Extracción de resúmenes

Cálamo Educación S.L.
María Teresa Tijeras Pascual
Cálamo Educación S.L.
María Teresa Tijeras Pascual

Indice

Extracción de resúmenes

El objetivo de esta unidad es presentar una panorámica de las tecnologías de generación de resúmenes automáticos.

0. Objetivos de la unidad

El objetivo de esta unidad es presentar una panorámica de las tecnologías de generación de resúmenes automáticos. Un resumen puede definirse como una versión reducida o simplificada de un texto que mantiene un contenido semántico suficiente para su comprensión. Existen dos tipos de resúmenes: los resúmenes extractivos, que consisten en seleccionar de entre todas las frases del texto aquellas cuyo contenido semántico es más importante, y los resúmenes abstractivos, que incorporan mecanismos complejos de reescritura del texto para generar una versión parafraseada de la reducción semántica. En esta unidad se van a estudiar los fundamentos de ambas técnicas.

En el **apartado 1** se presentan los conceptos más importantes relacionados con esta tarea.

E l **apartado 2** se centra en describir los fundamentos de los resúmenes extractivos, analizando pormenorizadamente el flujo de trabajo de los sistemas clásicos, así como los modelos basados en aprendizaje automático.

A continuación, el **apartado 3** se centra en describir los resúmenes abstractivos.

1. Introducción

Un resumen puede definirse como una versión abreviada o simplificada de uno o varios textos, que conserva una parte significativa de la información semántica del texto o los textos originales. El resumen permite comprender la información más importante leyendo menos cantidad de texto, o haciéndolo más sencillo, utilizando una versión simplificada del lenguaje (con menos tecnicismos o un registro menos formal, adaptado al destinatario).

resumir Conjugar

Del lat. *resumĕre* 'retomar', 'reanudar', 'reabsorber'.

1. tr. Reducir a términos breves y precisos, o considerar tan solo y repetir abreviadamente lo esencial de un asunto o materia. U. t. c. prnl.

2. tr. Dicho del actuante: Repetir el silogismo del contrario.

3. prnl. Dicho de una cosa: Convertirse, comprenderse, resolverse en otra.

4. prnl. *C. Rica.* Dicho de un líquido: Filtrarse o absorberse.

Definición de "resumir"
Fuente: *Diccionario de la lengua española*, RAE

Formalmente, la tarea de extracción de resúmenes (o síntesis de textos) puede definirse como el proceso de destilar la información más importante de una fuente (o fuentes) para producir una versión abreviada o simplificada, empleando técnicas de procesamiento del lenguaje natural, para un usuario y una tarea en concreto.

Se trata de un reto complejo, ya que cuando los humanos abordan la tarea de resumir un texto, lo suelen leer por completo para entenderlo y luego escriben un resumen destacando los puntos principales. Dado que los ordenadores carecen de los conocimientos y la capacidad lingüística de los humanos, la síntesis automática de textos se convierte en una tarea muy difícil, nada trivial.

Existen dos tipos de resúmenes: resúmenes extractivos, en los que el sistema selecciona aquellas frases que son más importantes del texto, y resúmenes abstractivos, donde se reescribe (parafrasea) el texto con otras palabras, que no estaban en el texto original, proporcionando un resumen más rico y elaborado. En los siguientes apartados se estudiará en detalle cada uno de ellos.

Aunque tradicionalmente la síntesis de texto se ha centrado en información de tipo textual, la entrada para el proceso de resumen también podría ser información multimedia, como imágenes, vídeo o audio, así como información en línea de diferentes fuentes.

Aparte del interés que tiene como tal obtener una versión más corta de un contenido, el resultado de la extracción del resumen puede ser útil para integrarse con otros sistemas que llevan a cabo otras tareas de procesamiento del lenguaje natural, como los sistemas de extracción de entidades, de clasificación de textos o de respuesta a preguntas. La síntesis textual permite enfocar los resultados de las otras tareas en el contenido que se supone más importante, aumentando así su precisión, además de reducir el tiempo de procesamiento, al funcionar sobre documentos de menor longitud.

