas con la frecuencia y sintaxis que son menos relevantes semánticamente.
- Le baja la relevancia a las palabras poco importantes (como los stopwords) y, a la vez, mantiene la información que contribuye a la semántica de las oraciones.
- Útil para adaptaciones de dominio, es decir, los vectores entrenados con diferentes tipos de corpus son utilizados para calcular los embeddings de las oraciones en distintos contextos.

[A simple but tough-to-beat baseline for sentence embeddings](#)



diplomatura universitaria en  
**inteligencia artificial**



FACULTAD DE CIENCIAS  
**EXACTAS**  
UNIVERSIDAD NACIONAL DEL CENTRO  
DE LA PROVINCIA DE BUENOS AIRES

# Modelos basados en Deep Learning

Word  
Embeddings

Sentence/Text  
Embeddings

Transformers

BERT

InferSent

Nota. No vamos a ver detalles de la implementación. Eso en el próximo curso.





# Modelos basados en Deep Learning

## BERT: Bidirectional Encoder Representations from Transformers

- A diferencia de los modelos direccionales en los que el texto es analizado de izquierda a derecha o de derecha a izquierda, BERT presenta un modelo bidireccional.
- Estos modelos tienen un mayor entendimiento del contexto del lenguaje y su flujo que los modelos de una única dirección.
- Permite aprender el contexto de una palabra considerando todo lo que se encuentra a su alrededor.
  - Embeddings dependientes del contexto.
- Se basa en el uso de Transformers.
- Los transformers, en su forma más simple, incluyen dos partes: un encoder (que lee el texto de entrada) y un decoder que produce la predicción de la tarea.
- Como el objetivo de BERT es generar un language model, solo el encoder es necesario.
- Dos pasos:
  - Pre-training. El modelo es entrenado con datos no etiquetados para diferentes pre-training tasks. No se basa estrictamente en la predicción de la siguiente palabra en la secuencia, sino de palabras random enmascaradas.
  - Fine tuning. El modelo es inicializado con los parámetros pre-entrenados, y luego son ajustados utilizando datos etiquetados, específicos de alguna tarea.
- Se puede utilizar para diversas tareas agregando una capa y realizando el fine tuning adecuado.



[BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding](#)



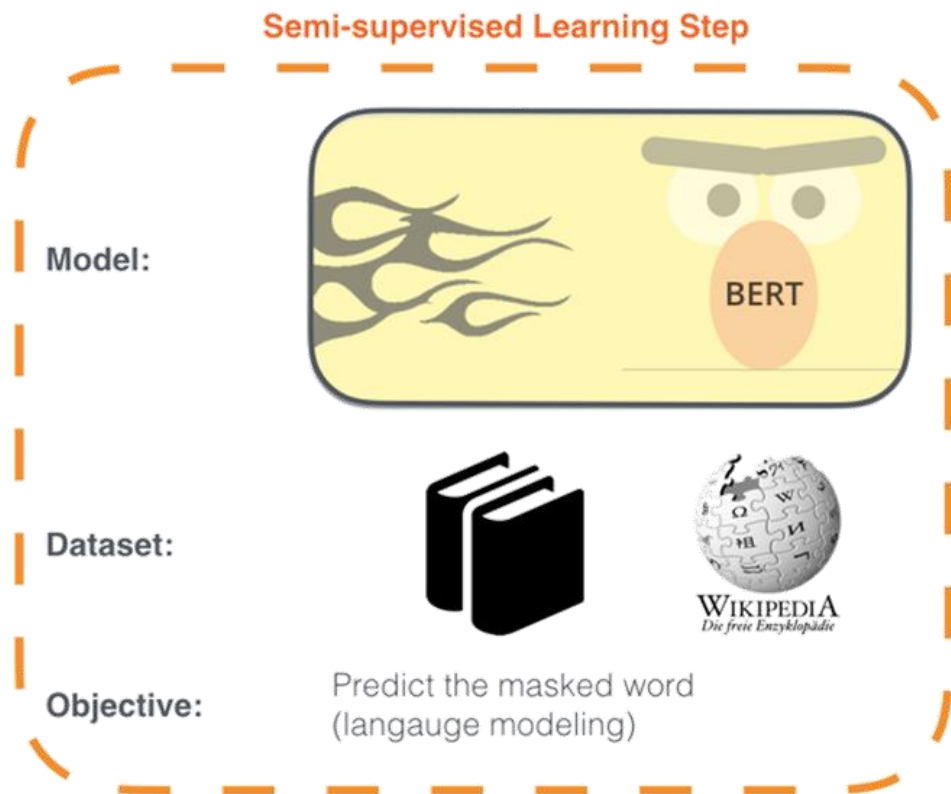


# Modelos basados en Deep Learning

## BERT: Bidirectional Encoder Representations from Transformers

**Training semi-supervisado sobre grandes cantidades de texto.**

Se entrena el modelo para una cierta tarea que permite que aprenda las particularidades del lenguaje.



# Modelos basados en Deep Learning

## BERT: Bidirectional Encoder Representations from Transformers

Entrenamiento supervisado en una tarea específica con un dataset etiquetado.

### Semi-supervised Learning Step

Model:



Dataset:



Objective:

Predict the masked word  
(language modeling)

### Supervised Learning Step

Classifier

75% Spam  
25% Not Spam

Model:  
(pre-trained  
in step #1)



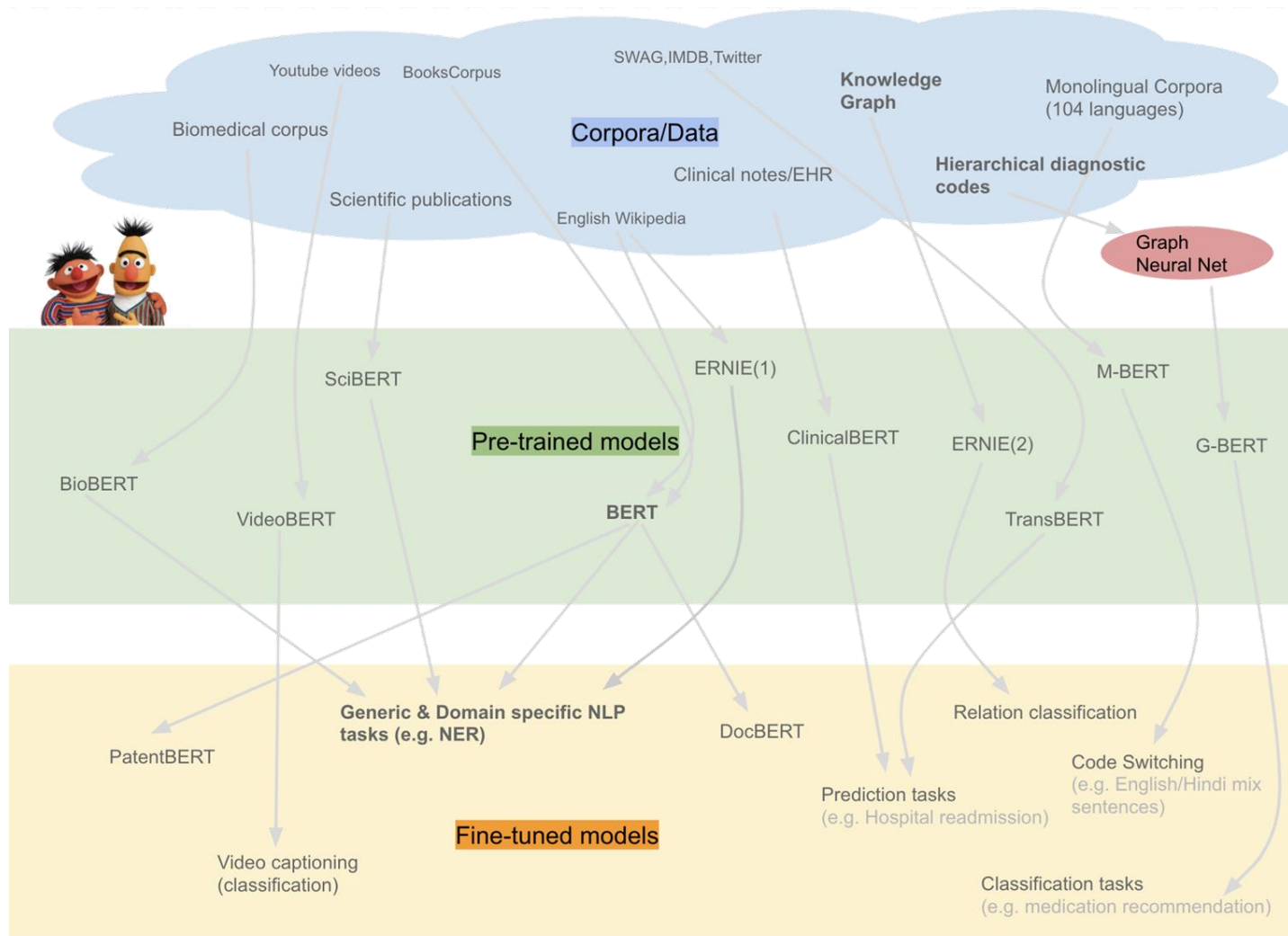
Dataset:

Email message	Class
Buy these pills	Spam
Win cash prizes	Spam
Dear Mr. Atreides, please find attached...	Not Spam



# Modelos basados en Deep Learning

## BERT: Bidirectional Encoder Representations from Transformers



- Clasificación de pares de oraciones.
- Clasificación de oraciones.
- Etiquetado en oraciones.
  - Se predicen POS tags para una palabra.
- Question/Answering
  - Dada una pregunta y el párrafo donde se la encuentra, determinar donde empieza y termina la respuesta.

