

Argumentación en Prensa Cubana

Luis Ernesto Ibarra Vázquez¹, Luis Enrique Dalmau Coopat², and Adrián Hernández Pérez³

Universidad de La Habana

Resumen El proyecto se basa en el estudio de la argumentación en los periódicos cubanos. El objetivo principal es el análisis de las estructuras argumentativas que aparecen en ellos para conocer cuáles son las más usadas, cuáles son los argumentos expresados, entre otras estadísticas. En una primera etapa se concentrará en la segmentación y en la clasificación entre argumentos o no argumentos del texto, luego se clasificarán las cláusulas según el rol que jueguen y por último se observará la relación existente entre dichas cláusulas.

Keywords: First keyword · Second keyword · Another keyword.

1. Introducción
2. Problema a resolver
3. Modelación
- 3.1. Seq2Seq
- 3.2. Convolución

Estudios previos (Santos y Zadrozny, 2014; Chiu y Nichols, 2015) han demostrado que CNN es un enfoque eficaz para extraer información morfológica (como el prefijo o el sufijo de una palabra) de los caracteres de las palabras y codificarla en representaciones neuronales. La Figura 1 muestra la CNN que usamos para extraer la representación a nivel de carácter de una palabra dada. La CNN es similar a la de Chiu y Nichols (2015), excepto que usamos solo embeddings de caracteres como entradas a la CNN, sin características de tipo de carácter. Se aplica una capa de dropout (Srivastava et al., 2014) antes de que se ingresen las embeddings de caracteres en CNN.

Figura 1: La red neuronal de convolución para extraer representaciones de palabras a nivel de carácter. Las flechas discontinuas indican una capa de dropout aplicada antes de que se ingresen los embeddings de caracteres en CNN.

3.3. LSTM

Las redes neuronales recurrentes (RNN) son una poderosa familia de modelos conexionistas que capturan la dinámica del tiempo a través de ciclos en el grafo. Aunque, en teoría, las RNN son capaces de capturar dependencias de larga distancia, en la práctica fallan debido a los problemas de desaparición/explosión del gradiente (Bengio et al., 1994; Pascanu et al., 2012). Los LSTM (Hochreiter y Schmidhuber, 1997) son variantes de los RNN diseñados para hacer frente a estos problemas de desaparición de gradientes. Básicamente, una unidad LSTM se compone de tres puertas multiplicativas que controlan las proporciones de información para olvidar y pasar al siguiente paso de tiempo. La Figura 2 muestra la estructura básica de una unidad LSTM.

Figura 2: Esquema de una unidad LSTM

Formalmente las formulas para actualizar una unidad LSTM en un tiempo t son:

$$\begin{aligned} i_t &= (W_i h_{t-1} + U_i x_t + b_i) \\ f_t &= (W_f h_{t-1} + U_f x_t + b_f) \\ c_t &= \tanh(W_c h_{t-1} + U_c x_t + b_c) \\ \tilde{c}_t &= f_t c_{t-1} + i_t c_t \\ o_t &= (W_o h_{t-1} + U_o x_t + b_o) \\ h_t &= o_t \tanh(\tilde{c}_t) \end{aligned}$$

donde \tanh es la función sigmoidal element-wise y CIRCLE SYMBOL es el producto element-wise. x_t es el vector de entrada (e.g. embeddings de palabras) en el tiempo t , h_{t-1} es el vector de estado oculto en el tiempo $t-1$.

BLSTM Para muchas tareas de etiquetado de secuencias, es beneficioso tener acceso a contextos pasados (izquierda) y futuros (derecha). Sin embargo, el estado oculto de LSTM h_t toma información solo del pasado, sin saber nada sobre el futuro. Una solución elegante es usar un BLSTM (Bidirectional LSTM) que combina un LSTM hacia adelante y uno hacia atrás.

