

Como enseñar a las máquinas a leer y comprender

Sanchez, Lesly
Universidad Nacional
de Ingeniería
Lima, Perú

lesly.sanchez.r@uni.pe

Yagi, Gladys
Universidad Nacional
de Ingeniería
Lima, Perú

gladys.yagi.v@uni.pe

Resumen—La capacidad de una máquina de poder leer y comprender un texto En el presente trabajo

Palabras Claves:

I. INTRODUCCIÓN:

Los avances en los algoritmos de Deep Learning, con el objetivo de darle a una máquina la capacidad de leer y comprender el lenguaje en documentos, han ido en aumento [1]. Esto se debe a que en los últimos años los conjuntos de datos para entrenamiento y prueba a gran escala para evaluar este tipo de capacidad han aumentado.

La capacidad de las máquinas para leer y comprender un determinado documento puede ser testeada mediante su habilidad para poder responder preguntas y respuestas por medio de contenido de documentos, predecir palabras en historias incompletas, resumir reseñas de productos o predecir si una premisa y su hipótesis se contradicen o son neutrales entre sí. Un conjunto de datos famoso por ser utilizado para la comprensión lectora de las máquinas es el CNN/Daily Mail.

Original Version	Anonymised Version
Context The BBC producer allegedly struck by Jeremy Clarkson will not press charges against the "Top Gear" host, his lawyer said Friday. Clarkson, who hosted one of the most-watched television shows in the world, was dropped by the BBC Wednesday after an internal investigation by the British broadcaster found he had subjected producer Oisin Tymon "to an unprovoked physical and verbal attack." ...	the <i>ent381</i> producer allegedly struck by <i>ent212</i> will not press charges against the " <i>ent153</i> " host, his lawyer said Friday. <i>ent212</i> , who hosted one of the most-watched television shows in the world, was dropped by the <i>ent381</i> Wednesday after an internal investigation by the <i>ent180</i> broadcaster found he had subjected producer <i>ent193</i> "to an unprovoked physical and verbal attack." ...
Query Producer X will not press charges against Jeremy Clarkson, his lawyer says.	producer X will not press charges against <i>ent212</i> , his lawyer says.
Answer Oisin Tymon	<i>ent193</i>

Figura 1. Versión original y anonimizada de un artículo extraído del conjunto de Datos de validación del Daily Mail. Las entidades anonimizadas están constantemente permutándose durante la prueba y entrenamiento.

Aquí podemos observar como se extraen las entidades de oraciones clave de los artículos de las noticias para ser reemplazados con espacios en blanco. Para así poder

entrenar a la máquina en responder preguntas dependiendo del texto restante.

En este trabajo, se analizará un modelo de red neuronal recurrente de forma que con la ayuda de la plataforma ParlAI, esta sea capaz de completar oraciones a partir de leer y comprender un libro para niños.

II. OBJETIVO DEL ESTUDIO

Comprender y analizar las redes neuronales convolucionales y como estas nos permiten que una máquina puede responder preguntas sobre un documento.

III. MARCO TEÓRICO

III-A. Natural Language Processing

El procesamiento del lenguaje natural (NLP) es un campo de la ciencia de la computación que se ocupa del uso de las computadoras para la extracción, interpretación, análisis y manipulación del lenguaje humano. Algo relacionado al NLP es el Question answering (QA), el cual es una tarea del NLP encargada de responder a preguntas en el lenguaje natural, aprovechando los métodos de recuperación de información y representación del conocimiento [2]. Siendo de esta manera la forma en la que la máquina aprenderá a leer y comprender.

III-B. Recurrent Neural Network

Las redes neuronales recurrentes (RNN) tienen la capacidad de modelar la información contextual en secuencias largas. Para el procesamiento del lenguaje natural se desea capturar las dependencias a largo plazo; además de mantener el orden contextual entre las palabras para resolver el significado global de un texto. Otra ventaja de las RNN es su capacidad para aprender representaciones de secuencias de longitud variable, como frases, documentos y muestras de habla [2].

III-C. Long Short-Term Memory

La memoria a largo y corto plazo (LSTM) utiliza puertas de entrada, salida y olvido para controlar la propagación del gradiente en la memoria de la red neuronal recurrente (RNN). Esto permite proteger las celdas de memoria que lleven algún estado oculto al siguiente paso. Los estados de activación permiten que la red aprenda las condiciones para cuando olvidar o mantener la información en la celda de memoria [2].