# Modelos basados en Deep Learning

## InferSent



- Modelo supervisado que provee una representación semántica de oraciones.
- El entrenamiento está basado en tareas de inferencia (NLI).
- Particularmente, en pares de oraciones etiquetadas como: entailment, contradicción, neutral.
- Probaron diferentes arquitecturas. La que mejores resultados dio:
  - Se basa en una LSTM bi-direccional que calcula los  $n$  vectores para las  $n$  palabras y cada vector es la concatenación de la salida de un LSTM forward y una LSTM backward que lee la oración en la dirección opuesta.
  - Luego, max pooling es aplicado a los vectores concatenados para formar la representación final.
- Útil para:
  - Clasificación binaria y multi-clase.
  - Semejanza semántica.
  - Detección de parafraseo.
  - Tareas de imágenes y captions.

[Supervised Learning of Universal Sentence Representations from Natural Language Inference Data](#)



Determinar cuán “cercanos” son dos textos

$$\text{sim}(\text{📄}, \text{📄}) = ?$$

Determinar cuán “cercanos” son dos textos

$$\text{sim}(\text{📄}, \text{📄}) = ?$$

- Ranking de documentos.
- Information retrieval.
- Clasificación.
- Clustering.
- Desambigüación.



Determinar cuán “cercanos” son dos textos

$$\text{sim}(\text{[icono de documento]}, \text{[icono de documento]}) = ?$$

- Ranking de documentos.
  - Information retrieval.
  - Clasificación.
  - Clustering.
  - Desambigüación.
- Hay multiples nociones de semejanza, dependiendo del dominio y de la aplicación.
  - Morfológica.
  - Semejanza de spelling.
  - Sinonimia.
  - Homofonia.
  - Semántica.



Determinar cuán “cercaños” son dos textos

$$\text{sim}(\text{[icono de documento]}, \text{[icono de documento]}) = ?$$

- Ranking de documentos.
  - Information retrieval.
  - Clasificación.
  - Clustering.
  - Desambigüación.
- Hay multiples nociones de semejanza, dependiendo del dominio y de la aplicación.
  - Morfológica.
  - Semejanza de spelling.
  - Sinonimia.
  - Homofonia.
  - Semántica.

Se pueden extender a  
oraciones y documentos!



# Semejanza de texto

## Semejanza Vs. Distancia



- Las distancias son inversamente proporcionales a las semejanzas.
- Las semejanzas se “diseñan” para el rango de valores [0,1].

$$semejanza = \frac{1}{1 + distancia}$$

$$distancia = \frac{1}{semejanza} - 1$$

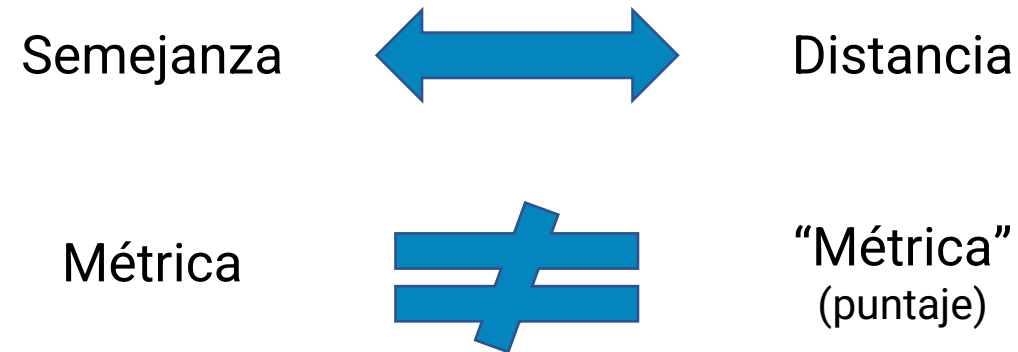
$$semejanza = 1 - distancia$$
$$distancia = 1 - semejanza$$

Las más comunes



# Semejanza de texto

## Semejanza Vs. Distancia

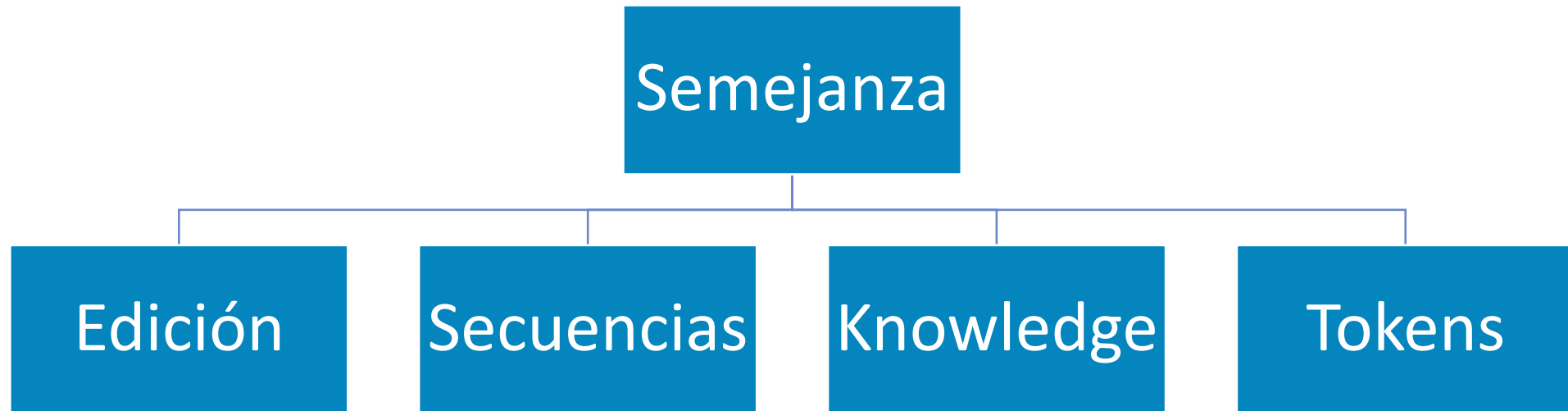


Las métricas deben cumplir con algunas Propiedades matemáticas que las “métricas” no:

- No negativas.  $\text{metric}(A, B) \geq 0$
- Dos elementos son iguales si la métrica entre ellos es 0 (distancia) o 1 (semejanza).  
 $\text{metric}(A, B) == 0: \text{assert}(A == B)$
- Simetría. Las métricas no tienen dirección.  $\text{metric}(A, B) = \text{metric}(B, A)$
- Desigualdad triangular.  $\text{metric}(A, C) \leq \text{metric}(A, B) + \text{metric}(B, C)$

