Aplicaciones de Word Vectors en el PLN

Gabriel Mordecki

Evaluación de vectores de palabras: Métodos implícitos y explícitos

Métodos implícitos de evaluación

- Se utilizan los vectores en alguna tarea de PLN.
- Se comparan los resultados obtenidos con cada set de vectores.
- Se consideran los vectores como un parámetro más del algoritmo.
- Muy costoso computacionalmente

Métodos explícitos de evaluación

- Se utilizan los vectores en tareas diseñadas específicamente para su evaluación.
  - Similitud entre palabras: comparación con puntaje por humanos.
  - Analogías sintácticas: recuperar *jugando* a partir del vector *jugar* y la relación entre *corriendo* y *correr*.
  - Analogías semánticas: recuperar *Obama* a partir del vector *Estados_Unidos* y la relación entre *Putin* y *Rusia*.
- Eficientes, precisos y comparables.

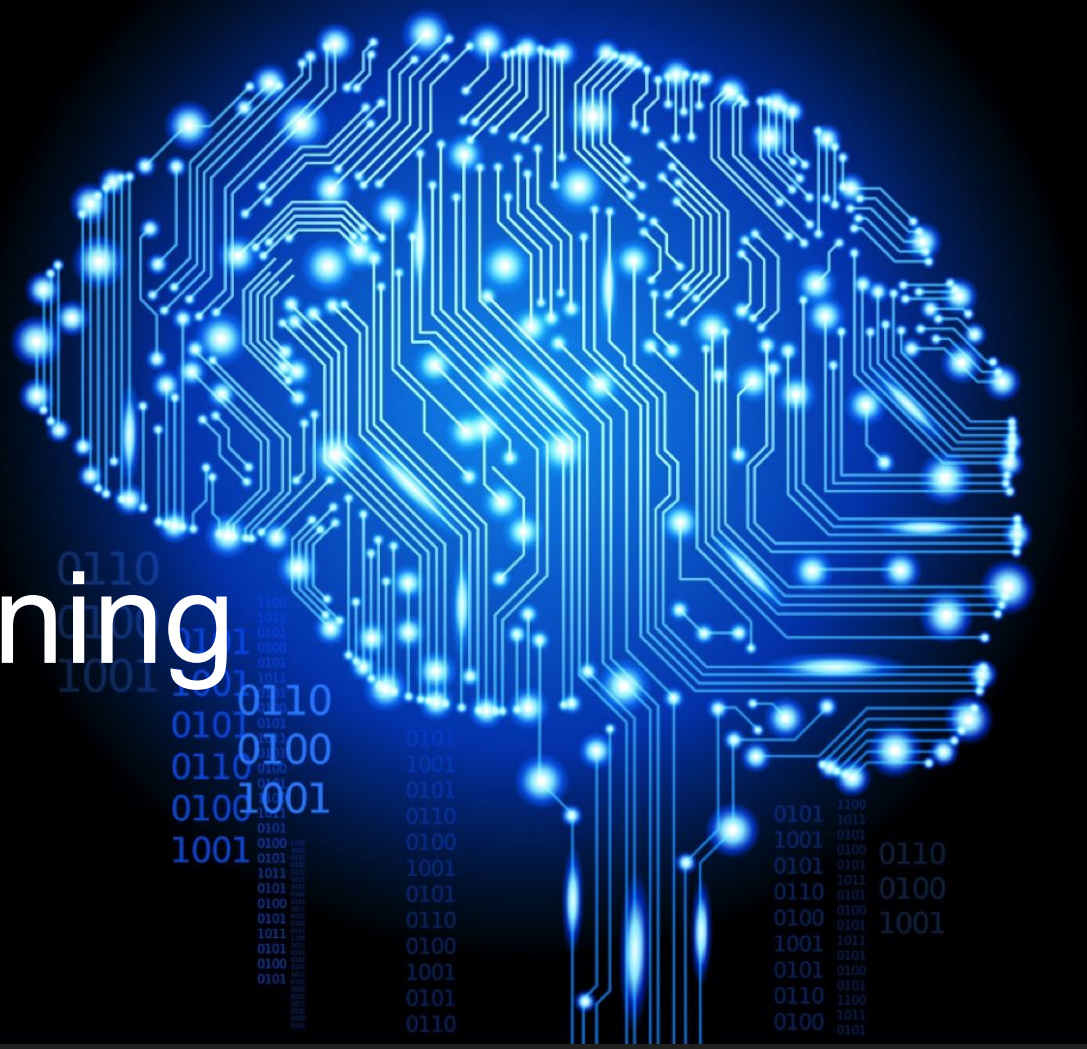
Linguistic regularities in continuous space word representations
Tomas Mikolov, Wen-tau Yih y Geoffrey Zweig

Sin embargo...

NO está clara la relación entre modelos explícitos e implícitos.


Mitos y verdades:
4 mitos sobre
Word Vectors

1: Word2vec es Deep Learning



Word2Vec como regresión logística

- Skip-gram y CBOW son pensados, diseñados y explicados como redes neuronales.
- Tienen el objetivo primordial de simplificar la red neuronal para mejorar los tiempos de ejecución y poder incluir más vocabulario.
- Objetivo para cada ejemplo en Skip-gram, maximizar

$$J_{\text{NEG}} = \log Q_\theta(D=1|w_t,h) + k \mathop{\mathbb{E}}_{\tilde{w}\sim P_{\text{noise}}} [\log Q_\theta(D=0|\tilde{w},h)]$$

  Donde Q_θ es una regresión logística binaria
- Si dejamos los contextos fijos, aprender los vectores se reduce a una regresión logística. Si dejamos los vectores fijos, al aprender los contextos sucede lo mismo.

word2vec Explained: Deriving Mikolov et al.'s Negative-Sampling Word-Embedding Method, by Yoav Goldberg and Omer Levy

Tensor Flow: Vector Representations of Words

Skip-Gram como factorización implícita de matriz PMI

- El método Skip-Gram con Negative Sampling puede verse como una factorización implícita de una matriz PMI desplazada por una constante, usando descenso por gradiente estocástico.
- W es la matriz de word vectors, C es la de contextos (usualmente se descarta).
- El objetivo de la red es aproximar en $W \cdot C' = M$ la matriz PMI - log(k). Siendo k el número de negative samples.
- PMI se usa desde los 90 como medida de asociación en PLN.
- Levy y Goldberg presentan métodos que aproximan mejor la matriz, pero funcionan peor que SGNS.