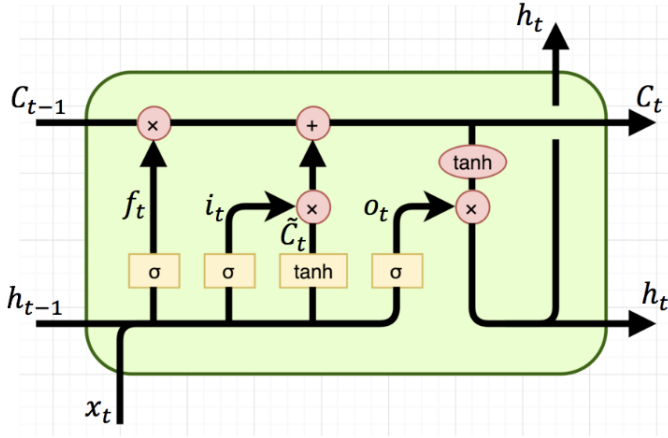


Figura 2. Diagrama de una celda LSTM

III-D. Seq2Seq

El modelo Sequence to Sequence se utiliza cuando tenemos una entrada de longitud variable que no es necesariamente una entidad asociada a otra. Un modelo Seq2Seq se compone más comúnmente de dos RNN que trabajan de manera cooperativa: encoder y decoder. Ambas siendo modelos LSTM. Es usado comúnmente para máquinas de traducción, resumen de texto, modelos de conversación, predicción de palabras, y más.

III-D1. Encoder: Se encargará de codificar el input palabra por palabra en un vector de estado o de contexto (en el caso de LSTM estas son llamadas estado oculto o vectores de estado de las celdas).

III-D2. Decoder: Este es el estado oculto final por parte codificador. Se encarga de encapsular todos los elementos de entrada para poder realizar una predicción.

En la Figura 2, el encoder codifica una secuencia de entrada pasándolas por las celdas LSTM, este encoder consta de los estados ocultos; luego, se pasa a la decodificación como estados iniciales junto con los inputs del decodificador para poder hacer predicciones.

Cada palabra es representada como x_i donde i es el orden de cada palabra de un artículo, h_i es un estado

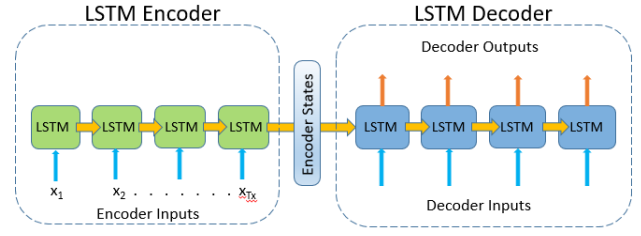


Figura 3. Modelo encoder-decoder con LSTM unidireccionales

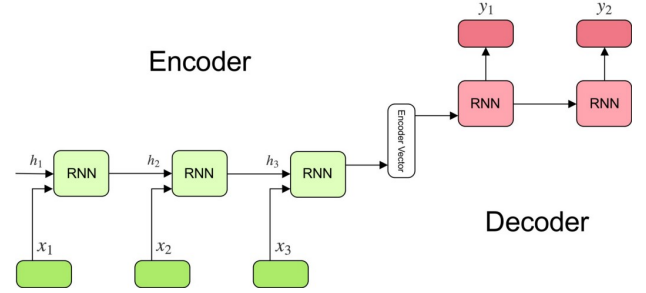


Figura 4. Encoder-decoder

oculto y W es el peso respectivo.

Los estados ocultos en el encoder se calculan usando la fórmula:

$$h_t = f(W^{(hh)}h_{t-1} + W^{(hx)}x_t) \quad (1)$$

Cualquier estado oculto para el decoder se calcula usando la fórmula:

$$h_t = f(W^{(hh)}h_{t-1}) \quad (2)$$

Y la salida se calcula mediante:

$$y_t = \text{softmax}(W^s h_t) \quad (3)$$

IV. ESTADO DEL ARTE:

En la actualidad existen novedosas arquitecturas de redes neuronales para tareas de comprensión lectora de la máquina. En esta sección describiremos algunas de estas.

IV-A. Deep LSTM Reader

Deep LSTM Reader propuesto en el artículo Teaching Machines to Read and Comprehend [1] es un modelo que pone a prueba la capacidad de los Deep LSTM encoder para manejar secuencias largas de inputs. Este modelo se alimenta palabra por palabra de artículos del CNN al Deep LSTM encoder y luego con un delimitador también se introducen consultas al codificador. Como resultado, se obtiene un modelo que procesa cada par de consultas del artículo como una única secuencia larga. Dado el artículo y la consulta, el modelo predice el token del artículo que responde a la consulta.