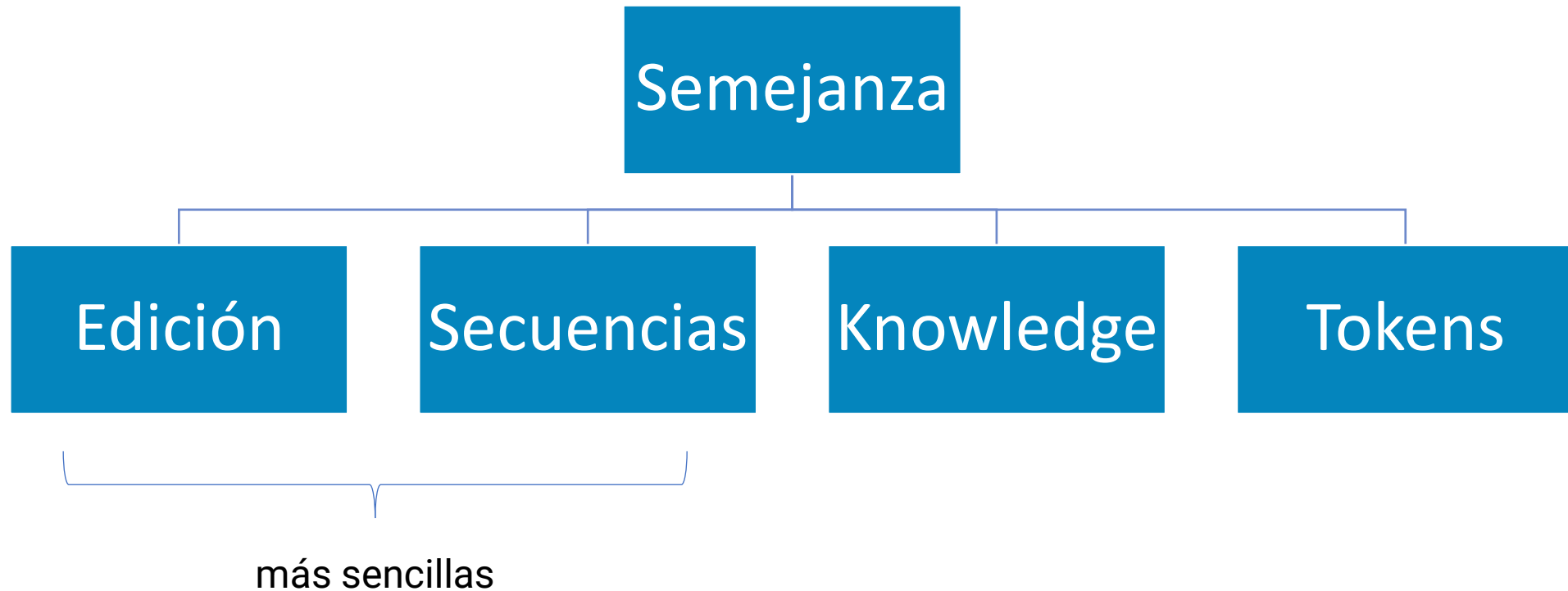
# Semejanza de texto

## Variantes



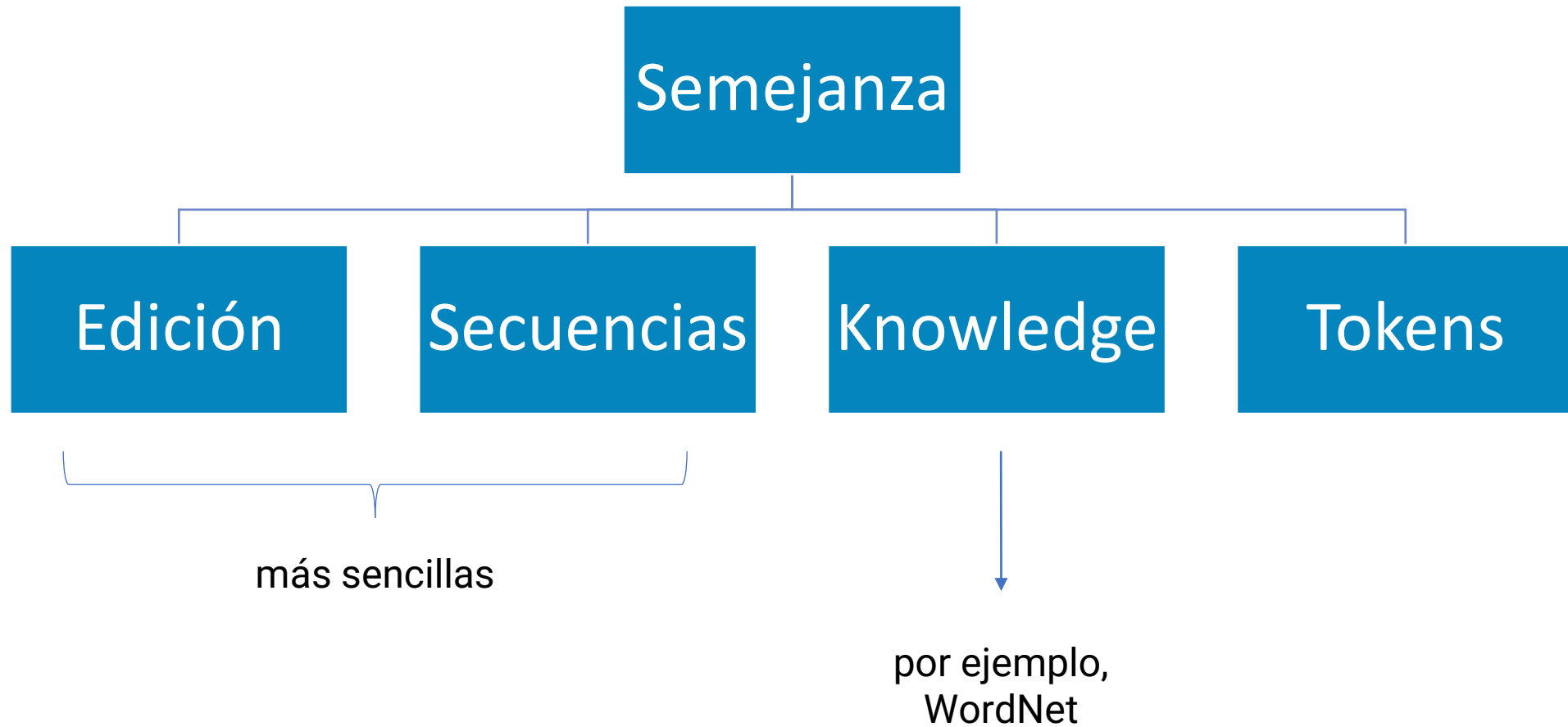
# Semejanza de texto

## Variantes



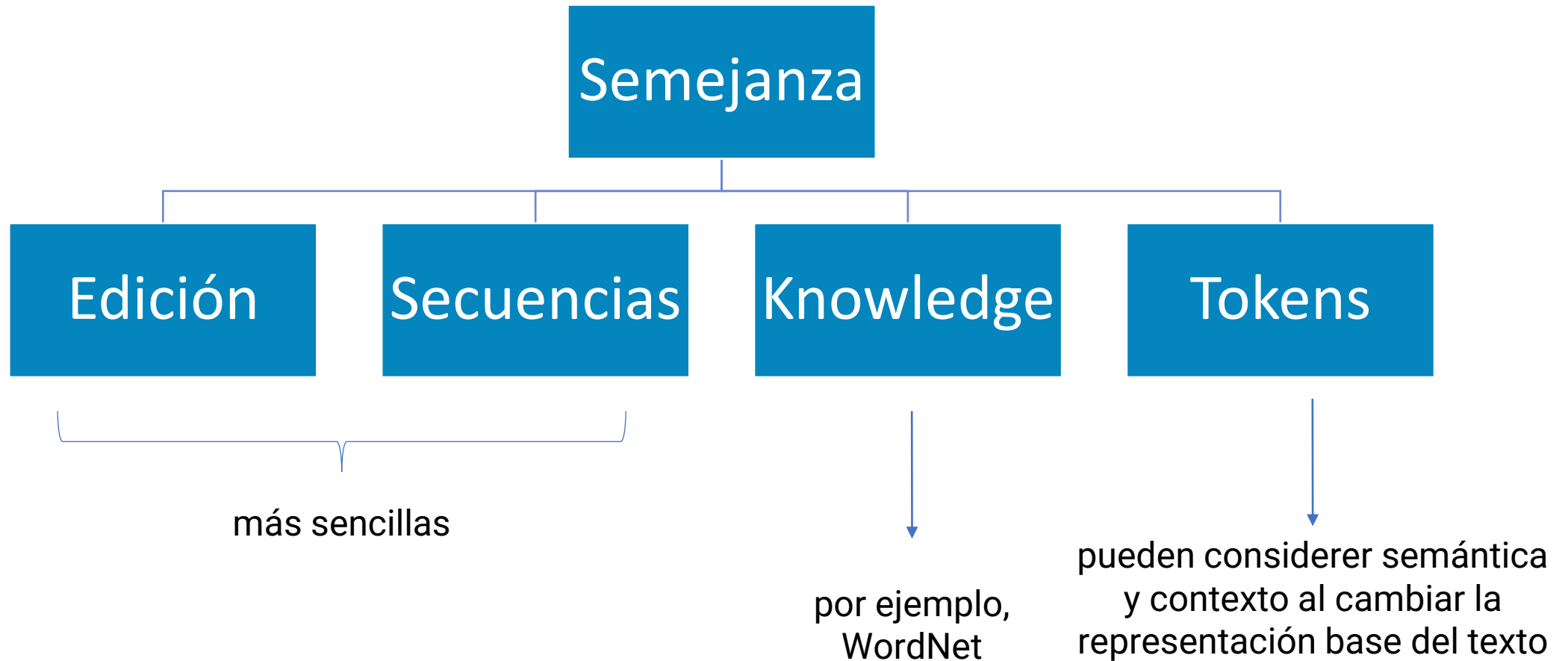
# Semejanza de texto

## Variantes



# Semejanza de texto

## Variantes





# Semejanza de texto

## Basadas en edición

- Compara dos Strings contando la cantidad minima de operaciones requeridas para transformar un String en el otro.

Hamming

Levenshtein

Jaro-  
Winkler

...



# Semejanza de texto

## Basadas en edición

- Compara dos Strings contando la cantidad minima de operaciones requeridas para transformar un String en el otro.

Hamming

Levenshtein

Jaro-  
Winkler

...

`edit(cat,hat) = 1`

`edit(map,cap) = 2`



# Semejanza de texto

## Basadas en edición

- Compara dos Strings contando la cantidad minima de operaciones requeridas para transformar un String en el otro.

Hamming

Levenshtein

Jaro-  
Winkler

...

```
edit(cat,hat) = 1
```

```
edit(map,cat) = 2
```

```
edit(lemon,apple) > edit(lemon,moon)
```



# Semejanza de texto

## Basadas en edición

- Compara dos Strings contando la cantidad minima de operaciones requeridas para transformar un String en el otro.

Hamming

Levenshtein

Jaro-  
Winkler

...

```
edit(cat,hat) = 1
```

```
edit(map,cat) = 2
```

```
edit(lemon,apple) > edit(lemon,moon)
```

Aunque diríamos que dos frutas se encuentran más relacionadas que una fruta y la luna



# Semejanza de texto

## Basadas en edición

- Compara dos Strings contando la cantidad minima de operaciones requeridas para transformar un String en el otro.