3.4. CRF

Para las tareas de etiquetado de secuencias (o predicción estructurada general), es beneficioso considerar las correlaciones entre las etiquetas en los vecindarios y decodificar conjuntamente la mejor cadena de etiquetas para una oración de entrada dada. Por ejemplo, en el etiquetado POS, es más probable que un adjetivo vaya seguido de un sustantivo que de un verbo, y en NER con anotación BIOES-style (Tjong Kim Sang y Veenstra, 1999), I-ORG no puede seguir a I-PER. Por lo tanto, modelamos la secuencia de etiquetas de forma conjunta utilizando un campo aleatorio condicional (CRF) (Lafferty et al., 2001), en lugar de decodificar cada etiqueta de forma independiente. Formalmente, usamos $z = z_1, \dots, z_n$ para representar una secuencia de entrada genérica donde z_i es el vector de entrada de la i -ésima palabra. $y = y_1, \dots, y_n$ representa una secuencia genérica de etiquetas para z . $Y(z)$ denota el conjunto de posibles etiquetas para z .

$$p(y|z; W, b) = \prod_{i=1}^n \phi(y_i|z, y_{i-1}, y_{i+1}) \sum_{y_0 \in Y(z)} \prod_{i=1}^n \phi(y_i|z, y_{i-1}, y_{i+1})$$

donde $\phi(y_i|z, y_{i-1}, y_{i+1}) = \exp(W y_i T_0 + y_{i-1} T_1 + y_{i+1} T_2 + b y_i)$ son funciones de potencia, $W y_i T_0$, $y_{i-1} T_1$, $y_{i+1} T_2$ y $b y_i$ son los vectores de pesos para las transiciones de etiquetas.

$$L(W, b) = \sum_{y \in Y(z)} \log p(y|z; W, b)$$

El entrenamiento de la máxima verosimilitud escoge parámetros tal que la log-verosimilitud es maximizada.

Decodificar es buscar la secuencia de la etiqueta y^* con la mayor probabilidad condicional:

$$y = \operatorname{argmax}_{y_{Y(z)}} p(y|z; W, b)$$

Para un modelo CRF secuencial (solo se consideran las interacciones entre dos etiquetas sucesivas), el entrenamiento y la decodificación se pueden resolver de manera eficiente adoptando el algoritmo de Viterbi.

3.5. BLSTM-CNNs-CRF

Finalmente, se construye este modelo de red neuronal alimentando los vectores de salida de BLSTM en una capa CRF. La Figura 3 ilustra la arquitectura de esta red en detalle. Para cada palabra, la CNN calcula la representación a nivel de carácter en la Figura 1 con incrustaciones de caracteres como entradas. Luego, el vector de representación a nivel de carácter se concatena con el vector de inserción de palabras para alimentar la red BLSTM. Finalmente, los vectores de salida de BLSTM se alimentan a la capa CRF para decodificar conjuntamente la mejor secuencia de etiquetas. Como se muestra en la Figura 3, las capas de exclusión se aplican tanto en los vectores de entrada como de salida de BLSTM. Los resultados experimentales muestran que el uso de dropout mejora significativamente el rendimiento de este modelo

Figure 3: The main architecture of our neural network. The character representation for each word is computed by the CNN in Figure 1. Then the character representation vector is concatenated with the word embedding before feeding into the BLSTM network. Dashed arrows indicate dropout layers applied on both the input and output vectors of BLSTM.

4. Implementación

4.1. tensorflow

5. Entrenamiento

5.1. Corpus Usado

5.2. Hiperparámetros y optimización de estos

6. Evaluación

6.1. Resultados

7. Conclusiones

Referencias

1. Author, F.: Article title. *Journal* **2**(5), 99–110 (2016)
2. Author, F., Author, S.: Title of a proceedings paper. In: Editor, F., Editor, S. (eds.) *CONFERENCE 2016, LNCS*, vol. 9999, pp. 1–13. Springer, Heidelberg (2016). <https://doi.org/10.1007/1234567890>
3. Author, F., Author, S., Author, T.: Book title. 2nd edn. Publisher, Location (1999)
4. Author, A.-B.: Contribution title. In: *9th International Proceedings on Proceedings*, pp. 1–2. Publisher, Location (2010)
5. LNCS Homepage, <http://www.springer.com/lncs>. Last accessed 4 Oct 2017