The background is a vibrant collage of various icons and speech bubbles. Icons include a globe, a laptop, a gear, a document with a pencil, a cloud with three people, a hand pointing at a screen, a mail envelope, a musical note, a smartphone, and a cloud with a gear. Speech bubbles in various colors (pink, yellow, blue, green) are scattered around the central text box. At the bottom, there are stylized human figures in different colors (pink, blue, orange, yellow, green).

Extracción de entidades y aspectos para la tarea del Análisis de Opiniones a nivel de Aspecto

Nuria Rodríguez Barroso

Tutores:

**Francisco Herrera Triguero
Eugenio Martínez Cámara**

- 1 Introducción
 - Objetivos
 - Motivación
 - Análisis de Opiniones
- 2 Fundamentos teóricos
- 3 Fundamentos matemáticos
- 4 Implementación y Experimentación
- 5 Conclusiones

Objetivos

Comparación de modelos basados en redes neuronales para la tarea de la Extracción de Entidades

- Desarrollo de modelos basados en redes neuronales convolucionales (CNN)
- Desarrollo de un modelo basado en redes neuronales recurrentes (LSTM)
- Análisis y comparativa de ambos modelos
- Estudio teórico de ambos modelos

- Está en la naturaleza del ser humano dar la opinión.
- La Web 2.0 supuso una nueva (y enorme) fuente de información.
- Información de gran utilidad para empresas.
- Cantidades inmensurables → **importancia de monitorización**



Análisis de Opiniones

Análisis de Opiniones

El **Análisis de Opiniones (AO)** se define como el estudio computacional de opiniones, sentimientos y emociones.

- Proceso de clasificación de polaridad.
- Información desestructurada.



Nivel de Análisis

Niveles de análisis

- ➊ **Nivel de documento:** analiza texto completo y asigna probabilidad global.
- ➋ **Nivel de oración:** detecta sentimiento en frases.
- ➌ **Nivel de entidad/aspecto:** detecta el sentimiento u opinión y la entidad a la que se refiere. Es el nivel **más preciso**. (**ABSA**)

Nivel de aspecto / Nivel de oración

*"Lo que más me gusta es la maravillosa **resolución de pantalla**, sin embargo el **precio** es muy elevado."*

Nivel de Análisis

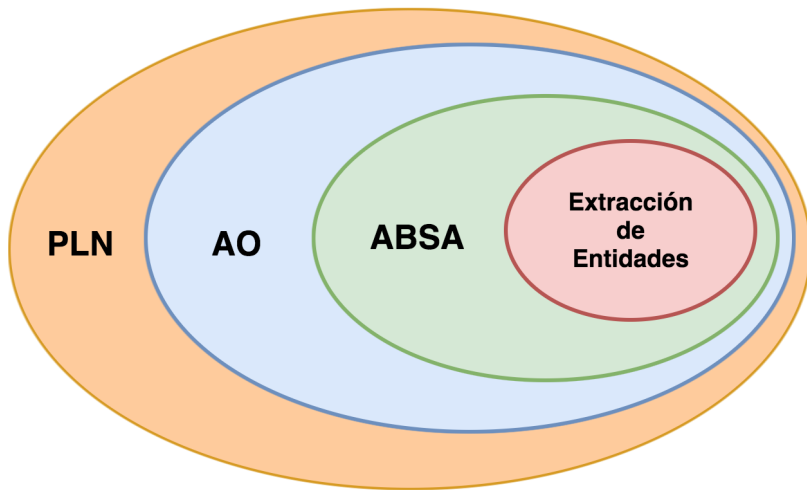
Niveles de análisis

- ➊ **Nivel de documento:** analiza texto completo y asigna probabilidad global.
- ➋ **Nivel de oración:** detecta sentimiento en frases.
- ➌ **Nivel de entidad/aspecto:** detecta el sentimiento u opinión y la entidad a la que se refiere. Es el nivel **más preciso**. (**ABSA**)

Nivel de aspecto / Nivel de oración

*"Lo que más me gusta es la maravillosa **resolución de pantalla**, sin embargo el **precio** es muy elevado."*

