Generador de Keywords de Artículos Científicos

León, Rafael, 13361, leo13361@uvg.edu.gt

Abstract

Los artículos científicos se popularizaron por la distribución en línea de revistas y casas de publicación científicas como Nature. Estos sirven para categorizar artículos y hacerlos accesibles a través de buscadores indexados como Google.com. Por tanto los artículos que sobresalen al realizar una búsqueda son los que tienen los keywords más acertados al texto escrito por el usuario en el buscador. [1]
Seleccionar keywords correctos es de suma importancia para la visibilidad de un artículo de investigación. Este proyecto propuso la construcción de una red neuronal artificial de tipo Transformer que permita identificar el tema del que se habla en la introducción tipo abstracto de un artículo y devuelve un listado de palabras clave relacionadas al mismo.[6] Con el afán de construir keywords a un artículo que provengan de artículos previamente indexados se utilizó un modelo de entrenamiento supervisado en donde se presentaron 37,900 artículos científicos y sus keywords. Con los que se realizó el entrenamiento, validación cruzada y evaluación. Se obtuvo un valor de perplejidad de 6.604 ppl

Keywords: NLP, Transformers, Artificial Intelligence

Introducción

Los artículos científicos en línea son la forma principal de distribución de conocimiento científico en la actualidad. Las revistas y casas de publicación que antes se dedicaban a realizar números impresos en los que distribuían artículos periódicamente, han migrado a un modelo de distribución inmediata a través de sitios web propietarios o de distribución pública. Estos usualmente contienen barreras de pago para los autores para poder publicar en ellos y barreras de pago para los lectores para poder acceder a los textos completos. Esta barrera de costo, acarrea varios problemas como: dispersión del conocimiento, restricción de acceso y reducción de la visibilidad.
Para lidiar con el problema de reducción de la visibilidad las casas de publicación adjuntan un sistema de clasificación de artículos por medio de palabras clave, (keywords o tags en inglés). Sin embargo, este sistema de clasificación tiene un problema muy grave: **Los autores deciden qué keywords colocar a su artículo.[3]** Este problema hace que artículos sumamente relevantes a un tema no aparezcan en las búsquedas ya que el autor colocó los keywords equivocados. Invisibilizandolos aún más en el océano de artículos disponibles al alcance de los dedos en indexadores y buscadores en línea como google.com.
Por tanto, se propone diseñar una mejor forma de obtener los keywords idóneos para un artículo científico que no dependa del autor. Esto facilita el acceso a los artículos ya que a través de sistemas de clasificación de texto, es posible obtener el significado y el tema

relacionado a un texto. Esto se logró a través de tecnologías innovadoras de inteligencia artificial que permiten procesar texto en distintos idiomas y darle significado en otro idioma (inglés -> inglés simplificado).
Por tanto, este problema, pasa a ser un problema de procesamiento de lenguaje natural clásico de estilo secuencia a secuencia, mejor conocido como seq2seq. Este modelo se encarga de tomar un texto en una secuencia y traducirlo a otra secuencia. Estos modelos se han utilizado con mucho éxito para realizar traducciones de inglés a otros idiomas y son altamente reutilizables ya que se componen por dos módulos: el módulo de codificación y el módulo de decodificación. Estos módulos son intercambiables y un módulo de decodificación en un lenguaje puede ser reutilizado para lidiar con distintos módulos de codificación sin necesidad de entrenarlos nuevamente.
Este proyecto, busca utilizar los modelos más actualizados de seq2seq, llamados **transformadores** (Transformers en inglés) para la generación automatizada de keywords para abstractos científicos. Dada la naturaleza de la tarea, un entendimiento de los keywords que se utilizan en la actualidad y su significado es de suma importancia para producir resultados competentes y que faciliten su búsqueda en internet. Claramente estos matices de selección de keywords son complicadas para un humano que no tiene conocimiento extenso de las keywords en uso y podría ser que con un caso tan sencillo como los keywords de este artículo, puedan haber cientos de opciones de keywords. Por ejemplo, este artículo utiliza el keyword “NLP” referente a “Natural Language Processing”, “Transformers” referente a la implementación de redes neuronales y “Artificial Intelligence” referente a la técnica. Estos tres keywords podrían ser representados de muchas formas distintas como “Text processing”, “Transformer Architecture”, “Neural Network Artificial Intelligence”, algunos de estos son más populares y es posible que sea más difícil de encontrar el artículo si estos keywords son malos.