La tarea de resumen es posible aplicarla sobre un solo texto o sobre varios, en cuyo caso se denomina resumen de documentos múltiples o resumen multidocumento. El objetivo del resumen multidocumento es extraer información de múltiples textos escritos sobre el mismo tema.

Además, los documentos fuente pueden estar en un mismo idioma (escenario monolingüe) o en diferentes idiomas (escenario multilingüe).

La primera publicación sobre extracción de resúmenes es de 1958 (H.P. Lun, “The automatic creation of literature abstracts”, IBM), donde se utilizaba una técnica estadística llamada diagramas de frecuencia de palabras. Desde entonces, han surgido muchos enfoques diferentes. En 2015 surgió el análisis basado en TF-IDF (frecuencia de términos y frecuencia inversa de documentos). En 2016 se abordó el resumen multidocumento, con una técnica de resumen basada en patrones. Al año siguiente se utilizó la técnica LSA *(latent semantic analysis)* combinada con NMF *(non negative matrix factorization)*.

Hoy en día, aunque no han sustituido a los enfoques anteriores y a menudo se combinan con ellos, los métodos de aprendizaje automático son los más empleados para llevar a cabo el resumen extractivo de documentos individuales. La investigación actual se centra en la generación de resúmenes en tiempo real y el empleo de técnicas de *deep learning* para generar resúmenes abstractivos.

2. Resúmenes extractivos

2.1. Conceptos

Un resumen extractivo es aquel compuesto por la selección ordenada de las frases más importantes del texto, realizada por el sistema.

Los resúmenes se forman 1) asignando una puntuación a cada frase del texto según su importancia semántica y, en algunos algoritmos, el contexto donde aparecen en el texto (por ejemplo, las frases de los títulos o del principio del texto son las más importantes), 2) seleccionando, en segundo lugar, las frases más significativas y, 3) por último, añadiéndolas literalmente al resumen en el mismo orden de aparición.

Estos resúmenes, por tanto, recogen frases exactas del texto original.

En cuanto al tamaño del resumen, suele poderse especificar de manera explícita en el sistema. Algunas configuraciones habituales son elegir: un porcentaje de frases del texto original (por ejemplo, un 25%), un número de frases esperadas (por ejemplo, 4 frases) o el número de palabras esperadas (por ejemplo, 200 palabras). En ocasiones, en vez de dar un valor exacto, se da un intervalo mínimo y máximo (de frases o porcentaje).

Hasta hace poco tiempo, la mayor parte de la investigación en materia de resúmenes se ha centrado en el resumen extractivo, ya que es más sencillo y produce resúmenes gramaticales que requieren relativamente poco análisis lingüístico.

2.2. Proceso clásico

Estos algoritmos se centran en seleccionar las frases más importantes, asignándoles una puntuación *(score)* en función de diferentes indicadores:

1

El foco principal es la cantidad de información que aporta la frase dentro del texto. Las frases que más cantidad de información aporten serán las que se seleccionen para ser parte del resumen.

2

Algunos sistemas tienen en cuenta la presencia en la frase de ciertas palabras o unidades fraseológicas clave (*keywords* o *keyphrases*), que pueden incrementar o disminuir la puntuación asignada a dicha frase. Por ejemplo, una frase que comience por “en resumen” puede ver incrementado su score para asegurarse de que se incluya en el resumen generado.

3

Otros sistemas tienen en cuenta también el contexto de cada frase. Por ejemplo, si una de las frases seleccionadas para el resumen es parte de una enumeración tipo: “a) uno, b) dos, c) tres”, se seleccionan también el resto de ítems de la enumeración para no perder información.

4

Si se tiene información sobre la clase de texto que se quiere resumir (por ejemplo, noticias, documentación técnica, etc.) es posible tomar decisiones específicas que mejoren la salida (por ejemplo, en resúmenes de noticias, la información más importante está al principio del texto, por lo que interesa multiplicar el score semántico de cada frase por un factor que dependa del número de orden de la frase en el texto).