Hamming

Levenshtein

Jaro-  
Winkler

...

### Pros

- Simple
- Útil para Strings cortos.
- Útil cuando la semántica no es tan importante como la forma de escritura.

### Cons

- Ineficiente para Strings largos.
- Computacionalmente complejo.
- Compara caracteres, con lo que no tiene en cuenta semántica.



# Semejanza de texto

## Basadas en edición

### Hamming

- Frecuentemente utilizada en teoría de la información y comunicación.
  - Detección/corrección de errores cuando se transmite información.
- Asume que son de igual longitud.
  - Difícil de usar cuando las longitudes son distintas.
- Se define como la cantidad de posiciones que tienen diferentes caracteres o símbolos.
  - Todas las diferencias valen lo mismo.
  - Medir la distancia entre variables categóricas.
- Se puede normalizar por la longitud.

1	0	1	1	0	0
1	1	1	0	0	0



# Semejanza de texto

## Basadas en edición

### Hamming

- Frecuentemente utilizada en teoría de la información y comunicación.
  - Detección/corrección de errores cuando se transmite información.
- Asume que son de igual longitud.
  - Difícil de usar cuando las longitudes son distintas.
- Se define como la cantidad de posiciones que tienen diferentes caracteres o símbolos.
  - Todas las diferencias valen lo mismo.
  - Medir la distancia entre variables categóricas.
- Se puede normalizar por la longitud.

1	0	1	1	0	0
1	1	1	0	0	0

Nat*i*ral  
Langua*j*e  
proc*cc*esing

Nat*u*ral  
Language  
proc*ess*ing

$$\text{Hamming}(u, v) = \sum_{i=1}^n (u_i \neq v_i)$$

$$\text{Hamming\_Norm}(u, v) = \frac{\sum_{i=1}^n (u_i \neq v_i)}{n}$$





# Semejanza de texto

## Basadas en edición

### Levenshtein

- Se define como la cantidad de ediciones que se necesitan para convertir un String en el otro.
  - Adición, eliminación o sustitución.
  - Cada edición afecta a un único caracter.
- No asume igualdad en longitud.
- El valor mínimo es la diferencia entre las dos longitudes.
- El valor máximo es la longitud más larga.
- Si los String son iguales, la distancia es cero.
- Hamming es una cota superior si los Strings tienen la misma longitud.

L	E	V	E	N	S	H	T	E	I	N	
L	E	V	I	N	N	S	T	E	I	H	N
-	-	-	S	-	R	-	I			E	
0	0	0	1	1	2	2	3	3	3	4	4



# Semejanza de texto

## Basadas en edición

### Levenshtein

- Se define como la cantidad de ediciones que se necesitan para convertir un String en el otro.
  - Adición, eliminación o sustitución.
  - Cada edición afecta a un único caracter.
- No asume igualdad en longitud.
- El valor mínimo es la diferencia entre las dos longitudes.
- El valor máximo es la longitud más larga.
- Si los String son iguales, la distancia es cero.
- Hamming es una cota superior si los Strings tienen la misma longitud.

L	E	V	E	N	S	H	T	E	I	N	
L	E	V	I	N	N	S	T	E	I	H	N
-	-	-	S	-	R	-	I			E	
0	0	0	1	1	2	2	3	3	3	4	4

$$Le_{u,v}(i,j) = \begin{cases} \max(i,j), & \min(i,j) = 0 \\ \min \begin{cases} Le_{u,v}(i-1,j) + 1 \\ Le_{u,v}(i,j-1) + 1 \\ Le_{u,v}(i-1,j-1) + 1_{u_i \neq v_j} \end{cases}, & otherwise \end{cases}$$



# Semejanza de texto

## Basadas en edición

### Levenshtein

$$Le_{u,v}(i,j) = \begin{cases} \max(i,j), & \min(i,j) = 0 \\ \min \begin{cases} Le_{u,v}(i-1,j) + 1 \\ Le_{u,v}(i,j-1) + 1 \\ Le_{u,v}(i-1,j-1) + 1_{u_i \neq v_j} \end{cases}, & otherwise \end{cases}$$

- La forma más común (y eficiente) de encontrar la distancia es utilizando una matriz.
- Cada celda contiene la distancia entre el i-caracter de una palabra y el j-caracter de la otra.
- La matriz se completa de izquierda a derecha y de arriba para bajo.
- El costo de cada operación se asume. 1.
- La distancia total se encuentra en la celda de la esquina inferior derecha.



# Semejanza de texto

## Basadas en edición

### Levenshtein

$$Le_{u,v}(i,j) = \begin{cases} \max(i,j), & \min(i,j) = 0 \\ \min \begin{cases} Le_{u,v}(i-1,j) + 1 \\ Le_{u,v}(i,j-1) + 1 \\ Le_{u,v}(i-1,j-1) + 1_{u_i \neq v_j} \end{cases}, & otherwise \end{cases}$$

- La forma más común (y eficiente) de encontrar la distancia es utilizando una matriz.
- Cada celda contiene la distancia entre el i-caracter de una palabra y el j-caracter de la otra.
- La matriz se completa de izquierda a derecha y de arriba para bajo.
- El costo de cada operación se asume. 1.
- La distancia total se encuentra en la celda de la esquina inferior derecha.

Vamos a encontrar la distancia entre:

BELIEVE

BELEIVE



# Semejanza de texto

## Basadas en edición

### Levenshtein

$$Le_{u,v}(i,j) = \begin{cases} \max(i,j), & \min(i,j) = 0 \\ \min \begin{cases} Le_{u,v}(i-1,j) + 1 \\ Le_{u,v}(i,j-1) + 1 \\ Le_{u,v}(i-1,j-1) + 1_{u_i \neq v_j} \end{cases}, & otherwise \end{cases}$$

- La forma más común (y eficiente) de encontrar la distancia es utilizando una matriz.
- Cada celda contiene la distancia entre el i-caracter de una palabra y el j-caracter de la otra.
- La matriz se completa de izquierda a derecha y de arriba para bajo.
- El costo de cada operación se asume. 1.
- La distancia total se encuentra en la celda de la esquina inferior derecha.

Vamos a encontrar la distancia entre:

BELIEVE

BELIEVE

BELIEVE

BELEIVE

BELEIVE

BELEIVE

La distancia es 2



# Semejanza de texto

## Basadas en edición

Levenshtein

$$Le_{u,v}(i,j) = \begin{cases} \max(i,j), & \min(i,j) = 0 \\ \min \begin{cases} Le_{u,v}(i-1,j) + 1 \\ Le_{u,v}(i,j-1) + 1 \\ Le_{u,v}(i-1,j-1) + 1_{u_i \neq v_j} \end{cases}, & otherwise \end{cases}$$

BELIEVE

BELEIVE

		i	B	E	L	I	E	V	E
j									
B									
E									
L									
E									
I									
V									
E									



# Semejanza de texto

## Basadas en edición

Levenshtein

$$Le_{u,v}(i,j) = \begin{cases} \max(i,j), & \min(i,j) = 0 \\ \min \begin{cases} Le_{u,v}(i-1,j) + 1 \\ Le_{u,v}(i,j-1) + 1 \\ Le_{u,v}(i-1,j-1) + 1_{u_i \neq v_j} \end{cases}, & otherwise \end{cases}$$

BELIEVE  
BELEIVE

		i	B	E	L	I	E	V	E
j									
B									
E									
L									
E									
I									
V									
E									

Vamos a completar para los casos:  $i=0$ ,  $j=0$

- Es decir, nada de uno con todos los otros.
- No asume que tienen la misma longitud.





# Semejanza de texto

## Basadas en edición

Levenshtein

$$Le_{u,v}(i,j) = \begin{cases} \max(i,j), & \min(i,j) = 0 \\ \min \begin{cases} Le_{u,v}(i-1,j) + 1 \\ Le_{u,v}(i,j-1) + 1 \\ Le_{u,v}(i-1,j-1) + 1_{u_i \neq v_j} \end{cases}, & otherwise \end{cases}$$

BELIEVE  
BELEIVE

	i	B	E	L	I	E	V	E
j	0	1	2	3	4	5	6	7
B	1							
E	2							
L	3							
E	4							
I	5							
V	6							
E	7							

Vamos a completar para los casos:  $i=0$ ,  $j=0$

- Es decir, nada de uno con todos los otros.
- No asume que tienen la misma longitud.