Trabajo Previo

Se han realizado estudios previos en donde se intenta obtener las keywords por medio de co-ocurrencia de palabras [2] . Los modelos de co-ocurrencia de palabras se han mostrado inferiores en otras tareas por lo que es posible que modelos más actualizados que utilizan redes neuronales puedan obtener mejores resultados. [3]

Marco Metodológico

Se plantearon dos modelos a diseñar, un modelo de redes neuronales recurrentes y un modelo de transformadores.

Arquitectura

Se trabajó con base a la última versión de la biblioteca Pytorch que es una biblioteca de acceso abierto para aprendizaje de máquina basada en la biblioteca Torch. Es comùnmente utilizada para aplicaciones como visión por computadora y procesamiento de lenguaje natural. Está desarrollada principalmente por el equipo de inteligencia artificial de Facebook (FAIR).
Es software gratuito y de acceso libre y se encarga de abstraer modelos de redes neuronales a un nivel mucho más accesible, realizando optimizaciones de la estructura de los tensores que alimentan las redes neuronales y de càlculos matemáticos colocando una interfaz en python que opera sobre c++ y realiza operaciones matemáticas de forma mucho más eficiente que python nativo. Haciéndola una de las bibliotecas más importantes para el desarrollo de aprendizaje de máquina junto con Tensorflow y Keras.

Especificaciones Técnicas

- Anaconda Python 3.7
- Pytorch
- Torchtext

Red neuronal Recurrente RNN (GRU)

Una red neuronal recurrente de tipo GRU (Gated recurrent unit) describe una red neuronal que tiene las características de una RNN pero con pasos agregados en cada capa de recurrencia para obtener mejores resultados. El cambio principal es que cada una de las capas contiene una unidad de memoria y una unidad de “olvido” que permite reducir las correlaciones por sobre entrenamiento y facilitan los pasos de ajuste hacia atrás por medio del almacenamiento temporal de los resultados de cada capa. Esto implica que el cálculo de la capa solo debe hacerse en una dirección (forward) y solo se lee y se ajusta en la dirección opuesta (back propagation)

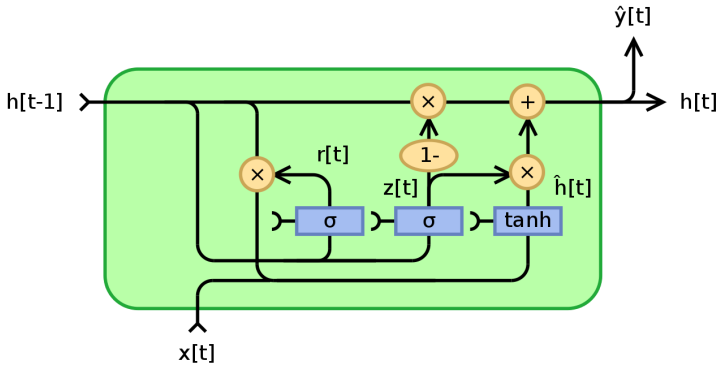


Figura 1: Gated Recurrent Unit. Solo 1 unidad. [5]

Transformador (Transformer)

Un transformador es un modelo de red neuronal de secuencia a secuencia que utiliza el concepto de atención. Atención en psicología es el proceso cognitivo de concentrarse selectivamente en una o varias cosas mientras se ignoran otras. Este mismo proceso se realiza en las redes neuronales por medio de vectores de atención, que colocan importancia sobre ciertos valores e ignoran otros.

Esta arquitectura fue propuesta por primera vez en el artículo “Attention is all you need” [6] en donde definen atención de una forma muy genérica basada en una llave, una pregunta o query y un valor. A través de un sistema denominado Multiheaded attention.

Previamente cuando se calculaba atención esta se realizaba a través de una función matemática como un producto punto. Con el estado oculto de las representaciones de palabras vistas anteriormente. Como los utilizados en implementaciones de atención con LSTM RNNs [7].

El modelo de Multiheaded attention es muy similar al proceso lineal de atención con producto punto. Sin embargo utiliza los valores de llave, query y valor para realizar el cálculo de atención.