La selección de frases es un enfoque sencillo, que no requiere bases de conocimiento adicionales, como ontologías o modelos lingüísticos.

Además, generalmente es una tarea que suele ser independiente del idioma. Aunque si se conoce el idioma se podría aplicar este conocimiento para optimizar el resumen.

El proceso clásico típico de los sistemas de resumen extractivos consiste en los siguientes pasos:

1. Segmentación del texto en frases

El texto se divide en frases, utilizando una función de segmentación adecuada al texto.

Tras obtener cada frase, para refinar el análisis posterior se suelen aplicar:

- Técnicas de filtrado, para eliminar frases con menos de X palabras o más de X números.
- Técnicas de normalización para pasar el texto a minúsculas o aplicar lematización o *stemming*.

2. Análisis de cada frase

Cada frase se procesa para obtener una representación de su contenido semántico, utilizando una o varias funciones de *scoring*.

Por ejemplo, para una frase, se podría calcular una o varias de las siguientes funciones heurísticas:

- Número de palabras (su longitud).
- Número de palabras con significado (es decir, típicamente: nombres, adjetivos, verbos y adverbios, ignorando las palabras de parada o *stopwords*).
- Número de palabras que están listadas en un vocabulario de palabras "relevantes" en el dominio de aplicación.
- Si tiene o no alguna *keyword* o *keyphrase* (se suelen considerar *keywords* las palabras más frecuentes del texto tras eliminar las *stopwords*).
- Relevancia respecto al título.
- Empleo de funciones avanzadas de cálculo semántico, por ejemplo, TextRank, LexRank, el análisis semántico latente (LSA, *latent semantic analysis*), la técnica de Luhn, KL-Sum o medidas de similitud semántica basadas en vectores de *embeddings*.

TextRank es una técnica de resumen extractivo que se basa en el concepto de que aquellas palabras que aparecen con mayor frecuencia son más significativas y, por lo tanto, las frases que contienen palabras muy frecuentes son importantes.

El algoritmo LexRank se basa en que, si una frase es similar a muchas otras frases del texto, tiene una alta probabilidad de ser importante.

LSA selecciona las frases semánticamente más significativas aplicando la técnica de descomposición en valores singulares (SVD, *singular value decomposition*) a la matriz de frecuencias término-documento del texto que se quiere resumir.

El enfoque del algoritmo de resumen de Luhn se basa en TF-IDF *(Term Frequency-Inverse Document Frequency)* y es útil cuando tanto las palabras poco frecuentes como las altamente frecuentes *(stopwords)* no son significativas.

El algoritmo KL-Sum (suma de Kullback-Lieber) se basa en la medida de la divergencia del vocabulario del texto de entrada comparada con la del vocabulario del resumen. El objetivo del algoritmo es encontrar un conjunto de frases cuya longitud sea inferior a L palabras y cuya distribución de vocabulario sea lo más parecida posible a la del documento fuente.

3. Puntuación de las frases

Asignación de la puntuación a cada frase, en función de la representación anterior, asignando un valor que indica la probabilidad de que sea recogida en el resumen.

Las anteriores funciones heurísticas se combinan (según cada sistema) para determinar la puntuación de cada frase. Cada heurística asigna una puntuación (positiva o negativa) a la frase. Cada heurística se pondera también según su importancia.

4. Generación del resumen

Generación de un resumen basado en las k frases más importantes (es decir, con mayor puntuación).

5. Adaptación del resumen

Además, algunos sistemas realizan un posprocesamiento del resumen generado para mejorar la salida adaptando ciertas expresiones. Por ejemplo, añadiendo un punto final a cada frase seleccionada, si no acaba en signo de puntuación, o eliminando expresiones como "por ello," o "en segundo lugar," que indican un contexto que se puede haber perdido en el resumen. También hay que tener cuidado con las referencias anafóricas o las elipsis, ya que si se mantienen podrían entorpecer el resumen cuando no se explicita la entidad a la que se referencia.