Nivel de Análisis



1 Introducción

2 Fundamentos teóricos

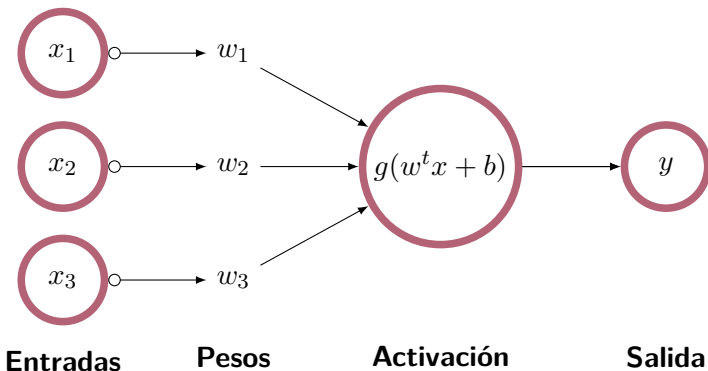
- Redes neuronales
- Redes neuronales convolucionales
- Redes neuronales recurrentes

3 Fundamentos matemáticos

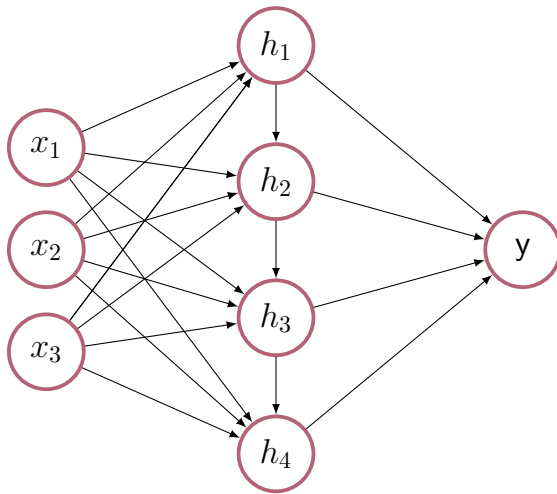
4 Implementación y Experimentación

5 Conclusiones

Neuronas artificiales



Red neuronal artificial

**Entradas****Capa oculta****Salida**

Word embeddings

Problema

Para la aplicación de redes neuronales al Procesamiento del Lenguaje Natural se necesita la conversión de la secuencia de palabras a una estructura tratable.

Solución: *Word embeddings*

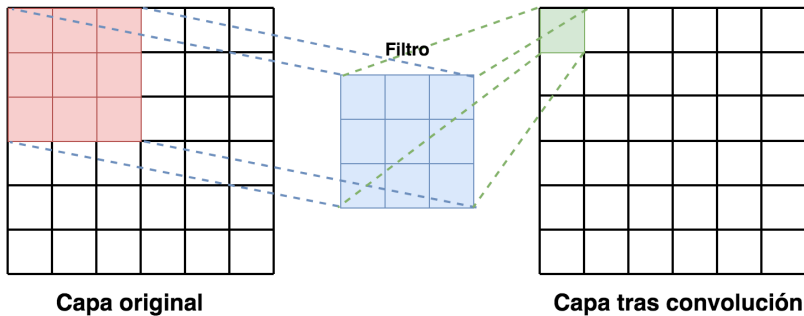
Los ***word embeddings*** son vectores que toman valores reales y que representan el significado de una palabra.

Redes neuronales convolucionales (CNN)

CNN

- Procesamiento de datos con estructura específica de cuadrícula → **secuencias de palabras (*embeddings*)**
- Conocimiento de contexto parcial debido a la **operación de convolución**

Operación de convolución



Redes neuronales recurrentes

Innovación

Añade la idea de compartir parámetros a lo largo del aprendizaje →

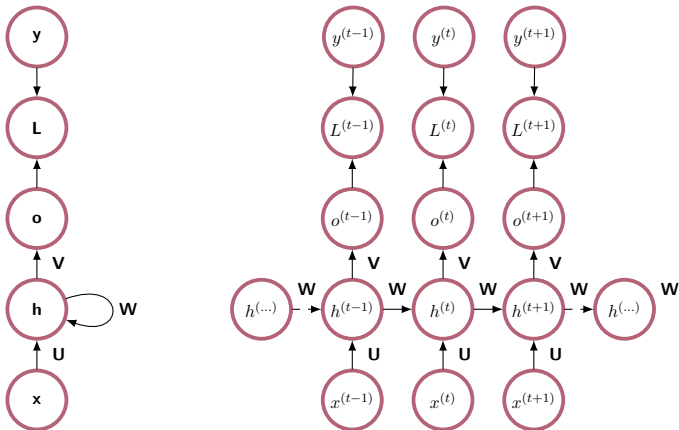
CONTEXTO

Motivación: Modelado de secuencias

[1] Estuve en Nepal en 2017.