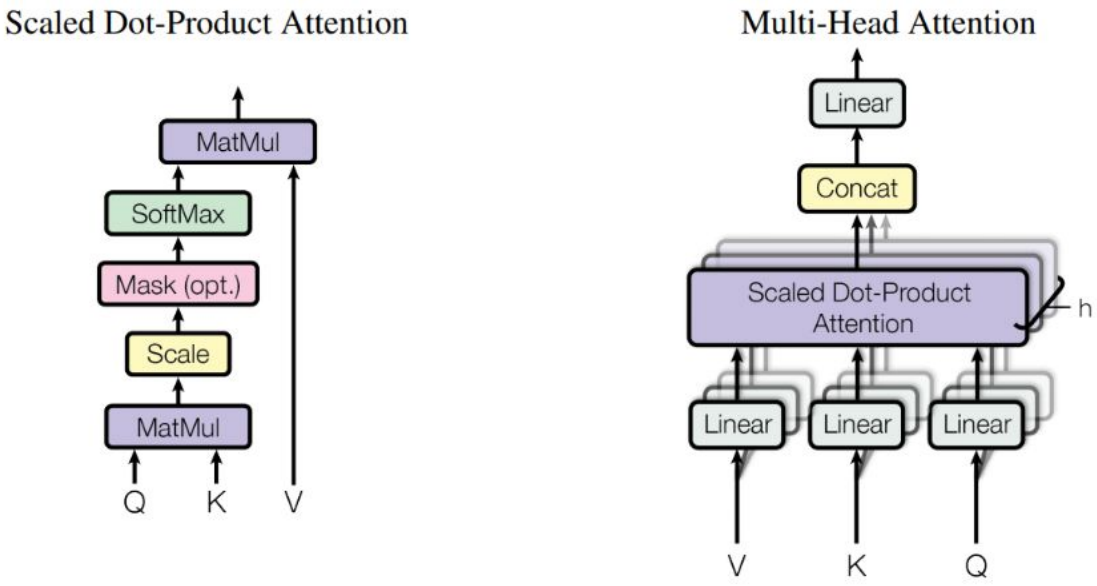


Figura 2: Diagrama de cálculo de Multiheaded attention fuente: [6]

Este proceso de atención define gran parte de la implementación de un transformador y es lo que le permite realizarse en paralelo. La figura 3: Diagrama una red neuronal de transformadores.