# Semejanza de texto

## Basadas en edición

Levenshtein

$$Le_{u,v}(i,j) = \begin{cases} \max(i,j), & \min(i,j) = 0 \\ \min \begin{cases} Le_{u,v}(i-1,j) + 1 \\ Le_{u,v}(i,j-1) + 1 \\ Le_{u,v}(i-1,j-1) + 1_{u_i \neq v_j} \end{cases}, & otherwise \end{cases}$$

BELIEVE

BELEIVE

	i	B	E	L	I	E	V	E
j	0	1	2	3	4	5	6	7
B	1							
E	2							
L	3							
E	4							
I	5							
V	6							
E	7							

- Seguimos con  $i=1$  y los  $j > 0$ :

$$Le(i=1, j=1) = \min \begin{cases} Le_{u,v}(0,1) + 1 \\ Le_{u,v}(1,0) + 1 \\ Le_{u,v}(0,0) + 1_{u_i \neq v_j} \end{cases}$$

$$Le(i=1, j=2) = \min \begin{cases} Le_{u,v}(0,2) + 1 \\ Le_{u,v}(0,1) + 1 \\ Le_{u,v}(0,1) + 1_{u_i \neq v_j} \end{cases}$$

$$Le(i=1, j=3) = \min \begin{cases} Le_{u,v}(0,3) + 1 \\ Le_{u,v}(1,2) + 1 \\ Le_{u,v}(0,2) + 1_{u_i \neq v_j} \end{cases}$$

$$Le(i=1, j=4) = \min \begin{cases} Le_{u,v}(0,4) + 1 \\ Le_{u,v}(1,3) + 1 \\ Le_{u,v}(0,3) + 1_{u_i \neq v_j} \end{cases}$$

$$Le(i=1, j=5) = \min \begin{cases} Le_{u,v}(0,5) + 1 \\ Le_{u,v}(1,4) + 1 \\ Le_{u,v}(0,4) + 1_{u_i \neq v_j} \end{cases}$$

$$Le(i=1, j=6) = \min \begin{cases} Le_{u,v}(0,6) + 1 \\ Le_{u,v}(1,5) + 1 \\ Le_{u,v}(0,5) + 1_{u_i \neq v_j} \end{cases}$$

$$Le(i=1, j=7) = \min \begin{cases} Le_{u,v}(0,7) + 1 \\ Le_{u,v}(1,6) + 1 \\ Le_{u,v}(0,6) + 1_{u_i \neq v_j} \end{cases}$$



# Semejanza de texto

## Basadas en edición

Levenshtein

$$Le_{u,v}(i,j) = \begin{cases} \max(i,j), & \min(i,j) = 0 \\ \min \begin{cases} Le_{u,v}(i-1,j) + 1 \\ Le_{u,v}(i,j-1) + 1 \\ Le_{u,v}(i-1,j-1) + 1_{u_i \neq v_j} \end{cases}, & otherwise \end{cases}$$

BELIEVE  
BELEIVE

	i	B	E	L	I	E	V	E
j	0	1	2	3	4	5	6	7
B	1	0	1	2	3	4	5	6
E	2							
L	3							
E	4							
I	5							
V	6							
E	7							

- Seguimos con  $i=1$  y los  $j > 0$ :

$$Le(i=1, j=1) = \min \begin{cases} Le_{u,v}(0,1) + 1 = 2 \\ Le_{u,v}(1,0) + 1 = 2 \\ Le_{u,v}(0,0) + 1_{u_i \neq v_j} = 0 \end{cases}$$

$$Le(i=1, j=2) = \min \begin{cases} Le_{u,v}(0,2) + 1 = 3 \\ Le_{u,v}(1,1) + 1 = 1 \\ Le_{u,v}(0,1) + 1_{u_i \neq v_j} = 2 \end{cases}$$

$$Le(i=1, j=3) = \min \begin{cases} Le_{u,v}(0,3) + 1 = 4 \\ Le_{u,v}(1,2) + 1 = 2 \\ Le_{u,v}(0,2) + 1_{u_i \neq v_j} = 3 \end{cases}$$

$$Le(i=1, j=4) = \min \begin{cases} Le_{u,v}(0,4) + 1 = 5 \\ Le_{u,v}(1,3) + 1 = 3 \\ Le_{u,v}(0,3) + 1_{u_i \neq v_j} = 4 \end{cases}$$

$$Le(i=1, j=5) = \min \begin{cases} Le_{u,v}(0,5) + 1 = 5 \\ Le_{u,v}(1,4) + 1 = 4 \\ Le_{u,v}(0,4) + 1_{u_i \neq v_j} = 5 \end{cases}$$

$$Le(i=1, j=6) = \min \begin{cases} Le_{u,v}(0,6) + 1 = 7 \\ Le_{u,v}(1,5) + 1 = 5 \\ Le_{u,v}(0,5) + 1_{u_i \neq v_j} = 6 \end{cases}$$

$$Le(i=1, j=7) = \min \begin{cases} Le_{u,v}(0,7) + 1 = 8 \\ Le_{u,v}(1,6) + 1 = 6 \\ Le_{u,v}(0,6) + 1_{u_i \neq v_j} = 5 \end{cases}$$



# Semejanza de texto

## Basadas en edición

Levenshtein

$$Le_{u,v}(i,j) = \begin{cases} \max(i,j), & \min(i,j) = 0 \\ \min \begin{cases} Le_{u,v}(i-1,j) + 1 \\ Le_{u,v}(i,j-1) + 1 \\ Le_{u,v}(i-1,j-1) + 1_{u_i \neq v_j} \end{cases}, & otherwise \end{cases}$$

BELIEVE  
BELEIVE

		i	B	E	L	I	E	V	E
j	0	1	2	3	4	5	6	7	
B	1	0	1	2	3	4	5	6	
E	2								
L	3								
E	4								
I	5								
V	6								
E	7								

- Seguimos con  $i=2$  y los  $j > 0$ : ...



# Semejanza de texto

## Basadas en edición

Levenshtein

$$Le_{u,v}(i,j) = \begin{cases} \max(i,j), & \min(i,j) = 0 \\ \min \begin{cases} Le_{u,v}(i-1,j) + 1 \\ Le_{u,v}(i,j-1) + 1 \\ Le_{u,v}(i-1,j-1) + 1_{u_i \neq v_j} \end{cases}, & otherwise \end{cases}$$

BELIEVE  
BELEIVE


		i	B	E	L	I	E	V	E
j	0	1	2	3	4	5	6	7	
B	1	0	1	2	3	4	5	6	
E	2	1	0	1	2	3	4	5	
L	3	2	1	0	1	2	3	4	
E	4	3	2	1	1	1	2	3	
I	5	4	3	2	1	2	2	3	
V	6	5	4	3	2	2	2	3	
E	7	6	5	4	3	2	2	2	2

La distancia entre BELIEVE y  
BELEIVE es 2



# Semejanza de texto

## Basadas en secuencia

- Compara dos Strings analizando las sub-secuencias que los componen.
- Parecidas a las basadas en edición.
- Longest common subsequence.  No toma en cuenta si la subsecuencia está interrumpida
- Longest common substring.
- Ratcliff-Obershelp similarity.

### Pros

- Útil para Strings cortos.
- Útil cuando la semántica no es tan importante como la forma de escritura.


### Cons

- Ineficiente para Strings largos.
- Computacionalmente complejo.
- Compara caracteres, con lo que no tiene en cuenta semántica.



# Semejanza de texto

## Basadas en secuencia

- Compara dos Strings analizando las sub-secuencias que los componen.
- Parecidas a las basadas en edición.
- Longest common subsequence.  No toma en cuenta si la subsecuencia está interrumpida
- Longest common substring.
- Ratcliff-Obershelp similarity.

aebcdnlp

taybcrd

### Pros

- Útil para Strings cortos.
- Útil cuando la semántica no es tan importante como la forma de escritura.

### Cons


- Ineficiente para Strings largos.
- Computacionalmente complejo.
- Compara caracteres, con lo que no tiene en cuenta semántica.





# Semejanza de texto

## Basadas en secuencia

- Compara dos Strings analizando las sub-secuencias que los componen.
- Parecidas a las basadas en edición.
- Longest common subsequence.  No toma en cuenta si la subsecuencia está interrumpida
- Longest common substring.
- Ratcliff-Obershelp similarity.

aebcdnlp

taybcd

abcd

sub-secuencia

### Pros

- Útil para Strings cortos.
- Útil cuando la semántica no es tan importante como la forma de escritura.


### Cons

- Ineficiente para Strings largos.
- Computacionalmente complejo.
- Compara caracteres, con lo que no tiene en cuenta semántica.



# Semejanza de texto

## Basadas en secuencia

- Compara dos Strings analizando las sub-secuencias que los componen.
- Parecidas a las basadas en edición.
- Longest common subsequence. 
- Longest common substring.
- Ratcliff-Obershelp similarity.

No toma en cuenta si  
la subsecuencia está  
interrumpida

ae**bc**dnlp

bc

tay**bc**rd

sub-string

### Pros

- Útil para Strings cortos.
- Útil cuando la semántica no es tan importante como la forma de escritura.

### Cons

- Ineficiente para Strings largos.
- Computacionalmente complejo.
- Compara caracteres, con lo que no tiene en cuenta semántica.



# Semejanza de texto

## Basadas en secuencias “simples”

- Compara dos Strings mirando:
  - Prefijo                      → Cantidad de caracteres en común al principio o fin
  - Sufijo.
  - Identity similarity.                      → “equals”
  - Length distance.                      → diferencia en longitud

### Pros

- Simples
- Eficientes.
- Aplicables para string cortos.