[2] En 2017 estuve en Nepal.

Red neuronal recurrente



Problema de las dependencias a largo plazo

Problema

- El cálculo del gradiente propagado durante muchas etapas tiende a **desaparecer** (la mayoría de las veces) o **explotar**.
- Composición de la misma función múltiples veces → **comportamiento alejado de lo lineal**.

Solución: LSTM

- Ejemplo canónico de redes neuronales recurrentes basadas en puertas.

LSTM

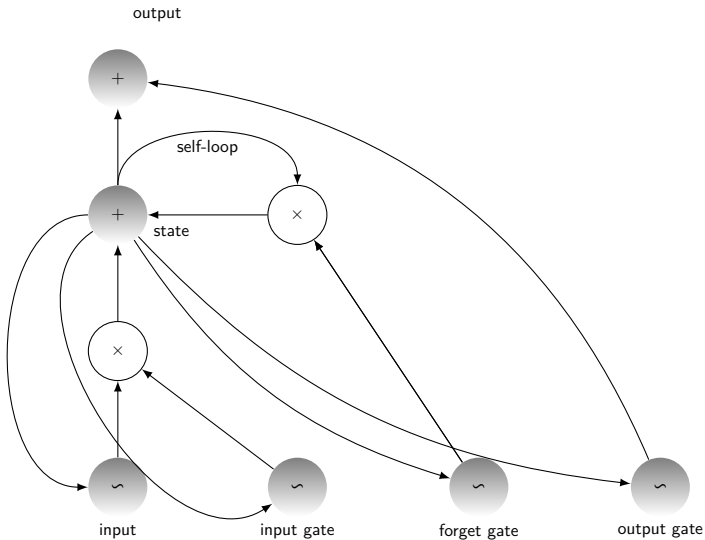
Solución: LSTM

- Celdas conectadas recurrentemente reemplazando capas ocultas de redes neuronales convencionales.
- Introducción de autobuses para controlar el comportamiento del gradiente.
- Pesos del autobús controlados por la unidad de olvido.

Motivación

En la literatura, han mostrado un buen comportamiento en problemas de PLN.

LSTM



biLSTM

Motivación

- Uso de redes neuronales recurrentes → **CONTEXTO**
- Uso de LSTM → **solución de las dependencias a largo plazo.**

Limitaciones

El paso de parámetros (contexto) se realiza solo en un sentido.

Solución: biLSTM

Uso de *biLSTM* (*bidirectional* LSTM):

- Consiste en dos capas LSTM, cada una en un sentido.
- **Contexto en ambos sentidos.**

1 Introducción

2 Fundamentos teóricos

3 Fundamentos matemáticos

- Modelos Ocultos de Markov (HMM)
- Conditional Random Fields (CRF)

4 Implementación y Experimentación

5 Conclusiones

Motivación

Particularidades de nuestro problema

- Las redes neuronales clásicas están diseñadas para producir una salida por entrada → nuestro problema produce una **salida secuencial**
- La función de pérdida clásica es la **entropía cruzada**, que para el caso discreto es

$$H = - \sum \log P(y|x)$$

Dado el caracter secuencial → **Conditional Random Fields (CRF)**

Definiciones previas

Proceso estocástico

Se define un **proceso estocástico** como un conjunto de variables aleatorias que evolucionan en función del tiempo.