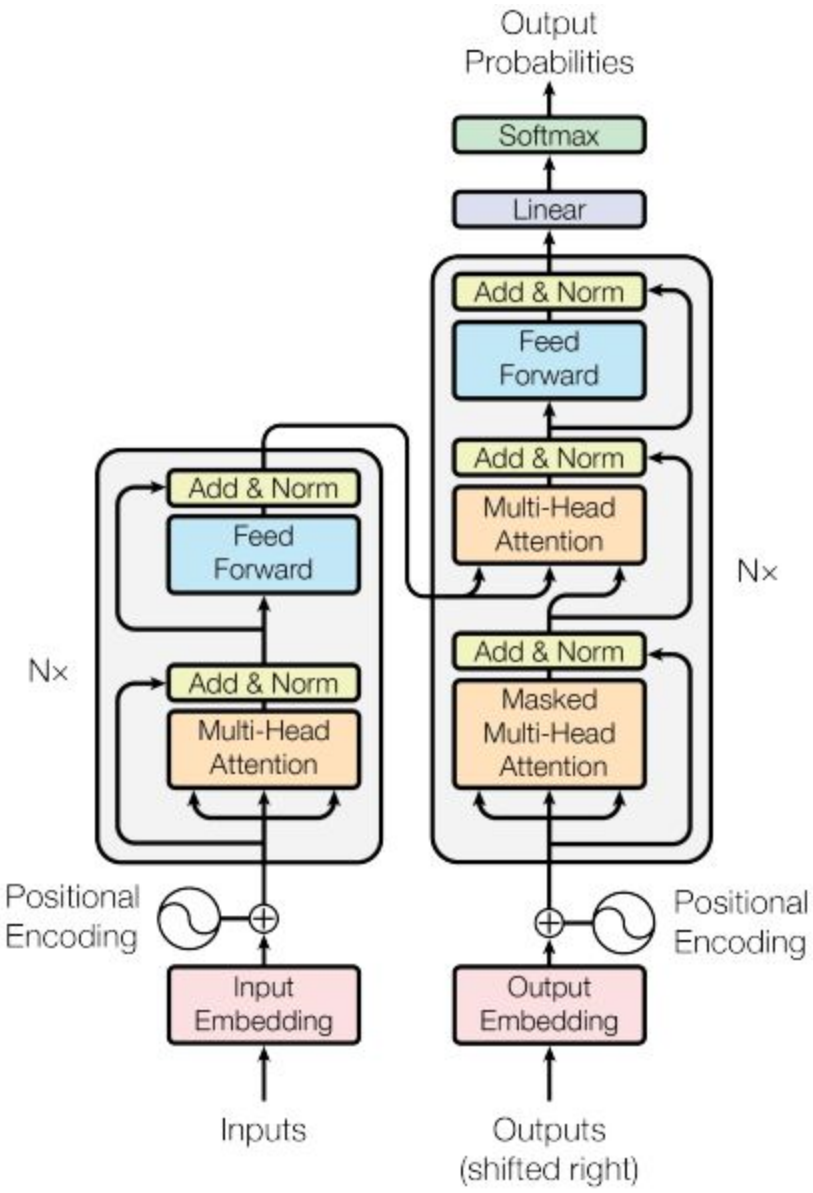


Figura 3: Transformer model

La implementación de transformadores utilizada en este proyecto es idéntica a la mostrada en el diagrama. Se considera que no eran necesarios realizar cambios y que un modelo de seq2seq de transformadores con una máscara en el decodificador era suficiente para completar la tarea planteada.

Experimento

Obtención de los datos

Se utilizó la base de datos GEAR.uk que contiene una colección de 87 millones de abstractos indexados con sus respectivas palabras clave. Para este proyecto, no se utilizaron los 87 millones de abstractos debido a limitaciones de hardware. Se realizaron varias solicitudes a la base de datos de GEAR.uk utilizando su API y se obtuvieron 40,000 artículos en temas variados:

```
topics = ['biology','computer AND science','medicine','cancer',
         'anthropology AND sociology','biochemistry','behavioural sciences',
         'history','psychology','archaeology','paleontology','math','physics',
         'artificial AND intelligence','machine AND learning','videogames',
         'histology','mechanics','astronomy','development','economy',
         'statistics','telemedicine','literature','ecology','geology',
         'art','virus AND disease','coronavirus','social AND media',
         'epidemiology'
]
```

Figura 4: Listado de temas utilizados.

Preprocesamiento de datos:

Se agruparon las búsquedas en un solo set de datos en el cual se verificó la presencia de abstracto y keywords. De estos 40,000 solo 37,965 contaban con las características requeridas y estos fueron los utilizados.
Los artículos se tokenizaron utilizando el tokenizador Spacy implementado en python y se agruparon en un set de datos tabular de Torchtext como pares “fuente (SRC)”, “objetivo (TRG)”.

Codificación de los datos:

Para que la red neuronal sea capaz de interpretar texto este debe ser codificado a valores numéricos. La codificaciòn utilizada es una codificación posicional. Es una capa de codificación estándar en donde el input no es el token (palabra) como sí, sino que es la posición del token con respecto a la secuencia, comenzando con el token de inicio de oración <sos> y terminando con el token de finalización de secuencia <eos>. Debido a limitaciones de hardware se utilizaron codificaciones posicionales para oraciones de 100 palabras de largo. Inicialmente se pensó utilizar 250 palabras por ser el largo estándar de un abstracto, pero no fue posible dadas limitantes de memoria.

Experimentos realizados

- **Red neuronal de RNN con GRU y secuencias de 250 palabras:** Se programó una red neuronal recurrente con GRU y se utilizó una tokenización de 250 palabras para probarla.
    - Resultados: Los resultados fueron inconclusos. Una de las principales limitantes de una red neuronal de tipo RNN es su naturaleza secuencial y el elevado costo de memoria que posee.
- **Red neuronal de RNN con GRU y secuencias de 100 palabras:** Se programó una red neuronal recurrente con GRU y se utilizó una tokenización de 100 palabras para probarla.
    - Resultados: Los resultados fueron inconclusos. Una de las principales limitantes de una red neuronal de tipo RNN es su naturaleza secuencial y el elevado costo de memoria que posee.
- **Red neuronal Transformer y secuencias de 250 palabras:** Se programó una red neuronal de tipo transformer como la descrita en el artículo “Attention is all you need” siguiendo el código de ejemplo en el perfil de github de bentrevett[10]. Sin embargo, se trabajó con una codificación posicional en lugar de una codificación estática, se utilizó el optimizador ADAM estándar con un aprendizaje estático y no se utilizó un suavizador de etiquetas.
    - Resultados: Los resultados fueron inconclusos. A pesar de estar utilizando un transformer que utiliza recursos de forma más eficiente y por medio de agrupación por baches es posible entrenarlo en paralelo, no se obtuvo resultados debido a limitantes de hardware.
- **Red neuronal Transformer y secuencias de 100 palabras:** Se programó una red neuronal de tipo transformer como la descrita en el artículo “Attention is all you need” sin embargo, se trabajó con una codificación posicional en lugar de una codificación estática, se utilizó el optimizador ADAM estándar con un aprendizaje estático y no se utilizó un suavizador de etiquetas.
    - Resultados: Se logró entrenar la red neuronal de forma exitosa y se obtuvieron resultados favorables. Esta red neuronal será la descrita de aquí en adelante.