### Cons

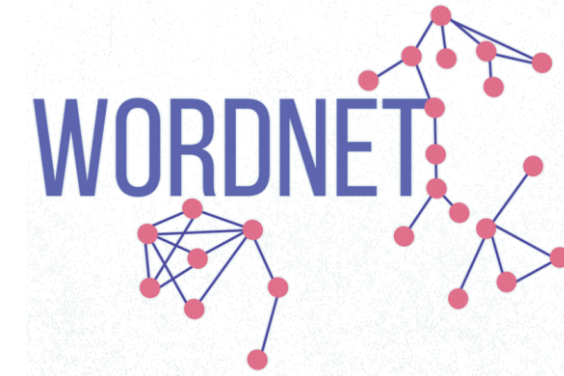
- Ineficiente para Strings largos.
- Primitivo. Aplicable en pocas tareas.
- No tiene en cuenta semántica.



# Semejanza de texto

## Basada en Knowledge

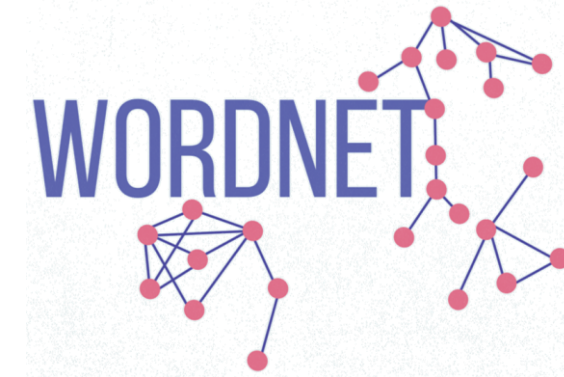
- Cuantifican la semejanza utilizando una red semántica.
  - Por ejemplo, **WordNet**.
- WordNet agrupa los sustantivos, verbos, adjetivos y adverbios en conjuntos de sinónimos cognitivos (los synsets).
  - Cada uno expresa un concepto distinto.
- Los synsets están relacionados en una red compleja de relaciones léxicas.
- Partiendo de un synset, Podemos recorrer WordNet para encontrar aquellos con significado similar.



# Semejanza de texto

## Basada en Knowledge

- Cuantifican la semejanza utilizando una red semántica.
  - Por ejemplo, **WordNet**.
- WordNet agrupa los sustantivos, verbos, adjetivos y adverbios en conjuntos de sinónimos cognitivos (los synsets).
  - Cada uno expresa un concepto distinto.
- Los synsets están relacionados en una red compleja de relaciones léxicas.
- Partiendo de un synset, Podemos recorrer WordNet para encontrar aquellos con significado similar.



Path  
Similarity

Wu  
Palmer

Resnik

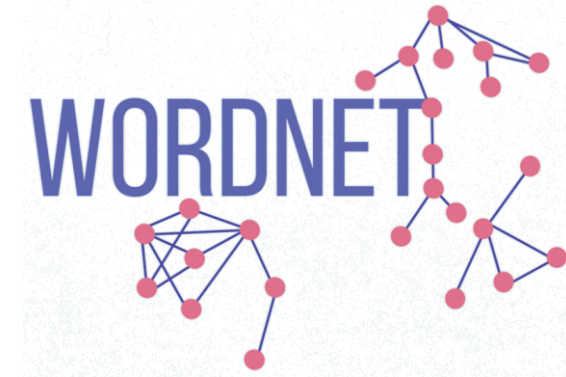
...



# Semejanza de texto

## Basada en Knowledge

- Cuantifican la semejanza utilizando una red semántica.
  - Por ejemplo, **WordNet**.
- WordNet agrupa los sustantivos, verbos, adjetivos y adverbios en conjuntos de sinónimos cognitivos (los synsets).
  - Cada uno expresa un concepto distinto.
- Los synsets están relacionados en una red compleja de relaciones léxicas.
- Partiendo de un synset, Podemos recorrer WordNet para encontrar aquellos con significado similar.



Path  
Similarity

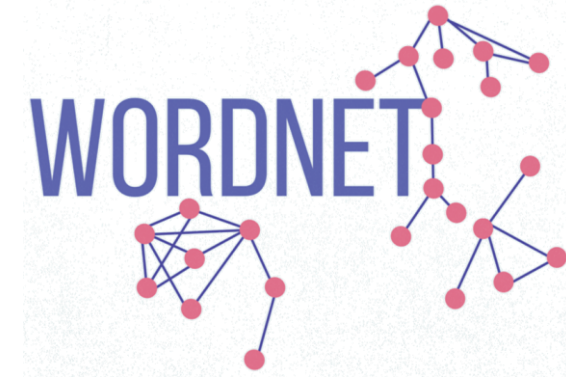




# Semejanza de texto

## Basada en Knowledge

- Cuantifican la semejanza utilizando una red semántica.
  - Por ejemplo, **WordNet**.
- WordNet agrupa los sustantivos, verbos, adjetivos y adverbios en conjuntos de sinónimos cognitivos (los synsets).
  - Cada uno expresa un concepto distinto.
- Los synsets están relacionados en una red compleja de relaciones léxicas.
- Partiendo de un synset, Podemos recorrer WordNet para encontrar aquellos con significado similar.



### Path Similarity

- Asigna un valor entre  $[0,1]$  basado en el camino más corto que conecta dos conceptos en la jerarquía de hiperónimos/hiponóminos.
- -1 en el caso de que no exista el camino.
- 1 si se compara consigo mismo.

**Hiperónimos** describen conceptos más generales. Como un super concepto que engloba al otro.

Por ejemplo, color es un hiperónimo de rojo, azul, verde...

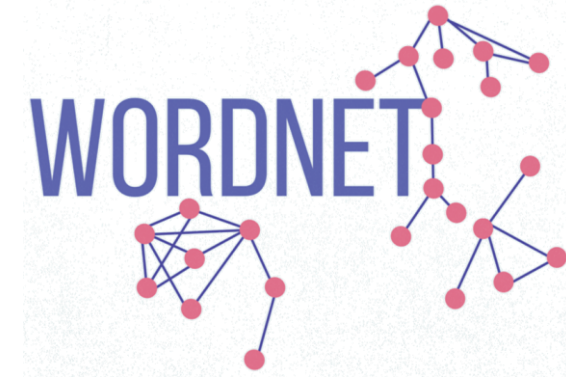




# Semejanza de texto

## Basada en Knowledge

- Cuantifican la semejanza utilizando una red semántica.
  - Por ejemplo, **WordNet**.
- WordNet agrupa los sustantivos, verbos, adjetivos y adverbios en conjuntos de sinónimos cognitivos (los synsets).
  - Cada uno expresa un concepto distinto.
- Los synsets están relacionados en una red compleja de relaciones léxicas.
- Partiendo de un synset, Podemos recorrer WordNet para encontrar aquellos con significado similar.



### Wu Palmer

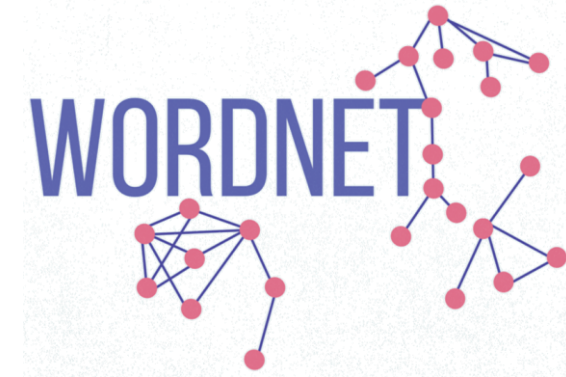
- Mide la semejanza basados en la profundidad de los dos conceptos en la taxonomía y la de su Least Common Subsumer (LCS, el ancestro más específico).
- El LCS no necesariamente coincidirá con el camino más corto entre los dos conceptos.
- Es el ancestro común más profundo en la taxonomía, no el más cercano.
- Cuando multiples LCS existen, se selecciona aquel con el camino corto a la raíz más largo.
- Cuando el LCS tiene multiples caminos a la raíz, se toma el más largo.



# Semejanza de texto

## Basada en Knowledge

- Cuantifican la semejanza utilizando una red semántica.
  - Por ejemplo, **WordNet**.
- WordNet agrupa los sustantivos, verbos, adjetivos y adverbios en conjuntos de sinónimos cognitivos (los synsets).
  - Cada uno expresa un concepto distinto.
- Los synsets están relacionados en una red compleja de relaciones léxicas.
- Partiendo de un synset, Podemos recorrer WordNet para encontrar aquellos con significado similar.



### Resnik

- Estima la semejanza como la probabilidad de encontrar al Least Common Subsumer (LCS) en un corpus grande.
- La probabilidad se conoce como “Information Content” (IC).
- Depende del corpus que fue utilizado para generar el IC.
- NLTK implementa esta y otras variaciones basadas en el IC.



# Semejanza de texto

## Basadas en tokens

- Compara dos Strings analizando los tokens que los componen.
  - “Sube” una unidad de análisis respecto a las anteriores.
  - Utiliza representaciones vectorizadas de los textos.
- Muy utilizadas.

Jaccard  
Tanimoto

Sorensen  
Dice

Overlap

Cosine  
Similarity

Euclidean

Manhattan

...

### Pros

- Eficientes.
- Aplicables para textos largos.
- Depende de la representación base puede incluir semántica o contexto.

### Cons

- Puede no ser útil para palabras individuales u oraciones cortas.
- La magnitud de los vectores no suele ser considerada.