Propiedad de Márkov

Se dice que un conjunto de procesos estocásticos verifican la **Propiedad de Márkov** si la distribución de X_{n+1} depende únicamente de X_n . Esto es:

$$P(X_{n+1}|X_n, X_{n-1}, \dots, X_1) = f(X_n)$$

A los procesos estocásticos que verifican esta propiedad se les conoce como **procesos de Márkov**

HMM

Modelo Oculto de Márkov

Se denota un **HMM** mediante quintupla (Y, X, π, A, B) donde:

- ❶ $Y = \{y_1, y_2, \dots, y_n\}$ es el **conjunto de estados**.
- ❷ $X = \{x_1, x_2, \dots, x_m\}$ es el conjunto de **valores observables en cada estado** donde m es el número de palabras posibles.
- ❸ $\pi = \{\pi_i\}$ son las **probabilidades iniciales** donde π_i representa la probabilidad de que el primer estado sea y_i .

HMM

Modelo Oculto de Márkov

- ④ $A = \{a_{ij}\}$ es el conjunto de **probabilidades de transición entre estados** donde a_{ij} es la probabilidad de estar en el estado j en el instante t habiendo estado en i en el instante $t - 1$. Esto es

$$a_{ij} = P(y_t = j \mid y_{t-1} = i) \quad (1)$$

- ⑤ $B = \{b_j(x_k)\}$ es el conjunto de **probabilidades de las observaciones** donde $b_j(x_k)$ representa la probabilidad de observar x_k cuando se está en el estado j en el instante de tiempo t . Esto es

$$b_j(x_k) = P(o_t = x_k \mid y_t = j) \quad (2)$$

La **secuencia de valores observables** se denota como $O = \{o_1, o_2, \dots, o_T\}$.

Conditional Random Fields

Antecedentes

Para modelar la distribución de $P(y, x)$ de una forma tratable, un HMM hace las siguientes suposiciones:

- ❶ Cada estado depende solamente de su predecesor inmediato.
- ❷ Cada observación de la variable x_t depende sólo del estado actual de y_t .

Distribución conjunta

Gracias a esas suposiciones se puede escribir la distribución conjunta como:

$$p(y, x) = \prod_{t=1}^T p(y_t \mid y_{t-1}) p(x_t \mid y_t) \quad (3)$$

Conditional Random Fields

Reescribiendo (3) de la siguiente forma

$$p(y, x) = \frac{1}{Z} \prod_{t=1}^T \exp \left\{ \sum_{i,j \in Y} \theta_{ij} \mathbf{1}_{\{y_t=i\}} \mathbf{1}_{\{y_{t-1}=j\}} + \sum_{i \in Y} \sum_{o \in O} \mu_{oi} \mathbf{1}_{\{y_t=i\}} \mathbf{1}_{\{x_t=o\}} \right\} \quad (4)$$

donde $\Theta = \{\theta_{ij}, \mu_{oi}\}$ son los parámetros de la distribución y Z es una constante de normalización.

Definimos las funciones características de la siguiente forma:

- ① $f_{ij}(y, y', x) = \mathbf{1}_{y=i} \mathbf{1}_{y'=j}$
- ② $f_{io}(y, y', x) = \mathbf{1}_{y=i} \mathbf{1}_{x=o}$

Conditional Random Fields

Usando las funciones características podemos simplificar (4) a

$$p(y, x) = \frac{1}{Z} \prod_{t=1}^T \exp \left\{ \sum_{k=1}^K \theta_k f_k(y_t, y_{t-1}, x_t) \right\} \quad (5)$$

Usando la definición de probabilidad condicionada tenemos

$$p(y \mid x) = \frac{p(y, x)}{\sum_{y'} p(y', x)} = \frac{\prod_{t=1}^T \exp \left\{ \sum_{k=1}^K \theta_k f_k(y_t, y_{t-1}, x_t) \right\}}{\sum_{y'} \prod_{t=1}^T \exp \left\{ \sum_{k=1}^K \theta_k f_k(y'_t, y'_{t-1}, x_t) \right\}} \quad (6)$$

Conditional Random Fields

Conditional Random Fields de Cadena Lineal

Sean Y, X dos vectores aleatorios, $\theta = \{\theta_k\} \in \mathbb{R}^K$ un vector de parámetros y $\mathcal{F} = \{f_k(y, y', x_t)\}_{k=1}^K$ un conjunto de funciones características reales. Entonces un CRF de cadena lineal es una distribución $p(Y | X)$ de la forma

$$p(y | x) = \frac{1}{Z(x)} \prod_{t=1}^T \exp \left\{ \sum_{k=1}^K \theta_k f_k(y'_t, y'_{t-1}, x_t) \right\} \quad (7)$$

donde $Z(X)$ es una función de normalización dependiente de la entrada

$$Z(x) = \sum_Y \prod_{t=1}^T \exp \left\{ \sum_{k=1}^K \theta_k f_k(y'_t, y'_{t-1}, x_t) \right\} \quad (8)$$