Proceso clásico de generación de un resumen extractivo

El principal inconveniente de estas técnicas de selección de frases para abordar la tarea de generación del resumen es la pérdida de coherencia en el resultado. No obstante, en general, los resúmenes así generados suelen ser suficientemente inteligibles para los lectores humanos y dan una idea bastante buena sobre las ideas fundamentales de un texto.

2.3. Modelos basados en aprendizaje automático

Estos enfoques consisten en abordar la selección de las frases más importantes, como un problema de aprendizaje automático supervisado, que da como resultado la inclusión o no de una frase en el resumen.

Primero se construye un corpus de textos etiquetados manualmente, donde cada ejemplo es un texto para el que se indica cuáles son sus frases más importantes.

Este corpus de entrenamiento se utiliza para entrenar un modelo de aprendizaje supervisado que aprende a detectar qué frases son las más importantes de un texto dado. Puede abordarse como una clasificación binaria, decidiendo entre 1 (incluir la frase en el resumen) o 0 (no incluir la frase en resumen), o bien entrenar un clasificador que prediga un valor numérico para cada frase que indique su score para generar el resumen.

En principio, cualquier algoritmo de aprendizaje supervisado serviría para realizar la clasificación. En particular, se han empleado popularmente el algoritmo de árboles de decisión, Naive Bayes y el algoritmo de inducción de reglas *(rule induction)*. También se han utilizado métodos de ensamblaje *(ensemble)* que combinan las decisiones de varios clasificadores (por ejemplo, por votación).

Recientemente también se han empleado clasificadores basados en *deep learning*, entrenando arquitecturas CNN con *embeddings* y capas de atención a las que se enseña qué frases son las más importantes para el modelo.

2.4. Escenarios específicos

En ocasiones, se utilizan sistemas de generación de resúmenes para escenarios específicos, donde el texto que se pretende resumir no es plano, sino que incorpora información textual de cierta complejidad. En este caso, suele ser necesario abordar el problema con modelos híbridos, combinando varios enfoques, y empleando los resultados de otras tareas de procesamiento del lenguaje natural como el reconocimiento de entidades o la clasificación automática, ya que cada uno de los enfoques por separado no permite llevar a cabo la tarea con suficiente calidad.

Por ejemplo, supongamos un escenario de generación de resúmenes de transcripciones de llamadas a un *contact center*. La información de partida es la serie de intervenciones intercaladas entre un agente y el cliente que llama, y el objetivo es resumir la evolución de dicha llamada y su resolución.

Agente: Buenos días. Le atiende Susana. ¿En qué puedo ayudarle?

Cliente: Buenos días. Llamaba para consultar un problema con mi factura.

Agente: Por supuesto. ¿De la última factura?

Cliente: Sí, del mes de agosto.

Agente: Compruebo que se ha hecho un cargo adicional de 25€ por servicios, pero al tener contratada la tarifa plana, parece que no es correcto.

Cliente: Eso creo yo, gracias.

Agente: Voy a anular ese cargo y hacer que se genere una factura rectificativa.

Cliente: ¿Me devuelven el dinero?

Agente: Sí, se hará un abono en su cuenta en el plazo de 10 días.

Cliente: ¡Muchas gracias!

Agente: Muchas gracias a usted por ponerse en contacto con nosotros. ¿Alguna cosa más?

Cliente: Nada más. ¡Gracias de nuevo y buenos días!

Para resumir esta interacción, un sistema debería seleccionar el motivo de la llamada (interacción por parte del cliente), las diferentes acciones intermedias (por parte del agente) y la resolución final (por parte del agente). Un ejemplo de resumen sería:

(Cliente) Llamaba para consultar un problema con mi factura. (Agente) Compruebo que se ha hecho un cargo adicional de 25€ por servicios, pero al tener contratada la tarifa plana, parece que no es correcto. Voy a anular ese cargo y hacer que se genere una factura rectificativa.

Este texto contiene información suficiente para comprender el motivo de la llamada y su resolución.

Para obtener este resumen, un sistema debe incluir los siguientes componentes:

1
Transcripción de voz a texto.

2

Diarización del locutor, es decir, identificar quién es el agente y quién es el cliente, separando sus intervenciones.