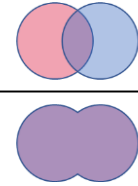
# Semejanza de texto

## Basadas en tokens

### Jaccard Tanimoto

- Se define como la intersección dividida por la union de los dos conjuntos de términos.
- En principio no considera repeticiones.

$$J(t, u) = \frac{|t \cap u|}{|t \cup u|} = \frac{|t \cap u|}{|t| + |u| - |t \cap u|}$$



# Semejanza de texto

## Basadas en tokens

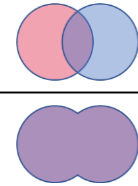
### Jaccard Tanimoto

- Se define como la intersección dividida por la union de los dos conjuntos de términos.
- En principio no considera repeticiones.

What is the best slideshow app among the best Android app?

What are the best app for android?

$$J(t, u) = \frac{|t \cap u|}{|t \cup u|} = \frac{|t \cap u|}{|t| + |u| - |t \cap u|}$$



# Semejanza de texto

## Basadas en tokens

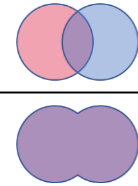
### Jaccard Tanimoto

- Se define como la intersección dividida por la union de los dos conjuntos de términos.
- En principio no considera repeticiones.

What is the **best** **slideshow** **app** among the **best** **Android** **app**?

What are the **best** **app** for **android**?

$$J(t, u) = \frac{|t \cap u|}{|t \cup u|} = \frac{|t \cap u|}{|t| + |u| - |t \cap u|}$$



# Semejanza de texto

## Basadas en tokens

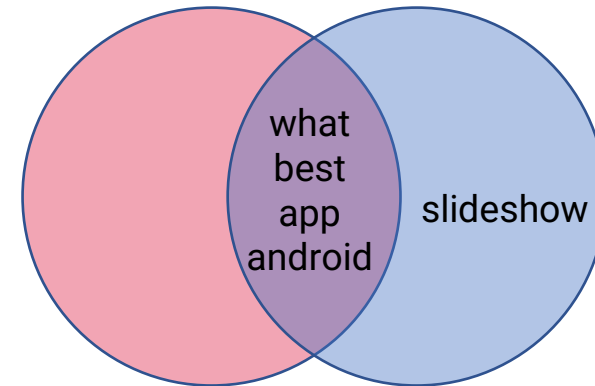
### Jaccard Tanimoto

- Se define como la intersección dividida por la union de los dos conjuntos de términos.
- En principio no considera repeticiones.

$$J(t, u) = \frac{|t \cap u|}{|t \cup u|} = \frac{|t \cap u|}{|t| + |u| - |t \cap u|}$$

What is the **best** **slideshow** **app** among the **best** **Android** **app**?

What are the **best** **app** for **android**?



Jaccard = 0.8





# Semejanza de texto

## Basadas en tokens

### Jaccard Tanimoto

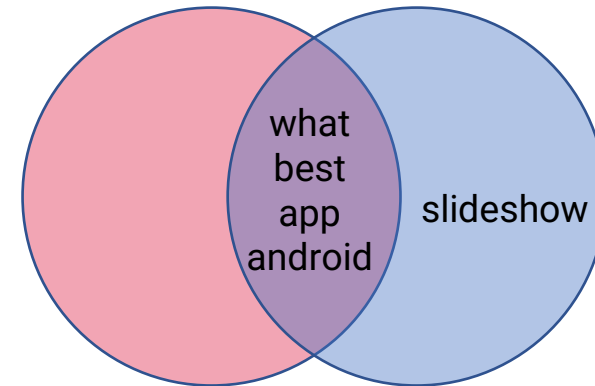
- Se define como la intersección dividida por la union de los dos conjuntos de términos.
- En principio no considera repeticiones.

**What** is the **best** **slideshow** **app** among the **best** **Android** **app**?

**What** are the **best** **app** for **android**?

- Tanimoto tiene una definición similar, pero solo aplica a variables binarias.
- Influenciada por el tamaño de los datos.
  - Datasets grandes pueden incrementar significativamente el tamaño de la union manteniendo un tamaño de intersección similar.

$$J(t, u) = \frac{|t \cap u|}{|t \cup u|} = \frac{|t \cap u|}{|t| + |u| - |t \cap u|}$$



Jaccard = 0.8

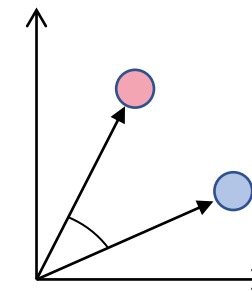


# Semejanza de texto

## Basadas en tokens

### Cosine similarity

- Mide el coseno del ángulo que se forma entre los vectores.
- Vectores con orientaciones similares tendrán un valor cercano a 1.
- Vectores ortogonales tendrán un valor 0.
- Aún cuando la distancia Euclidean sea grande (por diferencias de longitudes), la semejanza puede ser alta (orientación).



$$\text{CosineS}(u, v) = \frac{\sum_{i=1}^n u_i v_i}{\|u\|_2 \|v\|_2}$$

- Una desventaja es que la magnitud de los vectores no es tomada en cuenta, solo la orientación.
- Se utiliza cuando se tienen datos con una gran dimensionalidad y no nos importan las magnitudes.
  - Datos también de distinta longitud.
- En el caso de textos, asume que el hecho de que una palabra aparezca más frecuentemente en un documento que en otro no implica que ese documento se encuentre más relacionado con la palabra.
- Mejor que la distancia Euclidean para datos de gran dimensionalidad.

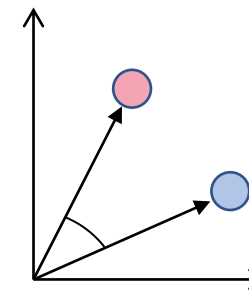


# Semejanza de texto

## Basadas en tokens

### Cosine similarity

- Mide el coseno del ángulo que se forma entre los vectores.
- Vectores con orientaciones similares tendrán un valor cercano a 1.
- Vectores ortogonales tendrán un valor 0.
- Aún cuando la distancia Euclidean sea grande (por diferencias de longitudes), la semejanza puede ser alta (orientación).



$$\text{CosineS}(u, v) = \frac{\sum_{i=1}^n u_i v_i}{\|u\|_2 \|v\|_2}$$

What is the best slideshow app among the best Android app?

What are the best app for android?

$$\text{CosineS} = (1 + 2 + 2 + 1) / \sqrt{(1^2 + 2^2 + 1^2 + 2^2 + 1^2)} \sqrt{(1^2 + 1^2 + 1^2 + 1^2)}$$

$$= 6 / \sqrt{11} * \sqrt{4} = 6 / \sqrt{44} = 0.9$$

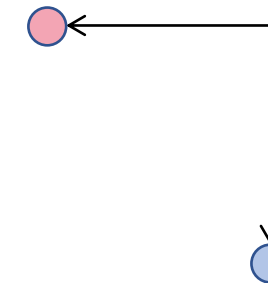


# Semejanza de texto

## Basadas en tokens

### Manhattan

- Se define como la distancia entre dos puntos en una grilla basada estrictamente en caminos verticales u horizontales.
- Resta la diferencia entre cada par de elementos en las diferentes posiciones.
- Similar a Hamming.
  - En principio, asume que son de la misma longitud.
  - Se puede normalizar por la longitud.



$$ManhattanD(u, v) = \|u - v\|_1 = \sum_{i=1}^n |u_i - v_i|$$

- Si bien funciona bien con datos de gran dimensionalidad, es menos intuitiva que la Euclidean.
  - Especialmente con muchas dimensiones.
- Puede resultar en un valor mayor que la Euclidean dado que no considera el camino más corto.
- Cuando hay atributos binarios o discretos funciona bien dado que considera los caminos que se hubieran tenido que tomar entre esos atributos.
  - Euclidean crea una línea recta que no sería realista.

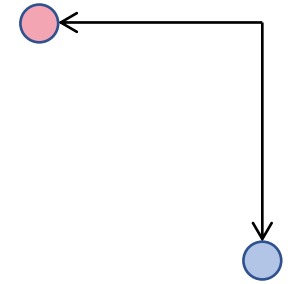


# Semejanza de texto

## Basadas en tokens

### Manhattan

- Se define como la distancia entre dos puntos en una grilla basada estrictamente en caminos verticales u horizontales.
- Resta la diferencia entre cada par de elementos en las diferentes posiciones.
- Similar a Hamming.
  - En principio, asume que son de la misma longitud.
  - Se puede normalizar por la longitud.



$$ManhattanD(u, v) = \|u - v\|_1 = \sum_{i=1}^n |u_i - v_i|$$

La vamos a adaptar para los textos, considerando la diferencia de frecuencia entre los términos.

What is the best slideshow app among the best Android app?

$$ManhattanD = 1 + 1 + 1 = 3$$

What are the best app for android?