Conditional Random Fields

Problema

En la definición $Z(x)$ se realiza un sumatorio sobre todos los posibles estados de la secuencia \rightarrow número **exponencialmente grande**

Solución: Algoritmo de Viterbi

Utilización del **algoritmo de Viterbi**, un algoritmo de programación dinámica para obtener la secuencia de estados ocultos más probable que produce una secuencia observada de sucesos.

Reduce el número de cálculos guardando en memoria sumas intermedias que se repiten.

- 1 Introducción
- 2 Fundamentos teóricos
- 3 Fundamentos matemáticos
- 4 Implementación y Experimentación
 - Consideraciones previas
 - Primer modelo: CNN
 - Segundo modelo: CNN-4-6
 - Tercer modelo: biLSTM
- 5 Conclusiones

Explicación del problema

EXTRACCIÓN DE ASPECTOS/ENTIDADES

- Problema de clasificación multiclase
- Modelado de secuencias
- Etiquetado secuencial
- Modelos basados en redes neuronales

Herramientas

- **Python 3.0:**
 - Facilidad de uso y comprensión
 - Código abierto
 - Multiplataforma
 - Bibliotecas para aprendizaje automático
- **TensorFlow:**
 - Útil para *deep learning*
 - Rendimiento
 - Flexibilidad
- **Keras:** API de alto nivel escrita en *Python* capaz de ejecutarse encima de *Tensorflow*.



Procesamiento de datos

Modelado de secuencias de palabras

Necesitamos datos numéricos en estructura de cuadrícula

- *Tokenización*
- *Padding*/Relleno
- *Word Embeddings*

Procesamiento de datos

Etiquetado de entidades/aspectos

Para el etiquetado de entidades se ha utilizado el formato **IOB** que consta de las siguientes etiquetas:

- **I** - (in)
- **O** - (out)
- **B** - (begin)

Ejemplo

(Lo,O) (que,O) (más,O) (me,O) (gusta,O) (es,O) (la,O) (maravillosa,O)
(resolución,B) (de,I) (pantalla,I) (, ,O) (sin,O) (embargo,O) (el,O)
(precio,B) (es,O) (muy,O) (elevado,O) (.,O)

Consideraciones generales

- ❶ **Punto de partida:** Artículo de extracción de entidades CNN de Poria et al.
- ❷ **Datos utilizados:** comentarios sobre portátiles de *SemEval-2014*
 - 3045 opiniones de entrenamiento
 - 800 opiniones de test
- ❸ ***Word embeddings* utilizados:** *GloVe*
- ❹ **Medidas de error:** se han considerado diferentes etiquetados:
 - *Macro-average*
 - Etiquetado parcial
 - Etiquetado total

Para cada uno de los etiquetados se han tomado las siguientes medidas:

- Precisión
- *Recall*
- F1-score

Medidas de error

Macro-average

$$PRE_{macro} = \frac{PRE_B + PRE_I + PRE_O + PRE_{Padding}}{4}$$

$$RECALL_{macro} = \frac{RECALL_B + RECALL_I + RECALL_O + RECALL_{Padding}}{4}$$

Problema

Menos significativa por el **desbalanceo de clases**

Medidas de error

Etiquetado parcial

Trata de medir la precisión y el *recall* a la hora de detectar la pertenencia de palabras a entidades.

Etiquetado total

Considera el etiquetado correcto si se ha encontrado la entidad completa.

Medida más útil para el resto de tareas del AO.