Resultados Cualitativos de Transformer con secuencias de 100 palabras

Resultados Positivos:

```
src = ['order', 'statistics', 'and', 'linear', 'functions', 'of', 'order', 'st
atistics', 'are', 'frequently', 'used', 'as', 'estimators', 'of', 'location',
'and', 'scale', '.', 'the', 'moments', 'of', 'order', 'statistics', 'have', 'b
een', 'tabled', 'for', 'many', 'commonly', 'used', 'distributions', 'under',
'assumptions', 'of', 'independence', 'and', 'identical', 'distribution', '.',
'applications', 'of', 'estimators', 'based', 'on', 'the', 'order', 'statistic
s', 'have', 'spread', 'to', 'situations', 'in', 'which', 'identical', 'distrib
ution', 'can', 'not', 'be', 'assumed', ',', 'even', 'though', 'independence',
'is', 'still', 'a', 'sensible', 'assumption', '.', 'as', 'an', 'example', ',',
'estimators', 'based', 'on', 'order', 'statistics', 'are', 'used', 'in', 'th
e', 'field', 'of', 'communications', 'where', 'the', 'median', 'and', 'other',
'functions', 'of', 'order', 'statistics', 'of', 'a', 'moving', 'sample']
trg = ['statistics', 'statisticsandprobability']
predicted trg = ['statistics', 'probability/statistics']
```

El primer ejemplo de resultados parece ser muy favorecedor. Se esperaban los keywords “statistics” , “statistics and probability” y se obtuvieron los keywords “statistics”, “probability/statistics”.

```
src = ['in', 'this', 'paper', 'we', 'propose', 'a', 'new', 'supersymmetric',
'extension', 'of', 'conformal', 'mechanics', '.', 'the', 'grassmannian', 'vari
ables', 'that', 'we', 'introduce', 'are', 'the', 'basis', 'of', 'the', 'form
s', 'and', 'of', 'the', 'vector', '-', 'fields', 'built', 'over', 'the', 'symp
lectic', 'space', 'of', 'the', 'original', 'system', '.', 'our', 'supersymmetr
ic', 'hamiltonian', 'itself', 'turns', 'out', 'to', 'have', 'a', 'clear', 'geo
metrical', 'meaning', 'being', 'the', 'lie', '-', 'derivative', 'of', 'the',
'hamiltonian', 'flow', 'of', 'conformal', 'mechanics', '.', 'using', 'superfie
lds', 'we', 'derive', 'a', 'constraint', 'which', 'gives', 'the', 'exact', 'so
lution', 'of', 'the', 'supersymmetric', 'system', 'in', 'a', 'way', 'analogou
s', 'to', 'the', 'constraint', 'in', 'configuration', 'space', 'which', 'solve
d', 'the', 'original', 'non', '-', 'supersymmetric']
trg = ['particlephysics-theory']
predicted trg = ['particlephysics-theory']
```

El siguiente artículo evaluado habla sobre física de partículas y los resultados fueron exactos en el predicho.