3

Clasificación de las intervenciones (en inglés, *utterances*) (con modelo de reglas o basado en aprendizaje automático) para saber si se habla del motivo de la llamada, de petición de información, de la solución que se propone, de saludos y/o despedidas, etc.

4

Selección de las intervenciones importantes según el tipo de intervención y los valores de *score* semántico de las frases (ejemplo: seleccionar al menos un motivo, opcionalmente la petición de información intermedia y al menos una solución).

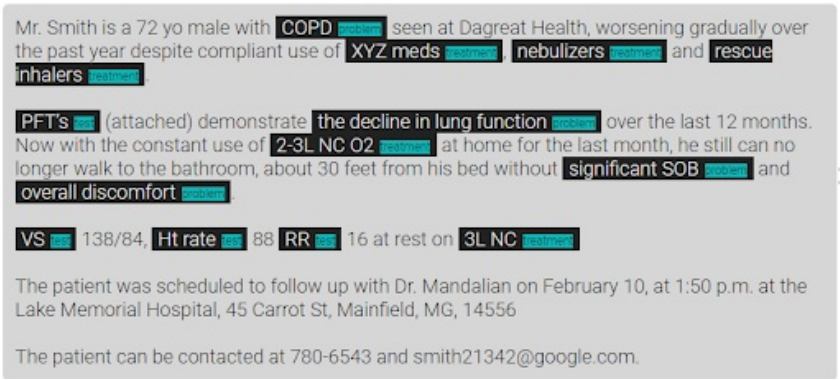
5

Generación del resumen, postprocesando algunas intervenciones para hacer más fluido el resumen (por ejemplo: "Llamaba para consultar un problema con mi factura" → "EL CLIENTE LLAMA para consultar un problema con LA factura").

Es evidente que, en este caso, lo más apropiado sería reescribir el texto para generar el resumen, como se estudiará más adelante, en el apartado de resúmenes abstractivos:

El cliente llama para consultar un problema con su factura, con un cargo adicional por servicios que no es correcto al tener tarifa plana. Se anula el cargo y se genera factura rectificativa.

Otro ejemplo de escenario específico es la generación de resúmenes de historiales clínicos.



Ejemplo de etiquetado de historiales clínicos
Fuente de la imagen: Konplik

2.5. Tarea del lingüista

Las tareas del lingüista en estos proyectos implican habitualmente:

1
Generación de conjuntos de evaluación del sistema: colección de textos con el resumen esperado. En esta tarea es especialmente difícil la creación de corpus de evaluación, ya que dicho corpus dependerá de la longitud del texto original y de la longitud esperada del resumen, de la definición de los criterios de evaluación, etc.
2
Definición de las métricas para evaluación. De nuevo, es una tarea difícil, ya que depende de los parámetros del sistema. Una posible métrica sería comparar las frases obtenidas con las frases esperadas, seleccionadas de forma manual, como si se tratara de un problema de clasificación *multi-label*.
3
Colaborar en la definición de los valores de los parámetros del sistema, según el escenario de aplicación, donde intervengan decisiones relacionadas con el procesamiento lingüístico o los criterios de evaluación.
4
Llevar a cabo las evaluaciones del sistema, analizar los resultados y definir estrategias de mejora.

3. Resúmenes abstractivos

3.1. Conceptos

Los métodos de resumen abstractivo tienen como objetivo producir resúmenes interpretando el texto mediante técnicas avanzadas de procesamiento del lenguaje natural con el fin de generar una versión corta con las ideas principales, reformulando la información, como suelen hacer los humanos.

Es decir, los sistemas de resumen abstractivo no se limitan a seleccionar las frases más importantes del texto original y presentarlas tal cual, por orden de aparición, sino que generan nuevas frases, reformulando o utilizando palabras que no tienen por qué estar en el texto original.

Un resumen abstractivo debería cubrir la información esencial del texto de entrada y ser lingüísticamente fluido.