Medidas de error

Etiquetado Parcial VS Etiquetado Total

(Lo,O) (que,O) (más,O) (me,O) (gusta,O) (es,O) (la,O) (maravillosa,O)
(resolución,B) **(de,I)** **(pantalla,O)** (, ,O) (sin,O) (embargo,O) (el,O)
(precio,B) (es,O) (muy,O) (elevado,O) (.,O)

- Recall parcial = $\frac{3}{4}$

- Precisión parcial = 1

- Recall total = $\frac{1}{2}$

- Precisión total = 1

Consideraciones generales

Arquitecturas implementadas

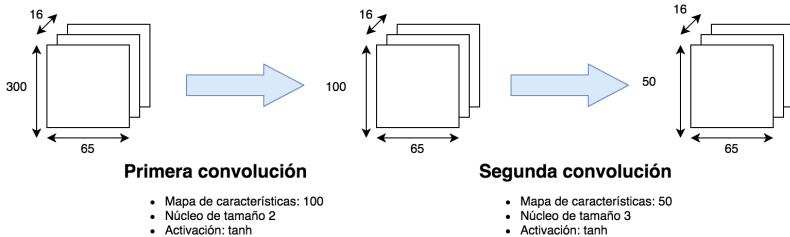
- Red neuronal convolucional (**CNN**) reimplementación del artículo.
- Mejora del primer modelo ampliando tamaño de las convoluciones (**CNN-4-6**).
- Modelo basado en **biLSTM**.

Consideraciones generales

Consideraciones generales para la experimentación y comparación de resultados

- $batch = 16$
- $epochs = 30$
- Se han realizado 20 repeticiones de cada experimento y los resultados mostrados son la media aritmética de ellos.

Arquitectura



Consideraciones

- Tras cada convolución se ha aplicado una capa de *max-pooling* de tamaño 2 y un *Dropout* con parámetro de regularización 0.5.
- Para la salida utilizamos una capa *fully-connected*.

Resultados

		CNN		Artículo
		Media	Varianza	Media
Macro	Precisión	0.92132	2.07×10^{-5}	-
	Recall	0.92426	2.1×10^{-5}	-
	F1	0.92279	2.04×10^{-5}	-
Etiqu. total	Precisión	0.78600	2.87×10^{-4}	-
	Recall	0.64176	5.3×10^{-4}	-
	F1	0.70625	2.05×10^{-4}	-
Etiqu. parcial	Precisión	0.89372	8.3×10^{-4}	-
	Recall	0.72280	1.5×10^{-3}	-
	F1	0.79922	1.9×10^{-4}	0.7732

Análisis de errores

Principales errores del primer modelo:

- Etiquetado de entidades complejas (más de dos palabras) → **reducido contexto**
- Oraciones con varias entidades → **reducido contexto**
- Asignación de etiqueta **B** a la palabras que no pertenecen a ninguna entidad

Análisis de errores

Entidades complejas.

[1] (One, O) (night, O) (I, O) (turned, O) (the, O) (freaking, O) (thing, O) (off, O)
 (after, O) (using, O) (it, O) (, , O) (the, O) (next, O) (day, O) (I, O) (turn, O) (it, O)
 (on, O) (, , O) (no, O) (**GUI, B**) (, , O) (**screen, B**) (all, O) (dark, O) (, ,) (**power, B**)
 (**light, I**) (steady, O) (, ,) (**hard, B**) (**drive, I**) (**light, I**) (steady, O) (and, O) (not, O)
 (flashing, O) (as, O) (it, O) (usually, O) (does, O).

[1] (One, O) (night, O) (I, O) (turned, O) (the, O) (freaking, O) (thing, O) (off, O)
 (after, O) (using, O) (it, O) (, , O) (the, O) (next, O) (day, O) (I, O) (turn, O) (it, O)
 (on, O) (, , O) (no, O) (**GUI, B**) (, , O) (**screen, B**) (all, O) (dark, O) (, ,) (**power, B**)
 (**light, I**) (steady, O) (, ,) (**hard, B**) (**drive, I**) (**light, O**) (steady, O) (and, O) (not, O)
 (flashing, O) (as, O) (it, O) (usually, O) (does, O).