Resultados Negativos

```
src = ['background', ':', 'the', 'shortage', 'of', 'medical', 'providers', 'i
n', 'rural', 'areas', 'is', 'one', 'of', 'the', 'greatest', 'barriers', 'to',
'accessing', 'healthcare', 'in', 'the', 'united', 'states', '.', 'mental', 'he
althcare', 'providers', 'are', 'especially', 'limited', 'in', 'numbers', 'an
d', 'availability', 'in', 'rural', 'areas', '.', 'although', 'all', 'individua
ls', 'living', 'rurally', 'have', 'challenges', 'with', 'receiving', 'care',
',', 'one', 'group', 'that', 'demonstrates', 'an', 'exceptional', 'deficienc
y', 'with', 'access', 'to', 'care', 'is', 'veterans', '.', 'the', 'numbers',
'of', 'veterans', 'living', 'rurally', 'are', 'significantly', 'higher', 'tha
n', 'vets', 'living', 'close', 'to', 'urban', 'or', 'suburban', 'centers', 'o
f', 'care', '.', 'considering', 'the', 'shortages', 'of', 'mental', 'health',
'providers', ',', 'alternatives', 'to', 'traditional', 'in', '-', 'person']
trg = ['mentalhealth', 'telemedicine', 'videoteleconferencing', 'veterans', 'm
edicineandhealthsciences']
predicted trg = ['healthcare', 'healthcare', 'healthcare', 'healthcare', 'heal
thcare', 'health', 'health', 'health', 'health', 'health', 'health', 'medicale
ducation']
```

Como se puede observar, el artículo habla sobre telemedicina y alternativas a medicina tradicional en persona. Sin embargo, la red neuronal, a pesar de que entiende que es un tema de salud, no es capáz de reconocer los matices del tema específico y produce muchos keywords repetidos sobre salud y cuidados de salud. No necesariamente se equivoca, pero no es lo suficientemente precisa para obtener los detalles de la clasificación.

```
src = ['this', 'paper', 'deals', 'with', 'the', 'possible', 'benefits', 'of',
'perceptual', 'learning', 'in', 'artificial', 'intelligence', '.', 'on', 'th
e', 'one', 'hand', ',', 'perceptual', 'learning', 'is', 'more', 'and', 'more',
'studied', 'in', 'neurobiology', 'and', 'is', 'now', 'considered', 'as', 'an',
'essential', 'part', 'of', 'any', 'living', 'system', '.', 'in', 'fact', ',',
'perceptual', 'learning', 'and', 'cognitive', 'learning', 'are', 'both', 'nece
ssary', 'for', 'learning', 'and', 'often', 'depends', 'on', 'each', 'other',
'.', 'on', 'the', 'other', 'hand', ',', 'many', 'works', 'in', 'machine', 'lea
rning', 'are', 'concerned', 'with', '"', 'abstraction', '"', 'in', 'order', 't
o', 'reduce', 'the', 'amount', 'of', 'complexity', 'related', 'to', 'some', 'l
earning', 'tasks', '.', 'in', 'the', 'abstraction', 'framework', ',', 'percept
ual', 'learning']
trg = ['artificialvision', 'reformulation', 'abstraction', 'machinelearning',
'autonomousrobotics', 'artificialvision.', 'info.info-aicomputersciencecs/arti
ficialintelligencecs.ai']
predicted trg = ['artificialintelligence', 'artificialintelligence', 'machinel
earning', 'artificialintelligence', 'artificialintelligence', 'machinelearnin
g', 'artificialintelligence', 'machinelearning']
```

Nuevamente ocurre el mismo problema que anteriormente con la especificidad de los resultados.

Resultados interesantes

Por cuestiones de tiempo no se validó que los artículos presentes fuesen completamente artículos en inglés, es posible que algunos abstractos se encontraran en otro idioma. Como es el caso del ejemplo a presentar.

```
src = ['si', 'bien', 'vivimos', 'en', 'una', 'sociedad', 'global', ',', 'los',
'educadores', 'se', 'enfrentan', 'a', 'numerosos', 'desafíos', 'a', 'la', 'hor
a', 'de', 'hallar', 'formas', 'significativas', '\n', 'de', 'conectar', 'a',
'los', 'alumnos', 'con', 'gente', 'de', 'otras', 'culturas', '.', 'este', 'art
ículo', 'muestra', 'un', 'caso', 'práctico', 'de', 'colaboración', 'entre', 'p
rofesores', '\n', 'de', 'los', 'estados', 'unidos', 'y', 'turquía', ',', 'en',
'el', 'que', 'alumnos', 'de', 'séptimo', 'grado', 'interactuaron', 'entre', 's
í', 'a', 'través', 'de', 'las', 'redes', 'sociales', 'con', 'el', '\n', 'fin',
'de', 'promover', 'la', 'comprensión', 'cultural', '.', 'al', 'analizar', 'un
a', 'única', 'actividad', 'de', 'aprendizaje', 'hallamos', 'que', 'los', 'alum
nos', 'tenían', 'la', 'oportunidad', '\n', 'de', 'compartir', 'ideas', 'inform
almente', 'a']
trg = ['mediadigitales', 'mediossociales', 'middleschool', 'popularculture',
'cross-cultural', 'digitalmedia', 'socialmedia', 'alfabetización', 'educacións
ecundaria', 'culturapopular', 'intercultural']
predicted trg = ['educación', 'educación', 'cultural', 'teaching', 'social',
'cultural', 'social', 'socialandculturalanthropology']
```