Neural Word Embedding as Implicit Matrix Factorization, Omer Levy and Yoav Goldberg


2: Los métodos estadísticos son mejores que los de conteo

Métodos estadísticos y de conteo tienen resultados similares

- Los resultados presentados en word2vec superaban significativamente el estado del arte.
- Además del método en sí, se presentaron optimizaciones que pueden trasladarse a lo métodos de conteo (como PMI con SVD).
- Estos hiperparámetros de optimización mejoran los resultados y hacen que ambos métodos se comporten de manera similar.
- Muchas veces, cambiar un hiperparámetro es mejor que cambiar de algoritmo o agrandar corpus.

Improving Distributional Similarity with Lessons Learned from Word Embeddings
Omer Levy, Yoav Goldberg y Ido Dagan

Métodos estadísticos y de conteo tienen resultados similares

- Hiperparámetros de pre-procesamiento:
  - Contexto de ventana dinámico: le da más peso a las palabras más cercanas en la ventana.
  - Subsampling: sacar palabras muy comunes. Al hacerlo antes de procesar el corpus agranda el tamaño de las ventanas.
  - Eliminar palabras raras

Improving Distributional Similarity with Lessons Learned from Word Embeddings
Omer Levy, Yoav Goldberg y Ido Dagan

Métodos estadísticos y de conteo tienen resultados similares

- Hiperparámetros de métricas de asociación:
  - Desplazamiento de la PMI: el k de cantidad de negative sampling
  - Suavizado de la probabilidad de negative sampling: al calcular la distribución de los contextos se eleva a la ¾ el conteo de apariciones.

    Esto disminuye el sesgo de PMI sobre las palabras raras.

Improving Distributional Similarity with Lessons Learned from Word Embeddings
Omer Levy, Yoav Goldberg y Ido Dagan

Métodos estadísticos y de conteo tienen resultados similares

- Hiperparámetros de post-procesamiento:
  - Sumar los contextos: tomar como vector de representación V = W + C.
  - Ponderación de vectores propios: SGNS factoriza la matriz en dos matrices sin sesgos, al contrario de SVD (tres matrices sesgadas).
  - Normalización de los vectores.

Improving Distributional Similarity with Lessons Learned from Word Embeddings
Omer Levy, Yoav Goldberg y Ido Dagan

Métodos estadísticos y de conteo tienen resultados similares

- Rule of thumb:
  - Siempre usar suavizado de la probabilidad de negative sampling.
  - No usar SVD “normal”, usar variantes con ponderación de vectores propios.
  - SGNS es una buena baseline: nunca es mucho peor que los otros métodos, es más rápido de entrenar y consume mucho menos disco y memoria.
  - En SGNS, usar “bastantes” negative samples.
  - Probar la suma de contextos V = W + C. No requiere más entrenamiento y a veces mejora resultados.

Improving Distributional Similarity with Lessons Learned from Word Embeddings
Omer Levy, Yoav Goldberg y Ido Dagan

3: Los vectores se usan con Deep Learning

No sólo

- Los vectores de palabras no sólo se pueden usar con redes neuronales
- También se pueden usar con otras cosas.


4: Creo
vectores,
me sirven
SIEMPRE

Los vectores dependen de su contexto

- Los word vectores describen las palabras según el contexto en el que éstas ocurren en un determinado corpus de texto.
- Por lo tanto, sus representaciones dependen del contexto y del corpus.
- También puede variar según el objetivo: si quiero hacer POS tagging, “bueno” y “malo” van a ser similares. Si quiero hacer análisis de sentimiento no me sirve.
- Sin embargo, a veces sí pueden funcionar en forma genérica.

Los vectores dependen de su contexto

- Resultados de palabras más similares entrenando con noticias vs foros argentinos

In: most_similar('puto')

[('jodido', 0.85119563),
 ('mierda', 0.79496604),
 ('cabron', 0.79337001),
 ('maldito', 0.7830928),
 ('joder', 0.77866089),
 ('puta', 0.76814502),
 ('gilipollas', 0.76421535),
 ('punetero', 0.75728369),
 ('estupido', 0.73833019),
 ('imbecil', 0.72823781)]

In: most_similar('puto')