IV-B. ReasoNet

ReasoNet propuesto en el artículo ReasoNet: Learning to Stop Reading in Machine Comprehension [3] es una arquitectura de red neuronal llamada red de razonamiento la cual tiene el fin de realizar las tareas de comprensión de la máquina. Las ReasoNets hacen uso de múltiples giros para explotar eficientemente y razonar sobre la relación entre las consultas, los documentos y las respuestas. Con el uso de aprendizaje por refuerzo, las ReasoNets pueden determinar dinámicamente continuar el proceso de comprensión tras digerir los resultados resultados intermedios, o terminar la lectura cuando concluye que información existente es adecuada para producir una respuesta.

IV-C. R-NET

ReasoNet propuesto en el artículo R-Net: Machine Reading Comprehension with self-matching networks [4] es un modelo de redes neuronales de extremo a extremo para responder a preguntas del estilo de la comprensión lectora, cuyo objetivo es responder a preguntas de un pasaje dado. En este artículo primero se empareja la pregunta y el pasaje con redes recurrentes basadas en la atención para obtener la representación del pasaje consciente de la pregunta. Luego propone un mecanismo de atención auto-ajustada para refinar la representación comparando el pasaje consigo mismo, lo que codifica eficazmente la información de todo el pasaje. Por último emplea redes de punteros para localizar las posiciones de las respuestas de los pasajes.

V. METODOLOGÍA:

En esta sección se describirá los pasos del desarrollo del proyecto y las herramientas necesarias.

La implementación de este proyecto se hará en lenguaje Python, para la creación de redes neuronales se usará Pytorch, que es una librería de Python la cual dispone de una interfaz sencilla para la creación de redes neuronales.

Usaremos ParlAI que es un framework en Python para permitir la investigación de Inteligencia Artificial de diálogo. Además este framework cuenta con muchos conjuntos de datos populares para realizar múltiples tareas sobre ellos, una amplia gama de ayudantes para crear agentes y capacitarlos en varias tareas.

Pasos del desarrollo del proyecto:

1. Primero instalamos el framework ParlAI
2. Luego importamos el dataset CBT
3. Implementamos los agentes para el modelo
4. Creamos el modelo
5. Entrenamos el modelo

VI. EXPERIMENTACIÓN Y RESULTADOS

VI-A. Conjunto de Datos

Para la evaluación del modelo se tomó el dataset Children'S Book Test (CBT) ya que este era el de menor

tamaño y esto haría que el modelo cargue más rapido el dataset en comparación a otros datasets conocidos. A continuación mostraremos una estadística sobre los 4 conjuntos de datos estándar de comprensión de texto y el CBT.

	CNN	Daily Mail	CBT CN	CBT NE	BookTest
# queries	380,298	879,450	120,769	108,719	14,140,825
Max # options	527	371	10	10	10
Avg # options	26.4	26.5	10	10	10
Avg # tokens	762	813	470	433	522
Vocab. size	118,497	208,045	53,185	53,063	1,860,394

Figura 5. Tabla comparativa de datasets de comprensión de texto

Entrenamos el modelo con un batch y tres batches, obtuvimos los siguientes resultados:

1. Para un batch

Evaluación para los datos de validación

```
03:50:52 | running eval: valid
04:11:08 | eval completed in 1216.06s
04:11:08 | valid:
all      accuracy  bleu-4  clen  ctpb  ctps  ctrunc  ctrunclen  exps  exs  f1  gpu_mem  llen  loss  lr  ltpb  ltps \
cbt:CN   .000525  6.25e-13  512.3  512.3  3370    0           0  6.579  8800  .000525  .2580  2.013  9.236  1  2.013  13.24
cbt:NE   .00010  1e-12  522.1  0       0       0       0  2000  .0010  2.004  11.07
cbt:CN   0       0  489    0       0       0  2000  0      2.047  10.06
cbt:P    .0015  1.5e-12  528.9  0       0  2000  .0015  2  5.784
cbt:V    0       0  509.2  0       0  2000  0      2  10.04

litrunc  ltrunclen  ppl  token_acc  token_em  total_train_updates  tpb  tps
all      0       0  27608    .3525  .000625  50  514.3  3384
cbt:CN   0       0  63955    .3403  .0010
cbt:NE   0       0  23331    .2694  0
cbt:P    0       0  325     .4740  .0015
cbt:V    0       0  22820    .3262  0
```