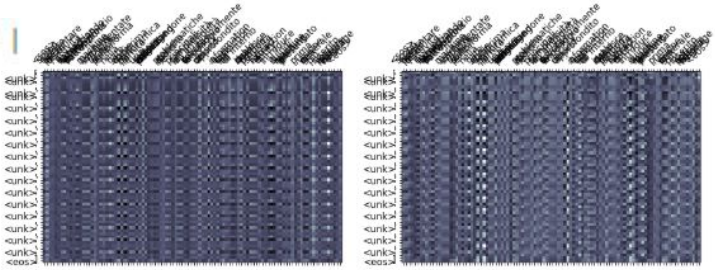
Es fascinante que la red neural fuese capaz de producir keywords en español y en inglés acertadas para el artículo.

Resultados cuantitativos:

Se obtuvo un valor de perplejidad de 6.604. Este valor es bastante bajo y parece que no se sobreentreno la máquina ya que se producen resultados con muchos tokens de <unk> de palabras de vocabulario no disponibles al momento de evaluar. Como se puede observar en el siguiente ejemplo cualitativo.

```
src = ['scopo', 'di', 'questo', 'testo', 'è', 'presentare', 'l’attività', 'd
i', 'named', 'entity', 'extraction', ',', 'focalizzando', 'l’attenzione', 'sul
l’approccio', 'machine', 'learning', 'e', 'sulle', 'tecniche', 'implementate',
'nella', 'piattaforma', 'gate', '/', 'annie', '.', 'è', 'dapprima', 'fornita',
'una', 'panoramica', 'del', 'settore', 'natural', 'language', 'processing',
',', 'analizzandone', 'le', 'radici', 'storiche', ',', 'le', 'problematiche',
'principali', 'e', 'la', 'complessità', 'dovuta', 'all’ambiguità', 'del', 'lin
guaggio', 'umano', '.', 'successivamente', 'è', 'approfondito', 'il', 'task',
'di', 'information', 'extraction', ',', 'all’interno', 'del', 'quale', 'si',
'inserisce', 'proprio', 'l’attività', 'di', 'named', 'entity', 'extraction',
'.', 'si', 'introduce', 'quindi', 'il', 'tema', 'machine', 'learning', ',', 'p
resentato', 'prima', 'in', 'una', 'visione', 'più', 'generale', 'e', 'poi', 'i
nserito', 'nel', 'settore', 'natural', 'language']
trg = ['naturallanguageprocessing', 'informationextraction', 'namedentityextra
ction', 'machinelearning', 'gate', 'ingegneriaescienzeinformatichel-dm270-cese
na']
predicted trg = []
```

```
| Test Loss: 1.888 | Test PPL:   6.604 |
```



Análisis

Tamaño de entradas

Como se puede observar el modelo construido parece ser bastante bueno para las tareas, sin embargo, es claro notar que si se hubiese logrado entrenar el modelo con 250 palabras el resultado sería mucho mejor. En este momento los inputs están truncados a más de la mitad y se pierde mucho del contexto del artículo ya que no se puede elegir de forma más precisa cuales son las 100 palabras que se toman de la tokenización.

Tokenización

Los valores de perplejidad y loss obtenidos son muy bajos. Considero que no representan realmente a la eficiencia de la red neuronal, luego de analizar los resultados obtenidos de la traducción, se observó que la tokenización de salida no estaba depurada de símbolos como comillas, paréntesis y otros, esto puede dar la indicación de que los valores de perplejidad tan bajos ocurrieran ya que el sistema aprendió la estructura de una lista de python convertida a String y era capaz de colocar los tokens correspondientes a los valores de salida de forma correcta.
Sin embargo, es probable que este error también ayudó a los resultados ya que la presencia de tokens adicionales redujo la diferencia entre largos de input y output y facilitó el trabajo del enmascaramiento de outputs por el decodificador. Estudios previos muestran que los lenguajes uno a uno son más fáciles de traducir, entonces aumentar la correspondencia 1 a 1 entre input y output puede haber sido beneficioso para el resultado obtenido.