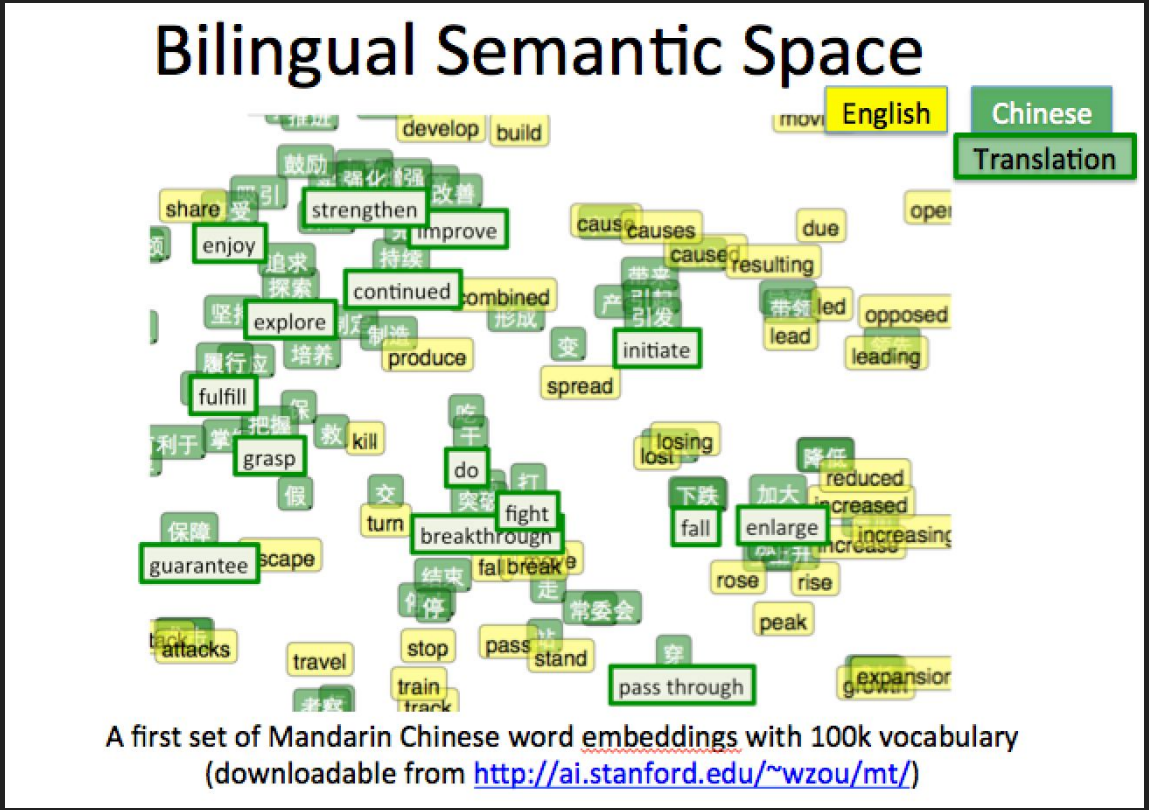
[('maricon', 0.80574465),
 ('pelotudo', 0.74482912),
 ('culoroto', 0.74205154),
 ('forro', 0.73786169),
 ('pitocorto', 0.73569256),
 ('malcogido', 0.73472273),
 ('mandiyuta', 0.73357219),
 ('repelotudo', 0.73124564),
 ('petero', 0.73021728),
 ('cejudo', 0.72725046)]

Vectores de palabras bi-lenguaje

- Crean vectores entrenados con inglés y chino al mismo tiempo
- Inicializan las matrices V con vectores de palabras y las A usando cuentas de alineación de traducciones automáticas.
- $J_{\text{TEO-en}\rightarrow\text{zh}} = || V_{zh} - A_{\text{en}\rightarrow\text{zh}} V_{en}||^2$
- Obtienen resultados que llegan al estado del arte en traducción

Bilingual Word Embeddings for Phrase-Based Machine Translation
Will Y. Zou, Richard Socher, Daniel Cer and Christopher D. Manning

Vectores de palabras bi-lenguaje



Bilingual Word Embeddings for Phrase-Based Machine Translation
Will Y. Zou, Richard Socher, Daniel Cer and Christopher D. Manning

Vectores de palabras multi-lenguaje

- Tres grandes métodos, difieren en el nivel de asunción de alineaciones:
  - A nivel de palabra (diccionarios)
  - A nivel de oración (documentos legales, textos religiosos)
  - A nivel de documento (noticias, wikipedia)
- Dos tipos de features:
  - Palabras en ambos idiomas