Es una tarea de gran dificultad, ya que la calidad del resumen requiere capacidades complejas como la paráfrasis, la generalización o la incorporación de conocimiento del mundo real. Por ello, a pesar de los recientes avances empleando las técnicas descritas en el siguiente apartado, aún se está lejos de alcanzar la calidad del nivel humano en esta tarea.

3.2. Modelos

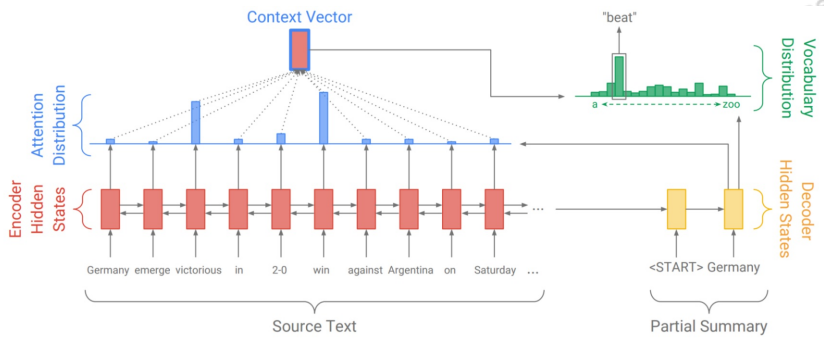
Los métodos abstractivos aprovechan los avances más recientes en el campo de *deep learning.*

El enfoque más empleado actualmente son los modelos de secuencia-a-secuencia (Seq2Seq), con una arquitectura y soluciones similares a los empleados en traducción automática, ya que se basan en considerar que el resumen abstractivo consiste en una traducción de una secuencia de origen (texto original) en otra secuencia destino (el texto del resumen).

Como se describió en la unidad 1, estos modelos consisten en una arquitectura compleja de redes neuronales con un codificador *(encoder)*, que lee el texto original y lo codifica, y un decodificador *(decoder)*, que genera el texto objetivo.

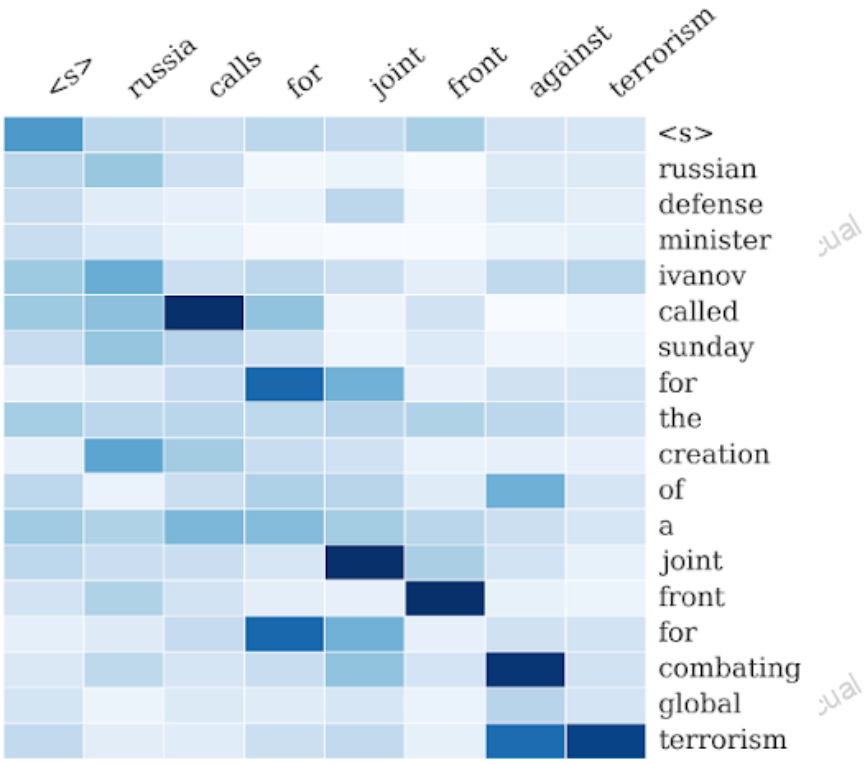
En estos algoritmos, el mecanismo de atención se vuelve esencial porque permite a la red neuronal reconocer y centrarse en las frases más importantes del texto de partida.