Arquitectura

Limitaciones primer modelo

La principal limitación del primero modelo es → **reducido contexto**

Solución propuesta

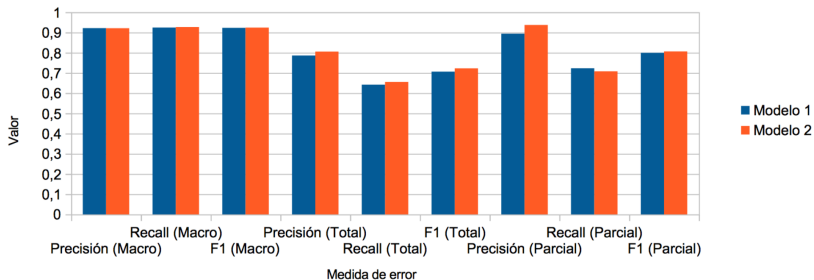
Aumentar el contexto que conoce cada palabra ampliando los filtros de convolución.

Estructura

La estructura del modelo será la misma con filtros de convolución 4 y 6.

Resultados

Comparativa modelos 1 y 2



Análisis de errores

Principales errores del segundo modelo y mejoras con respecto al primero:

- Etiquetado de entidades complejas (más de tres palabras) → **reducido contexto**
- Oraciones con varias entidades → **reducido contexto**
- Se disminuye asignación de etiqueta **B** a palabras que no pertenecen a ninguna entidad → **aumento precisión**

Análisis de errores

Entidades complejas y oraciones con múltiples entidades.

[1] (One, O) (night, O) (I, O) (turned, O) (the, O) (freaking, O) (thing, O) (off, O)
 (after, O) (using, O) (it, O) (, , O) (the, O) (next, O) (day, O) (I, O) (turn, O) (it, O)
 (on, O) (, , O) (no, O) (**GUI, B**) (, , O) (**screen, B**) (all, O) (dark, O) (, ,) (**power, B**)
 (**light, I**) (steady, O) (, ,) (**hard, B**) (**drive, I**) (**light, I**) (steady, O) (and, O) (not, O)
 (flashing, O) (as, O) (it, O) (usually, O) (does, O).

[1] (One, O) (night, O) (I, O) (turned, O) (the, O) (freaking, O) (thing, O) (off, O)
 (after, O) (using, O) (it, O) (, , O) (the, O) (next, O) (day, O) (I, O) (turn, O) (it, O)
 (on, O) (, , O) (no, O) (**GUI, O**) (, , O) (**screen, B**) (all, O) (dark, O) (, ,) (**power, B**)
 (**light, I**) (steady, O) (, ,) (**hard, B**) (**drive, I**) (**light, I**) (steady, O) (and, O) (not, O)
 (flashing, O) (as, O) (it, O) (usually, O) (does, O).

Análisis de errores

Entidades complejas y oraciones con múltiples entidades.

[1] (So,O) (, , O) (I,O) (paid, O) (a, O) (visit, O) (to, O) (**LG,B**)
(notebook,I) (**service, I**) (**center, I**) (at,O) (Alexandra,O) (Road,O) (, ,
 O) (hoping, O) (they, O) (can, O) (make, O) (the, O) (**hinge, B**)
 (tighter, O) (. , O)

[1] (So,O) (, , O) (I,O) (paid, O) (a, O) (visit, O) (to, O) (**LG,B**)
(notebook,I) (**service, I**) (**center, O**) (at,O) (Alexandra,O) (Road,O) (, ,
 O) (hoping, O) (they, O) (can, O) (make, O) (the, O) (**hinge, O**)
 (tighter, O) (. , O)

Arquitectura

Limitaciones de las CNN

Principal limitación de las redes neuronales convolucionales en este problema → **reducido contexto**

Solución propuesta

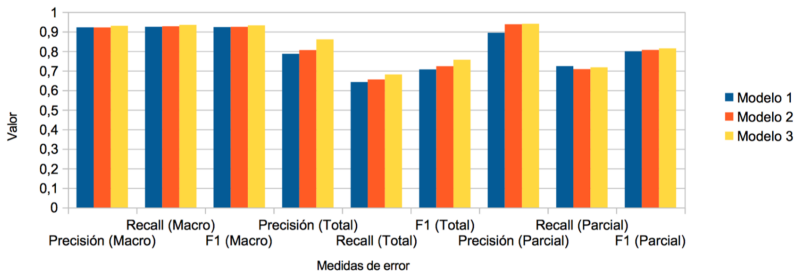
Cambiar el tipo de arquitectura a una red neuronal recurrente → diseñadas para modelado de secuencias

Estructura

La estructura del modelo será un *biLSTM* con 256 unidades ocultas.