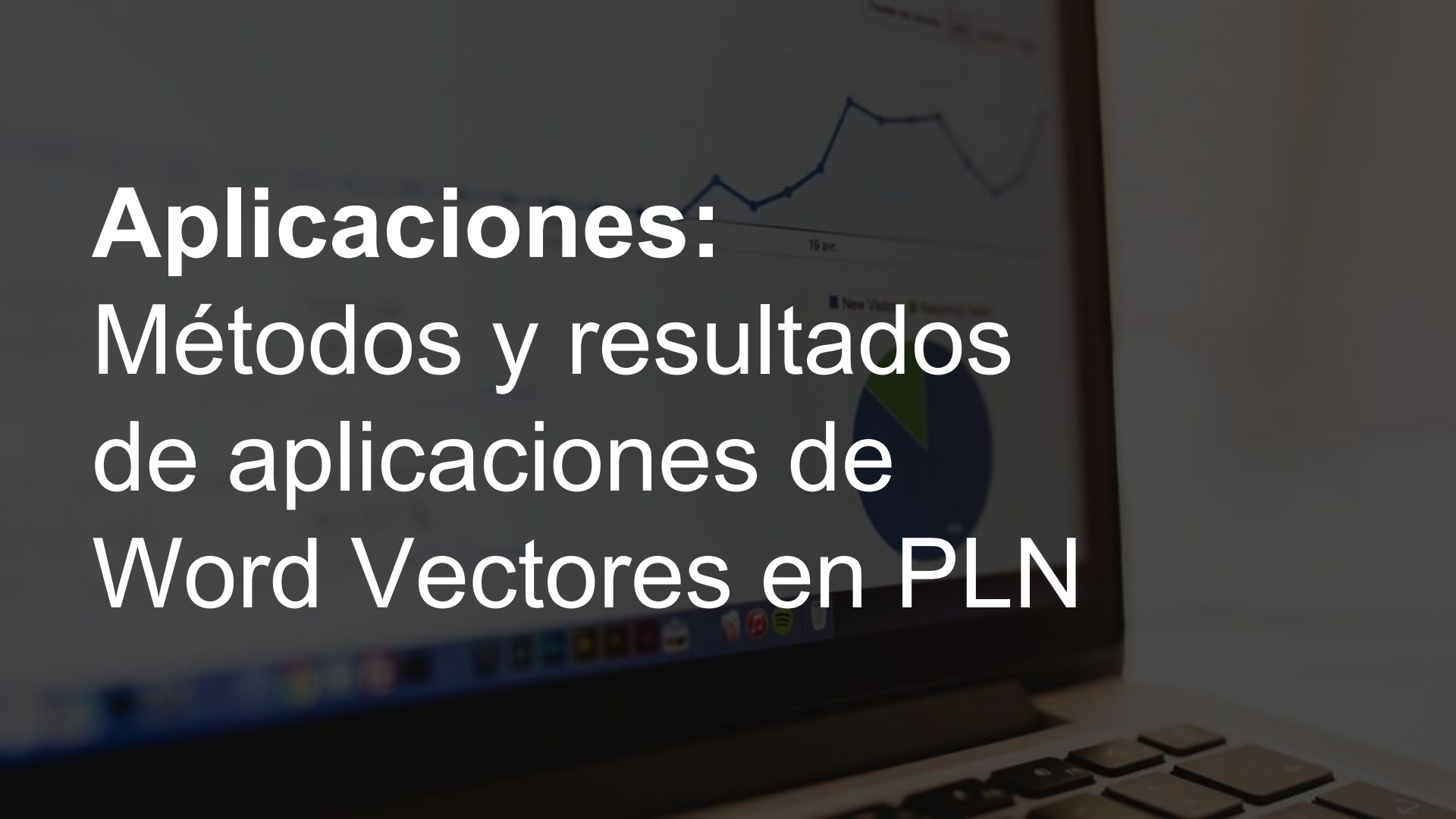
    BilBOWA: SGNS pero agrega contexto en otros lenguajes.
  - ID de oración: la palabra es representada como las oraciones en las que está.

Reconsidering Cross-lingual Word Embeddings
Omer Levy, Anders Søgaard, Yoav Goldberg (¡18 Agosto 2016!)

Vectores de palabras multi-lenguaje

- Los métodos no llegan al estado del arte.
- Sin embargo, son más eficientes.
- Hipotetizan que los métodos tradicionales tiene más optimizaciones que los nuevos no, y eso marca la diferencia.

Reconsidering Cross-lingual Word Embeddings
Omer Levy, Anders Søgaard, Yoav Goldberg (¡18 Agosto 2016!)

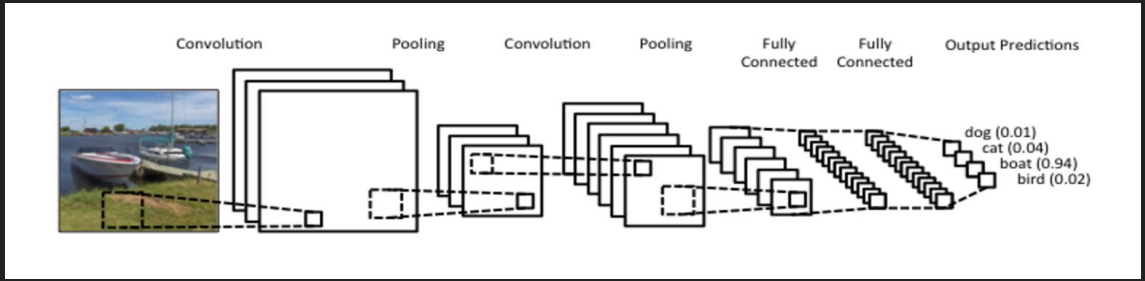


Aplicaciones: Métodos y resultados de aplicaciones de Word Vectores en PLN

(

Redes neuronales convolucionales(CNN)

- Convolución: una función que se aplica a una ventana deslizante de una matriz
- Las CNN son redes neuronales cuyas capas son distintas convoluciones con funciones de activación no-lineares (TahH, ReLU)
- Así capturan datos locales (lower features)
- Se aprenden en el entrenamiento los valores de las convoluciones



WILDML: Understanding CNN for NLP (Denny Britz)

Redes neuronales convolucionales(CNN)

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

	0	1	0	
	1	-4	1	
	0	1	0	

[WILDML: Understanding CNN for NLP (Denny Britz)]

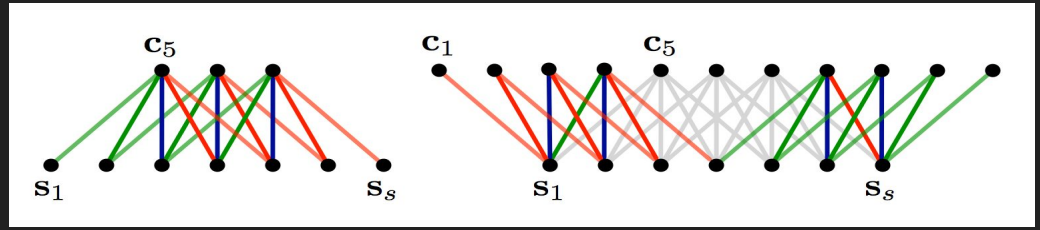
Redes neuronales convolucionales(CNN)

- En PLN, representamos oraciones como matrices donde cada fila es una palabra (y las palabras están ordenadas).
- Tamaño variable en largo.
- Las filas son la representación vectorial de la palabra, y puede tener otras cosas.
- El ancho de la convolución es el mismo que el de la matriz, distinto a imágenes.
- “All models are wrong, but some are useful”
- Son rápidas

WILDML: Understanding CNN for NLP (Denny Britz)