Discusión de resultados

A pesar de los comentarios que podrían justificar un sobre ajuste de la red neuronal los resultados son en su mayoría buenos, a pesar de que como se evaluó en los resultados malos, estos no contienen los matices de nombramiento de keywords que podrían tener, son exitosos en categorizar texto en el tema relacionado al mismo y parece ser que los ejemplos en los que realiza esta tarea correctamente son más que en los que no.
Los resultados parecen indicar que los modelos de transformadores para seq2seq son efectivos para realizar tareas de clasificación con categorías no predefinidas, a pesar de que estos sistemas usualmente se realizan con modelos de clasificadores o clustering, los resultados de una tarea de traducción de inglés a inglés simplificado son muy alentadores para estudios futuros de clasificadores [7, 8]. Las tareas de sumarización pueden solucionar problemas propuestos en procesamiento de lenguaje natural y la capacidad para sumarizar con

una red neuronal diseñada originalmente para la traducción de francés a inglés es un descubrimiento de suma importancia para el ámbito.

Conclusiones:

La implementación de una red neuronal de transformadores permitió que con recursos limitados se entrenara un modelo relativamente exitoso en la tarea propuesta. Es posible que con mejor hardware se obtengan resultados mucho más precisos.
Los modelos de seq2seq son un acercamiento válido para la categorización de texto, esto podría ser visto como una tarea de traducción y sumarización.
Parece ser que un modelo poco genérico es capaz de categorizar varios idiomas a la vez sin necesidad de realizar codificaciones diferentes para cada idioma, esto presenta nuevos acercamientos a los problemas de traducción ya que los vectores de entrenamiento podrían ser mucho más complejos que solo un idioma y servir para realizar traducciones de una forma más generalizable.

Referencias Bibliográficas

[1] Noorden, R. Van 2014. Elsevier opens its papers to text-mining. Nature. 506, 17 (2014), 22.
[2] Wartena, C. et al. 2010. Keyword extraction using word co-occurrence. Proceedings - 21st International Workshop on Database and Expert Systems Applications, DEXA 2010. October (2010), 54–58. DOI:https://doi.org/10.1109/DEXA.2010.32.
[3] Ekkekakis, P., Hartman, M. E., & Ladwig, M. A. (2018). Mass media representations of the evidence as a possible deterrent to recommending exercise for the treatment of depression: Lessons five years after the extraordinary case of TREAD-UK. Journal of Sports Sciences, 36(16), 1860–1871. https://doi.org/10.1080/02640414.2018.142385
[4] Moutinho, L. (2007). Article information : Abstract and Keywords. European Journal of Marketing (Vol. 21, pp. 5–44).
[5] Jozefowicz, R., Zaremba, W., & Sutskever, I. (2015). An empirical exploration of Recurrent Network architectures. In 32nd International Conference on Machine Learning, ICML 2015 (Vol. 3, pp. 2332–2340). International Machine Learning Society (IMLS).
[6] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A, Kaiser Ł, Polosukhin I Advances in Neural Information Processing Systems, vol. 2017-December (2017) pp. 5999-6009 Published by Neural information processing systems foundation
[7] Ramachandran, P., Liu, P. J., & Le, Q. V. (2017). Unsupervised pretraining for sequence to sequence learning. In EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings (pp. 383–391). Association for Computational Linguistics (ACL). https://doi.org/10.18653/v1/d17-1039
[8] Zhang, J. X., Ling, Z. H., Jiang, Y., Liu, L. J., Liang, C., & Dai, L. R. (2019). Improving Sequence-to-sequence Voice Conversion by Adding Text-supervision. In ICASSP, IEEE

International Conference on Acoustics, Speech and Signal Processing - Proceedings (Vol. 2019-May, pp. 6785–6789). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ICASSP.2019.8682380
[10] bentrevett/pytorch-seq2seq: Tutorials on implementing a few sequence-to-sequence (seq2seq) models with PyTorch and TorchText.

Anexos:

Enlace a Repositorio en GitHub
https://github.com/Rafalp190/nlp-keyword-generator