La figura siguiente muestra la típica arquitectura *encoder-decoder* para generar el resumen abstractivo. A la frase de entrada "Germany emerge victorious in 2-0 win against Argentina on Saturday [...]", tras su codificación por el encoder, pasaría al decoder que generaría la salida "Germany beats Argentina [...]".



Ejemplo de resumen abstractivo con modelo de secuencia-a-secuencia

Este mecanismo de atención permite centrarse en las palabras más importantes, que previsiblemente estarán presentes en el resumen generado. La figura siguiente muestra la similitud semántica entre cada palabra del resumen generado ("Russia calls for joint front against terrorism") con cada palabra del texto original ("Russian Defense Minister Ivanov called Sunday for the creation of a joint front for combating global terrorism"). Esta tabla permite intuir qué palabras del texto original han conducido a qué palabras del resumen.

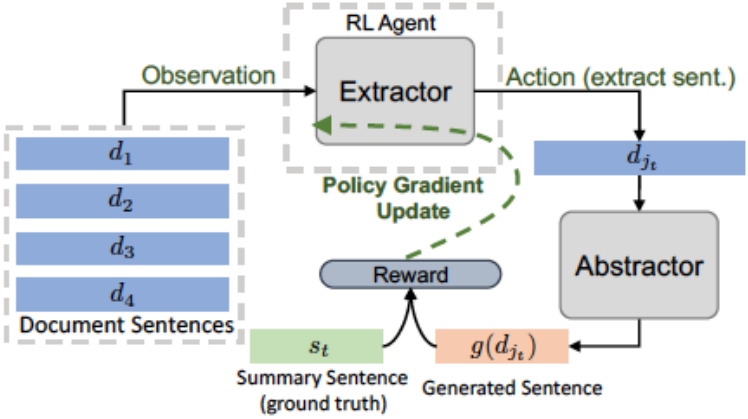


Análisis de la similitud semántica entre el texto original y el resumen

Otras técnicas empleadas hoy en día se basan en aprendizaje por refuerzo y procesos adversariales para mejorar la precisión del modelo.

La siguiente figura muestra el empleo de dos redes neuronales en una arquitectura híbrida extractiva-abstractiva que simula la forma en que los humanos resumimos los textos largos. En este modelo, una de las redes neuronales se utiliza para seleccionar las frases más significativas del texto a través de un extractor basado en aprendizaje por refuerzo y, luego, se emplea la otra red neuronal basada en arquitectura *encoder-decoder* para reescribir cada una de las frases extraídas.

Para enseñar al modelo a reconocer las frases más importantes del texto, se utiliza aprendizaje por refuerzo basado en recompensas, si la frase seleccionada por la red está realmente entre las que seleccionaría el humano.



Arquitectura híbrida con aprendizaje por refuerzo

3.3. Tarea del lingüista

Las tareas del lingüista son similares a las tareas de los proyectos con resúmenes extractivos, con la salvedad de que, al tratarse de una tarea más compleja (en realidad todavía sin resolver, casi en la línea de la investigación aplicada), en este caso los sistemas suelen tener (hoy por hoy) menos componentes de carácter lingüístico y más desarrollos técnicos de arquitectura de redes neuronales profundas, con lo que se necesita una mayor experiencia en estas áreas.

En cualquier caso, el papel del lingüista en estos proyectos sigue siendo esencial para la generación de conjuntos de evaluación del sistema (colección de textos y el resumen esperado) y la definición de las métricas de evaluación, así como su participación en la definición de los valores de los parámetros del sistema (hasta donde sea posible, por su especial complejidad), llevar a cabo la propia evaluación (analizando los resultados) y proponer estrategias de mejora del motor.

RESUMEN

En esta unidad se ha presentado una panorámica de las **tecnologías de generación de resúmenes automáticos**.