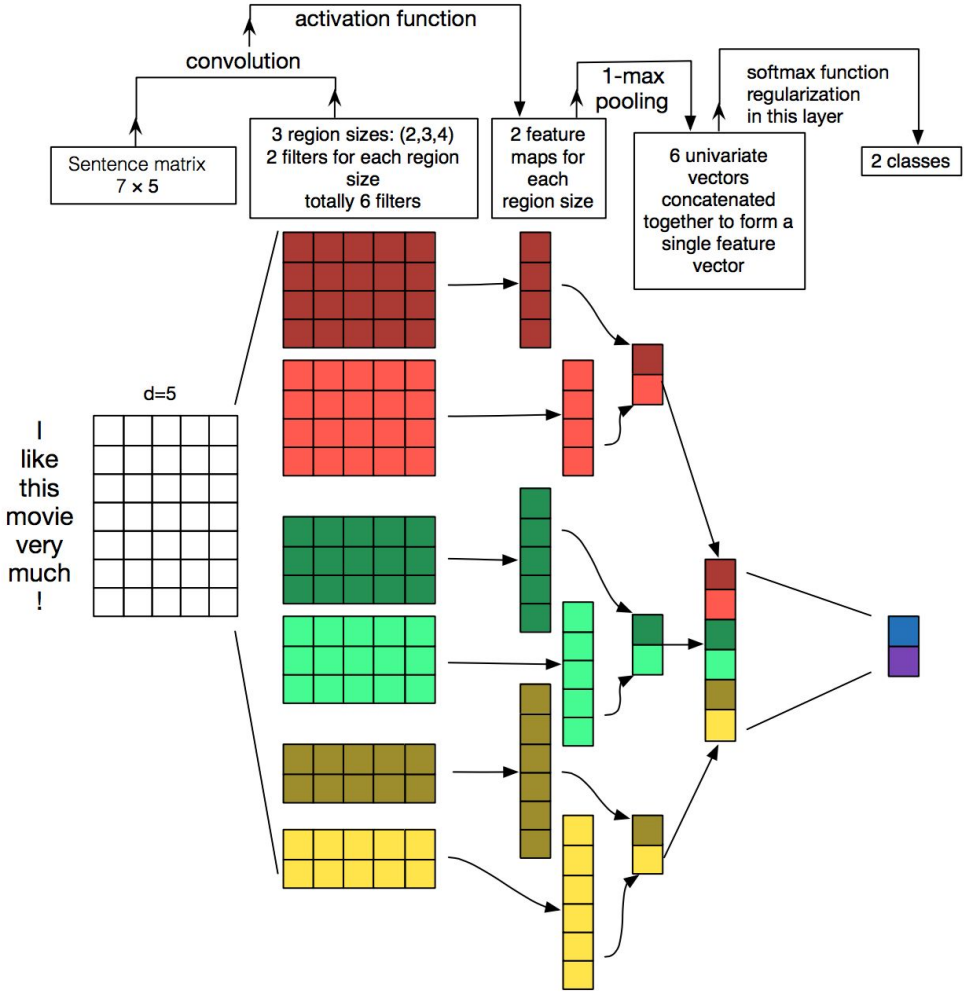
Redes neuronales convolucionales(CNN)

- Narrow vs Wide: cómo manejar los bordes, poner padding u obviarlos.



- Tamaño de desplazamiento (stride size): cuánto muevo la ventana de la convolución. Hasta ahora asumimos 1.
- Pooling: subsamplear después de la convolución.
  - Usualmente se usa el máximo.
  - Normaliza y disminuye la cantidad de salidas
  - Captura información global (hay negación)

WILDML: Understanding CNN for NLP (Denny Britz)


activation function
convolution
1-max pooling
softmax function regularization in this layer
Sentence matrix 7 × 5
3 region sizes: (2,3,4) 2 filters for each region size totally 6 filters
2 feature maps for each region size
6 univariate vectors concatenated together to form a single feature vector
2 classes
d=5
I
like
this
movie
very
much
!

)

Vectores como parte de arquitectura común multi-tarea

- Crear una arquitectura que entrene y resuelva a la vezmuchos problemas de PLN: POS, NER, chunking, Semantic Role Labeling, Semantically Related Words, modelos de lenguaje.
- Todo se basa en vectores de palabras con redes convolucionales.
- Todas las tareas comparten los vectores y tienen además sus propias capas de redes neuronales.
- Agregan la feature “la primera letra es mayúscula”
- Entrenan primero el modelo de lenguaje y ahí generan los vectores.
- Luego, los vectores se usan (y se actualizan) con todas las tareas a las vez.

A unified architecture for natural language processing: deep neural networks with multitask learning
Ronan Collobert and Jason Weston (2008)

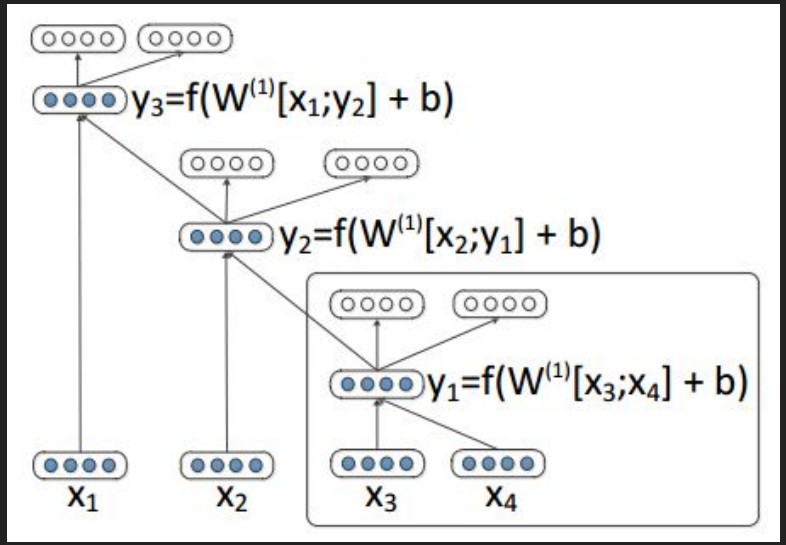
Vectores como parte de arquitectura común multi-tarea