{slideshow, app, best}



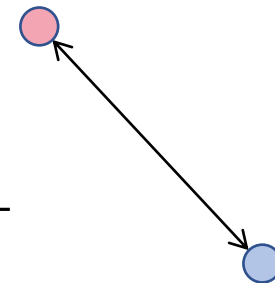
# Semejanza de texto

## Basadas en tokens

### Euclidean

- Muy utilizada.
- Distancia de los vectores en el plano.
- Intuitiva.
- Fácil de implementar.
- Norma 2.

$$EuclideanD(u, v) = \|u - v\|_2 = \sqrt{\sum_{i=1}^n |u_i - v_i|^2}$$



- A pesar de que es comúnmente utilizada, no es invariante a la escala.
  - Requiere normalizer los datos antes de utilizarla.
- A medida que la dimensionalidad de los datos se incrementa, disminuye la utilidad.
  - La geometría de gran dimensionalidad no funciona como intuitivamente se espera del espacio de 2, 3 dimensiones.
- Funcionan bien en bajas dimensiones y la magnitud de los vectores es importantes.
  - Se suele emplear en kNN y HDBSCAN.



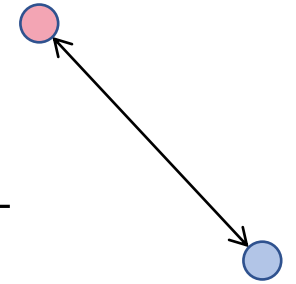
# Semejanza de texto

## Basadas en tokens

### Euclidean

- Muy utilizada.
- Distancia de los vectores en el plano.
- Intuitiva.
- Fácil de implementar.
- Norma 2.

$$EuclideanD(u, v) = \|u - v\|_2 = \sqrt{\sum_{i=1}^n |u_i - v_i|^2}$$



What is the **best slideshow app** among the **best Android app**?

$$EuclideanD = \sqrt{(1^2 + 1^2 + 1^2)} = \sqrt{3}$$

What are the **best app** for **android**?





# Semejanza de texto

Basadas en tokens → Agregando semántica



“el gato se comió al ratón”

“el ratón se comió la comida del gato”



# Semejanza de texto

Basadas en tokens → Agregando semántica



“el **gato** se **comió** al **ratón**”

“el **ratón** se **comió** la comida del **gato**”

Si consideramos las semejanzas que vimos recién, son similares...

Jaccard = 0.75



# Semejanza de texto

Basadas en tokens → Agregando semántica



“el **gato** se **comió** al **ratón**”

“el **ratón** se **comió** la comida del **gato**”

Si consideramos las semejanzas que vimos recién, son similares...

Jaccard = 0.75

- Estas oraciones no tienen el mismo significado!

“el **gato** se **comió** al **ratón**”

“el **ratón** se **comió** la comida del **gato**”



# Semejanza de texto

Basadas en tokens → Agregando semántica



“el **gato** se **comió** al **ratón**”

“el **ratón** se **comió** la comida del **gato**”

Si consideramos las semejanzas que vimos recién, son similares...

Jaccard = 0.75

- Estas oraciones no tienen el mismo significado!

“el **gato** se **comió** al **ratón**”



sujeto



verbo



objeto

“el **ratón** se **comió** la comida del **gato**”



sujeto



verbo



objeto



# Semejanza de texto

Basadas en tokens → Agregando semántica

“El presidente saluda a la prensa  
en Chicago”

“Obama habla con los medios en Illinois”

Si consideramos las semejanzas que vimos recién, NO son similares...

Jaccard = 0



# Semejanza de texto

Basadas en tokens → Agregando semántica

“El presidente saluda a la prensa  
en Chicago”

“Obama habla con los medios en Illinois”

Si consideramos las semejanzas que vimos recién, NO son similares...

Jaccard = 0

- No comparten términos!
- Pero, tienen un significado similar



# Semejanza de texto

Basadas en tokens → Agregando semántica

“el **gato** se **comió** al **ratón**”

“el **ratón** se **comió** la comida del **gato**”

“El presidente saluda a la  
prensa en Chicago”

“Obama habla con los medios en Illinois”

- No considera el contexto.
- No considera el significado de las palabras en la oración.
- Podemos cambiar la representación!

Representación  
tradicional



Representación  
embeddings





# Semejanza de texto

## Agregando semántica

### Word Mover's

- Utiliza los embeddings de los términos en los dos textos para medir la mínima distancia que los términos en un texto tienen que “moverse” en el espacio semántico para llegar a los términos en el otro texto.
- Problema de transporte → Programación lineal!
  - Hay optimizaciones.
- No tiene parámetros.
- Es interpretable.



# Semejanza de texto

## Agregando semántica

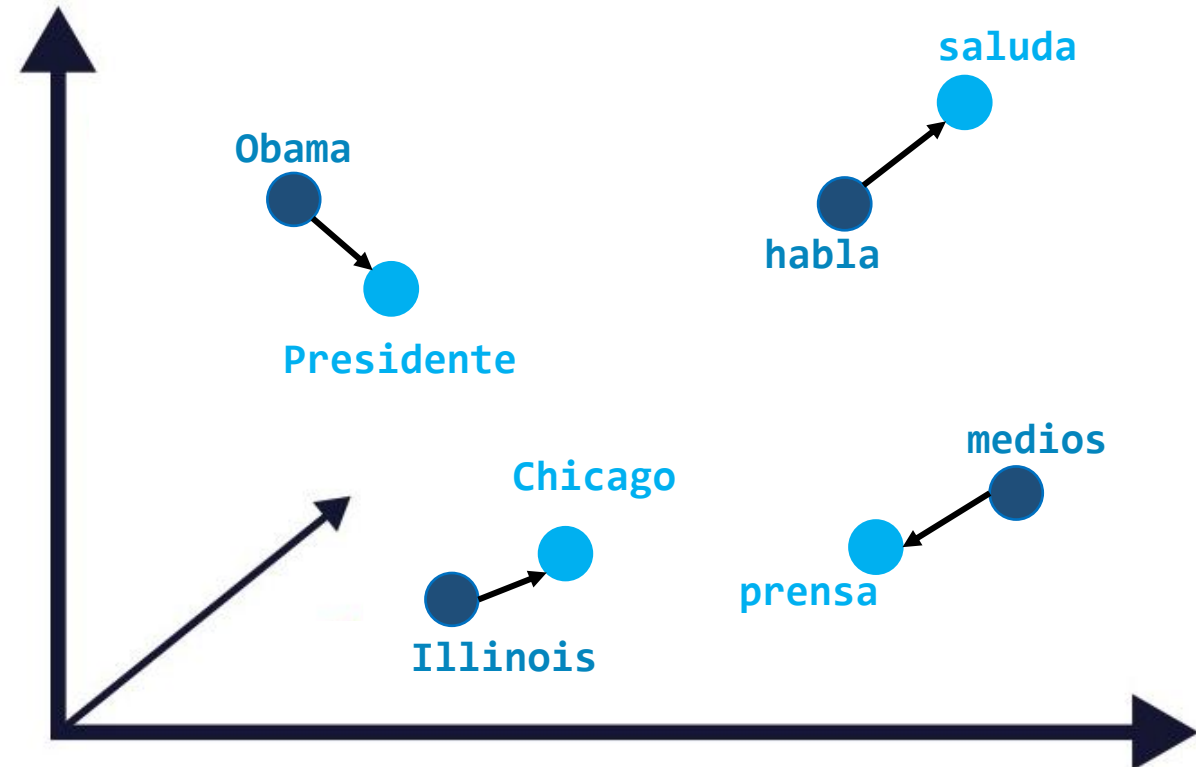
### Word Mover's

- Utiliza los embeddings de los términos en los dos textos para medir la mínima distancia que los términos en un texto tienen que “moverse” en el espacio semántico para llegar a los términos en el otro texto.
- Problema de transporte → Programación lineal!
  - Hay optimizaciones.

- No tiene parámetros.
- Es interpretable.

“El **presidente** **saluda** a la **prensa** en **Chicago**”

“**Obama** **habla** con los **medios** en **Illinois**”



# Semejanza de texto

## Comparación

Edit  
Secuencia



Tokens



Knowledge  
Semánticas

- No se adaptan a la longitud.
- No consideran sinónimos.
- No consideran errores de spelling.
- Dan a todas las palabras la misma importancia.
- No se adaptan a frecuencias.

- No consideran sinónimos.
- No consideran errores de spelling.
- Dan a todas las palabras la misma importancia.

- No consideran errores de spelling.
- Pueden haberse generado de forma no supervisada.



## Text Representation

### Cómo representar el texto de forma que las técnicas y modelos de machine learning puedan comprenderlos?

#### Modelos Tradicionales

- De basan en la parte léxica, es decir, en las palabras que componen el texto, su frecuencia y, ocasionalmente, algunas de las palabras inmediatamente a su alrededor.
- Al considerar aspectos solo léxicos, pierden aspectos relacionados al orden y secuencia.

#### Modelos de Deep Learning

- Transforman el texto original en una representación semántica de la palabra y su contexto.
- Capturan el significado, relaciones semánticas y los diferentes contextos en los que las palabras (y sus diferentes acepciones) son utilizadas.

Edit  
Secuencia



Tokens



Knowledge  
Semánticas

- No se adaptan a la longitud.
- No consideran sinónimos.
- No consideran errores de spelling.
- Dan a todas las palabras la misma importancia.
- No se adaptan a frecuencias.

- No consideran sinónimos.
- No consideran errores de spelling.
- Dan a todas las palabras la misma importancia.

- No consideran errores de spelling.



# Procesamiento de Lenguaje Natural

---

## Representación de Texto