Resultados

Comparativa modelos 1,2 y 3



Análisis de errores

Principales errores del tercer modelo y mejoras con respecto a los anteriores:

- Etiquetado de entidades complejas (más de tres palabras) → **falta de casos en el conjunto de entrenamiento**
- Oraciones con varias entidades → **solucionado**
- Asignación de etiqueta **B** a palabras que no pertenecen a ninguna entidad prácticamente inexistente → **aumento precisión**

Análisis de errores

Entidades complejas y oraciones con múltiples entidades.

[1] (One, O) (night,O) (I,O) (turned,O) (the,O) (freaking,O) (thing,O) (off,O)
 (after,O) (using,O) (it,O) (, ,O) (the,O) (next,O) (day,O) (I,O) (turn,O) (it,O)
 (on,O) (, ,O) (no,O) (**GUI,B**) (, , O) (**screen,B**) (all,O) (dark,O) (, ,) (**power,B**)
 (**light,I**) (steady,O) (, ,) (**hard,B**) (**drive, I**) (**light, I**) (steady, O) (and,O) (not,O)
 (flashing,O) (as,O) (it,O) (usually,O) (does,O).

[1] (One, O) (night,O) (I,O) (turned,O) (the,O) (freaking,O) (thing,O) (off,O)
 (after,O) (using,O) (it,O) (, ,O) (the,O) (next,O) (day,O) (I,O) (turn,O) (it,O)
 (on,O) (, ,O) (no,O) (**GUI,B**) (, , O) (**screen,B**) (all,O) (dark,O) (, ,) (**power,B**)
 (**light,I**) (steady,O) (, ,) (**hard,B**) (**drive, I**) (**light, I**) (steady, O) (and,O) (not,O)
 (flashing,O) (as,O) (it,O) (usually,O) (does,O).

Análisis de errores

Entidades complejas y oraciones con múltiples entidades.

[1] (So,O) (, , O) (I,O) (paid, O) (a, O) (visit, O) (to, O) **(LG,B)**
(notebook,I) **(service, I)** **(center, I)** (at,O) (Alexandra,O) (Road,O) (, ,
 O) (hoping, O) (they, O) (can, O) (make, O) (the, O) **(hinge, B)**
 (tighter, O) (. , O)

[1] (So,O) (, , O) (I,O) (paid, O) (a, O) (visit, O) (to, O) **(LG,B)**
(notebook,I) **(service, I)** **(center, O)** (at,O) (Alexandra,O) (Road,O) (, ,
 O) (hoping, O) (they, O) (can, O) (make, O) (the, O) **(hinge, B)**
 (tighter, O) (. , O)

Comparativa

		Artículo	CNN	CNN-4-6	biLSTM
Macro	Precisión	-	0.92132	0.92099	0.92910
	Recall	-	0.92426	0.92684	0.93373
	F1	-	0.92279	0.92390	0.93141
Total	Precisión	-	0.78600	0.80542	0.85985
	Recall	-	0.64176	0.65517	0.68033
	F1	-	0.70625	0.72243	0.75565
Parcial	Precisión	-	0.89372	0.93637	0.93987
	Recall	-	0.72280	0.70783	0.71689
	F1	0.7732	0.79372	0.80612	0.81407

- 1 Introducción
- 2 Fundamentos teóricos
- 3 Fundamentos matemáticos
- 4 Implementación y Experimentación
- 5 Conclusiones
 - Vías futuras

Conclusiones y trabajo futuro

- Se ha demostrado el buen funcionamiento de las *biLSTM* en problemas de modelado de secuencias.
- Se ha conseguido superar al artículo aumentando en un 2,5 % el F1 y en un 5 % la precisión.
- Se propone el desarrollo de un sistema software basado en *biLSTM* más complejo:
 - Arquitectura más compleja.
 - Hibridación con redes neuronales convolucionales.
 - Uso de modelos de atención.
 - Estudio del modelo en otros conjuntos del estado del arte.



¡Gracias por su atención!

Nuria Rodríguez Barroso
rbnuria6@gmail.com