- Proceso:
  - 1. Select the next task.
  - 2. Select a random training example for this task.
  - 3. Update the NN for this task by taking a gradient step with respect to this example.
  - 4. Go to 1.
- Modelo de lenguaje: problema de clasificación, dado una palabra y un contexto, ¿la palabra puede ir ahí, está relacionada con su contexto? Se entrena con Wikipedia (era 2008).
- Obtienen vectores con buenas propiedades y mejoran (en el caso de SRL mucho) el estado del arte.
  - ¡Sin usar features sintácticas!

A unified architecture for natural language processing: deep neural networks with multitask learning
Ronan Collobert and Jason Weston (2008)

Autoencoders recursivos para análisis de sentimiento

- Hacen análisis de sentimiento mediante el encoding en vectores de oraciones.
- Primero entrenan los vectores (en otra versión inician aleatorio)
- Luego, generan vectores para pares de palabras:



Semi-Supervised Recursive Autoencoders for Predicting Sentiment Distributions (2011)
Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, Christopher D. Manning

Autoencoders recursivos para análisis de sentimiento

- Importa el orden en que se arma el árbol (no necesariamente es algo sintáctico).
- Usan un algoritmo greedy que elige como los dos primeros los que tengan menos error, luego sigue con el segundo nivel hasta el final.
- El error de reconstrucción es ponderado por la cantidad de palabras que representa cada vector (de igual tamaño):

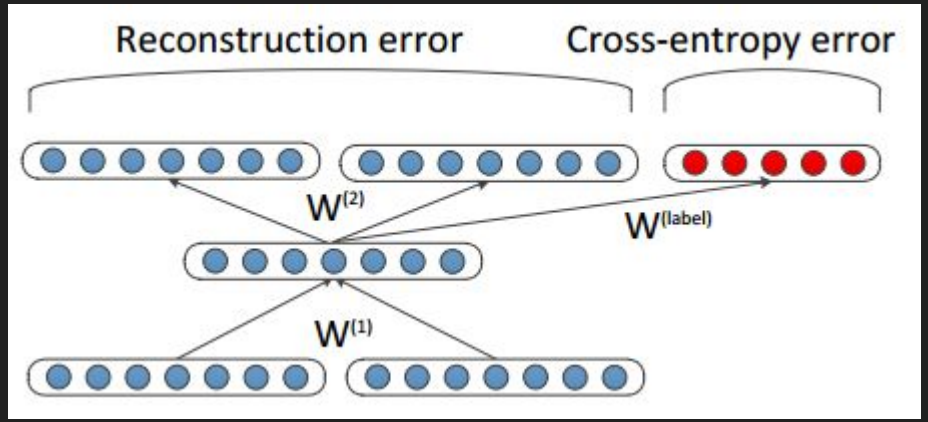
$$E_{rec}([c_1; c_2]; \theta) = \frac{n_1}{n_1 + n_2} \left|\left|c_1 - c_1'\right|\right|^2 + \frac{n_2}{n_1 + n_2} \left|\left|c_2 - c_2'\right|\right|^2$$

(n_1 es la cantidad de palabras de c_1)

Semi-Supervised Recursive Autoencoders for Predicting Sentiment Distributions (2011)
Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, Christopher D. Manning

Autoencoders recursivos para análisis de sentimiento

- No sólo se toma en cuenta el error de reconstrucción, sino de la etiqueta de sentimiento



$$E([c_1; c_2]_s, p_s, t, \theta) =$$

$$\alpha E_{rec}([c_1; c_2]_s; \theta) + (1 - \alpha) E_{cE}(p_s, t; \theta).$$

Alpha permite ponderar reconstrucción/etiqueta.

Apha = 0.2 es lo que les da mejores resultados

Semi-Supervised Recursive Autoencoders for Predicting Sentiment Distributions (2011)
Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, Christopher D. Manning

Autoencoders recursivos para análisis de sentimiento

- Obtienen excelentes resultados.
- Prueban con un test de secretos etiquetados con 5 clases. En una versión eligen una de las 5 (50% accuracy) y en otras generan una distribución multinomial entre ellas (0.7 Average KL-divergence)
- También superan el estado del arte en clasificación de polaridad en reviews de películas (77.5% acc) y opiniones (86.4% acc)

Semi-Supervised Recursive Autoencoders for Predicting Sentiment Distributions (2011)
Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, Christopher D. Manning

Doc2Vec Oraciones o párrafos a vectores

- Muy parecido a Word2Vec, pero utiliza además información de oraciones o párrafos.
- Proponen dos modelos
  - Distributed Memory Model of Paragraph Vectors (PV-DM).
  - Distributed Bag of Words version of Paragraph Vector (PV-DBOW)

Distributed Representations of Sentences and Documents
Quoc V. Le y Tomas Mikolov (2014)