Figura 6. Tabla 1

Evaluación para los datos de test

```
04:56:25 | test:
all      accuracy  bleu-4  clen  ctpb  ctps  ctrunc  ctrunclen  exps  exs  f1  gpu_mem  llen  loss  lr  ltpb  ltps \
cbt:CN   .0003  3.368e-13  524.4  524.4  3456    0           0  6.591  10000  .0003667  .1937  2.029  9.459  1  2.029  13.37
cbt:NE   .0008  8e-13  539.3  0       0       0  2500  .0008  2.004  11.03
cbt:CN   0  1.472e-13  503.7  0       0  2500  .0002667  2.11  10.97
cbt:P    .0004  4e-13  537.1  0       0  2500  .0004  2  5.86
cbt:V    0       0  517.3  0       0  2500  0      2.001  9.969

litrunc  ltrunclen  ppl  token_acc  token_em  total_train_updates  tpb  tps
all      0       0  35458    .3458  .0003  50  526.4  3469
cbt:CN   0       0  61960    .3154  .0008
cbt:NE   0       0  58171    .2527  0
cbt:P    0       0  350.8    .4698  .0004
cbt:V    0       0  21349    .3455  0
```

Figura 7. Tabla 2

2. Para tres batches

Evaluación para los datos de validación Evaluación para

```
05:24:48 | valid:
all      accuracy  bleu-4  clen  ctpb  ctps  ctrunc  ctrunclen  exps  exs  f1  gpu_mem  llen  loss  lr  ltpb  ltps \
cbt:CN   .007875  7.892e-12  512.3  1536  8201    0           0  16.01  8000  .007938  .1936  2.013  9.003  1  6.035  32.22
cbt:NE   .0010  1e-12  522.1  0       0       0  2000  .0010  2.004  11.36
cbt:CN   0  6.767e-14  489    0       0  2000  .00025  2.047  10.13
cbt:P    .0305  3.05e-11  528.9  0       0  2000  .0305  2  7.11
cbt:V    0       0  509.2  0       0  2000  0      2  11.02

litrunc  ltrunclen  ppl  token_acc  token_em  total_train_updates  tpb  tps
all      0       0  43190    .3975  .00775  51  1542  8233
cbt:CN   0       0  85669    .3787  .0010
cbt:NE   0       0  25088    .3464  0
cbt:P    0       0  1224     .4975  .0300
cbt:V    0       0  60780    .3675  0
```

Figura 8. Tabla 3

los datos de test

VII. ANÁLISIS

Observamos las tablas 1 y 3 para el análisis de la métrica accuracy que nos da la precisión del modelo, analizando las tablas obtenidas al entrenar el modelo con

```

05:35:27 | test:
accuracy  bleu-4  clen  ctpb  ctps  ctrunc  ctrunclen  exps  exs  f1  gpu_mem  llen  loss  lr  ltpb  ltps \
all      .0088  8.8e-12  524.4  1572  8216  0  0  15.67  10000  .0088  .1938  2.029  10  1  6.083  31.79
cbl:CN  0  0  539.4  0  0  0  0  2500  0  2.084  11.25
cbl:NE  0  0  503.7  0  0  0  0  2500  0  2.11  10.7
cbl:P   .0352  3.52e-11  537.1  0  0  0  0  2500  .0352  2  7.159
cbl:V   0  0  517.3  0  0  0  0  2500  0  2.001  10.9

litrunc  ltrunclen  ppl  token_acc  token_em  total_train_updates  cpb  tps
all      0  0  44172  .3888  .0087  51 1578  8248
cbl:CN  0  0  76860  .3615  0
cbl:NE  0  0  44272  .3062  0
cbl:P   0  0  1286  .4996  .0348
cbl:V   0  0  54270  .3880  0

```

Figura 9. Tabla 4

los datos de evaluación de uno y tres lotes, se nota un incremento del accuracy, de 0.000625 % a 0.007875 % . Ahora observamos las tablas 2 y 4 para el análisis de la métrica accuracy, analizando las tablas obtenidas con los datos de test de uno y tres lotes, se nota un incremento del accuracy, de 0.0003 % a 0.0088 % . Ambos incrementos del accuracy ocurrieron por la modificación de la cantidad de lotes, ya que esto hizo que el modelo se beneficie.

VIII. CONCLUSIONES

REFERENCIAS

- [1] Teaching machines to read and comprehend. *Avances en los sistemas de procesamiento de información neuronal*.
- [2] Uday Kamath, John Liu, and James Whitaker. *Deep learning for NLP and speech recognition*, volume 84. Springer, 2019.
- [3] Yelong Shen, Po-Sen Huang, Jianfeng Gao, and Weizhu Chen. Reasonet: Learning to stop reading in machine comprehension. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1047–1055, 2017.
- [4] Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. R-net: Machine reading comprehension with self-matching networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 189–198, 2017.