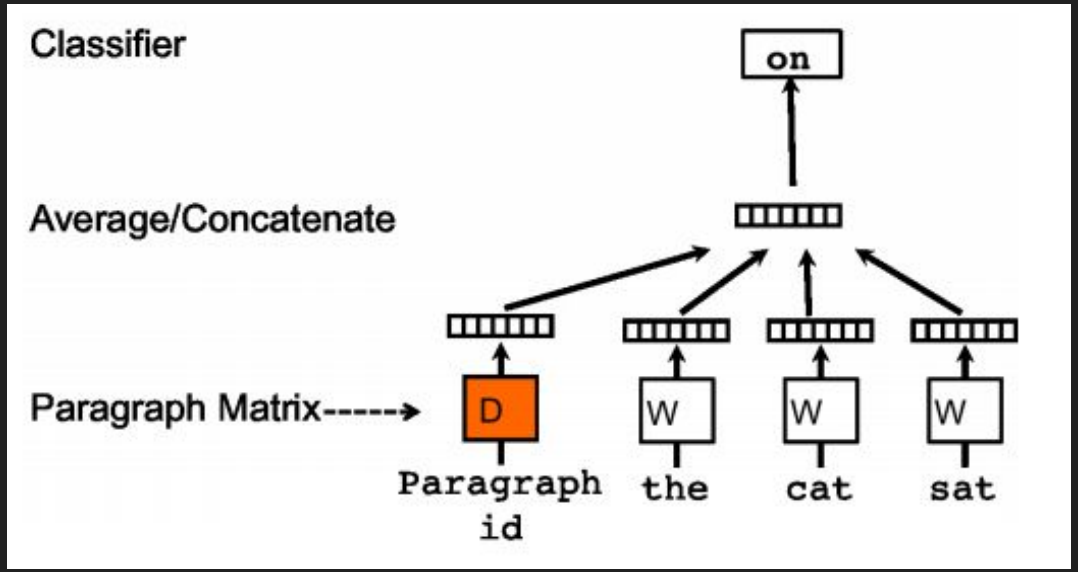
Doc2Vec Oraciones o párrafos a vectores

- Distributed Memory Model of Paragraph Vectors (PV-DM).
  - Cada párrafo se matchea con un ID único
  - Se intenta predecir la próxima palabra con una concatenación o promedio de las n anteriores más el vector correspondiente al párrafo.
  - El vector del párrafo “guarda” el contexto restante.
  - Los vectores de palabras se comparten para todos los párrafos
  - Para predecir, se computa el vector de párrafo con SGD. Los otros parámetros son fijos
  - Los vectores se usan con NN, SVM, LR, etc

Distributed Representations of Sentences and Documents
Quoc V. Le y Tomas Mikolov (2014)

Doc2Vec
Oraciones o párrafos a vectores

- Distributed Memory Model of Paragraph Vectors (PV-DM).



Distributed Representations of Sentences and Documents
Quoc V. Le y Tomas Mikolov (2014)

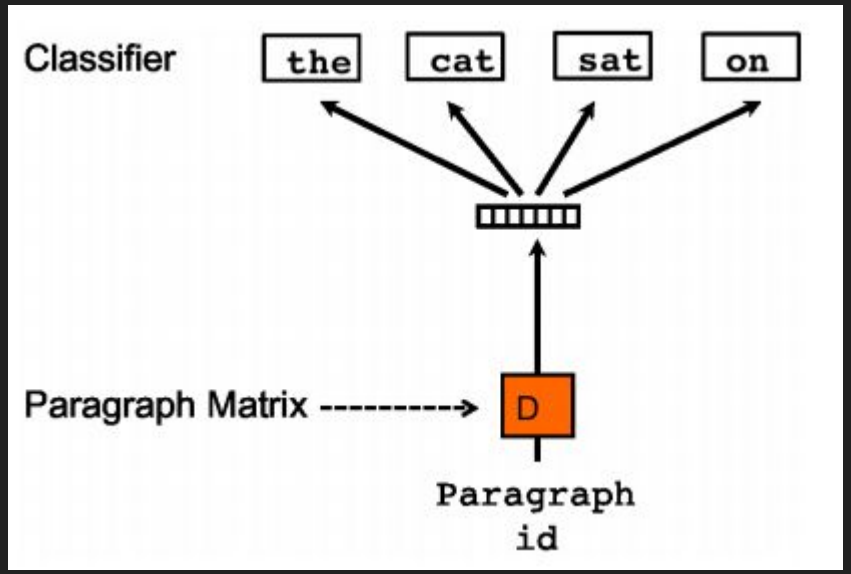
Doc2Vec Oraciones o párrafos a vectores

- Distributed Bag of Words version of Paragraph Vector (PV-DBOW)
  - Pese a su nombre, parecido a Skip-Gram
  - Intenta predecir las palabras de un párrafo dado su vector
  - Modelo más simple

Distributed Representations of Sentences and Documents
Quoc V. Le y Tomas Mikolov (2014)

Doc2Vec Oraciones o párrafos a vectores

- Distributed Bag of Words version of Paragraph Vector (PV-DBOW)



Distributed Representations of Sentences and Documents
Quoc V. Le y Tomas Mikolov (2014)

Doc2Vec Oraciones o párrafos a vectores

- En sus pruebas, sugieren representar el párrafo como la concatenación de los vectores generados por las dos técnicas.
- Hacen análisis de sentimiento y recuperación de información con excelentes resultados, superando el estado del arte.
- El análisis de sentimiento es con oraciones cortas y también con párrafos (dataset de IMDB y Stanford Sentiment Treebank)
- Sugieren hacer cross-validation con el tamaño de la ventana entre 5 y 12. Puntuación se toma como palabras normales. Usan padding “NULL”
- Es un método costoso computacionalmente.
- Está implementado en Gensim.

Distributed Representations of Sentences and Documents
Quoc V. Le y Tomas Mikolov (2014)

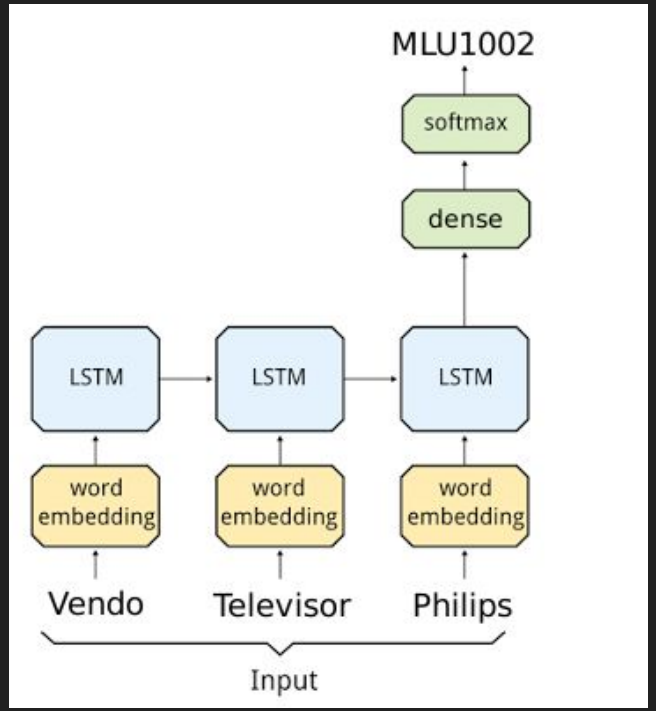
Etiquetado de polaridad no supervisado

- Formar un lexicón de palabras positivas y negativas de forma no supervisada.
- El sentimiento es específico al dominio.
- Preprocesamiento: lematización, stopwords, detección de términos multi-palabra (Log-Likelihood Ratio).
- Entrenan word2vec y calculan distancia a un set inicial de palabras positivas y negativas.
- Pese a datasets muy chicos (100k inglés, 4k español), consiguen buenos resultados:
  - Películas: “*repetitivo*” como negativo
  - Laptops: “*fast*” y “*trendy*” positivo.

Unsupervised Word Polarity Tagging by Exploiting Continuous Word Representations
Aitor García Pablos, Montse Cuadros, German Rigau

Clasificación de tópico

- Clasificar categoría de Mercado Libre según el título del producto
- Entrenado con títulos y descripciones de Mercado Libre (cientos de millones de palabras)
- Mejor resultado que bow
- El orden de entrada afecta el resultado



Vectores de palabras en la Fing

Nabu

DASHBOARD | CORPUS | EMBEDDINGS | TESTS

The corpus currently has 5,989,632,066 words!

elpais.com
Size: 809,067,269

univision.com
Size: 679,100,427

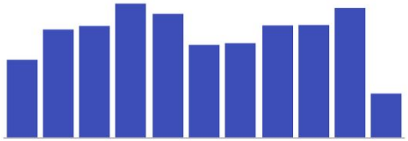
lanacion.com.ar
Size: 652,727,505

es.wikipedia.com
Size: 416,932,056

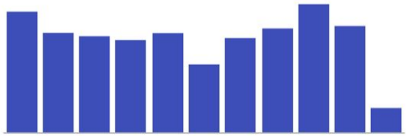
abc.com.py
Size: 415,044,172

euromatrixplus.net/multi-un
Size: 313,845,665

informador.com.mx
Size: 257,206,983



infobae.com
Size: 244,158,555



Nabu

DASHBOARD | CORPUS | EMBEDDINGS | TESTS

Search: Universidad de la República

Filter source: 180.com.uy, elpais.com.uy

Use advanced query

SEARCH

Results

Date Scraped	Snippet	Source
04/03/2016	50% Bonificación en el precio del boleto del transporte urbano para los estudiantes de la *Universidad* de la *República*.	elpais.com.uy
04/03/2016	Homenaje a Barrán La *Universidad* de la *República* realizará un homenaje al profesor José Pedro	elpais.com.uy
04/03/2016	, con el título "El sistema terciario y la *Universidad* de la *República*: los desafíos para los próximos	elpais.com.uy

Nabu

DASHBOARD | CORPUS | EMBEDDINGS | TESTS

W
word2vec algo=skipgram dim=300 win=10 alpha=0.025 hs=False neg=10 epochs=10
Even fuller word2vec, more epochs
Training - 0.0%

W
word2vec algo=skipgram dim=100 win=7 alpha=0.025 hs=False neg=5 epochs=2
Full word2vec

S
svd dim=300 win=5 cds=0.75 context=True
Reduced SVD

W
word2vec algo=skipgram dim=300 win=7 alpha=0.025 hs=False neg=5 epochs=1
Full word2vec, no subsampling

W
word2vec algo=skipgram dim=300 win=7 alpha=0.025 hs=False neg=7 epochs=1
Full word2vec

Create new embedding

Model: word2vec

Description: Este es un nuevo embedding...

Model parameters

Dimension: 300

Minimum Count: 5

Window: 5

Subsampling: 0

Algorithm: Skipgram

Hierarchical Softmax

Negative Sampling

Epochs

CREATE

Nabu

DASHBOARD | CORPUS | EMBEDDINGS | TESTS

(0 tests being run)

A	A01	Full Mikolov's test set
A	A02	Occupation genders
A	A03	Verbs: full
A	A04	Spain conjugations: full

Tests > **A01**

ANALOGIES

Name

A01

Description

Full Mikolov's test set

Example

'Atenas' is to 'Grecia' as 'Bagdad' is to... ('Irak')

Running tests

Evaluation results

Date	Embedding Description	Embedding Model	Accuracy	Details
15/03/2016 10:24	Full word2vec, more epochs, with accents	word2vec	63%	
01/03/2016 12:25	Full word2vec, more epochs	word2vec	60%	
13/03/2016 19:40	Full word2vec, extra dimensions	word2vec	57%	

Muchas gracias

Espacio para preguntas, aplausos, críticas
y todo ese tipo de cosas