Un resumen es una versión abreviada o simplificada de uno o varios textos que conserva una parte significativa de la información semántica original.

Existen dos tipos de resúmenes: los **resúmenes extractivos**, que consisten en seleccionar de entre todas las frases del texto aquellas cuyo contenido semántico es más importante, y los **resúmenes abstractivos**, que incorporan mecanismos complejos de reescritura o paráfrasis del texto para generar esa versión reducida.

Los algoritmos de resúmenes extractivos mediante enfoques clásicos se centran en **seleccionar las frases más importantes del texto**, asignándoles una puntuación *(score)* con diferentes técnicas. Entre los algoritmos más populares están **TextRank, LexRank y LSA**.

Otros enfoques consisten en abordar la selección de las frases más importantes como un **problema de aprendizaje automático supervisado** que da como resultado la inclusión o no de una frase en el resumen. Se construye un corpus de textos etiquetados manualmente, donde cada ejemplo es un texto para el que se indica cuáles son sus frases más importantes, y se utiliza para entrenar un modelo de clasificación, con cualquier algoritmo de aprendizaje supervisado.

Los sistemas de resumen abstractivo no se limitan a seleccionar las frases más importantes del texto original, sino que **generan nuevas frases, reformulando o utilizando palabras** que no tienen por qué estar en el texto original. Es una tarea de gran dificultad, ya que la calidad del resumen requiere capacidades complejas como la paráfrasis, la generalización o la incorporación de conocimiento del mundo real.

Actualmente, el enfoque más empleado para resúmenes abstractivos son los **modelos de secuencia-a-secuencia**, con una arquitectura similar a la utilizada en traducción automática, ya que se basan en considerar que el resumen abstractivo consiste en una traducción de una secuencia de origen (texto original) en otra secuencia destino (el texto del resumen).

Ejercicios

Ejercicio 6: Extracción de resúmenes extractivos y abstractivos

Duración estimada del ejercicio

50 minutos

El objetivo de este ejercicio es abordar diferentes ejemplos de extracción de resúmenes de tipo extractivo y abstractivo, utilizando código Python en el entorno de Google Colaboratory.

Crea un cuaderno de Colab nuevo, activando el entorno de ejecución acelerado con GPU, como ya hiciste en otros ejercicios.

1. Resúmenes extractivos

Primero vamos a definir una variable con el texto que queremos resumir. Copia el siguiente código Python en una nueva celda de código y ejecútalo pulsando el icono de “Reproducir”.

Contiene la entradilla al artículo de Wikipedia sobre Pablo Picasso.

```
text = '''

Pablo Ruiz Picasso (Málaga, 25 de octubre de 1881–Mougins, 8 de abril de 1973) fue un pintor y escultor español, creador, junto con Georges Braque, del cubismo. Es considerado desde la génesis del siglo XX como uno de los mayores pintores que participaron en los variados movimientos artísticos que se propagaron por el mundo y ejercieron una gran influencia en otros grandes artistas de su tiempo. Sus trabajos están presentes en museos y colecciones de toda Europa y del mundo. Además, abordó otros géneros como el dibujo, el grabado, la ilustración de libros, la escultura, la cerámica y el diseño de escenografía y vestuario para montajes teatrales. También tiene una breve obra literaria. En lo político, Picasso se declaraba pacifista y comunista. Fue miembro del Partido Comunista de España y del Partido Comunista Francés hasta su muerte, acaecida el 8 de abril de 1973 a los noventa y un años de edad, en su casa llamada «Notre-Dame-de-Vie» de la localidad francesa de Mougins. Está enterrado en el parque del castillo de Vauvenargues (Bouches-du-Rhone).

'''
```

NOTA: En una versión anterior de estos materiales, se empleaba un paquete de Python llamado Gensim, que ofrece funcionalidad para algunas tareas de procesamiento del lenguaje natural como generación de *embeddings* y modelado de temas *(topic modeling)*. Para saber más sobre Gensim, puedes pinchar en este enlace. En este ejercicio se utilizaba para hacer extracción de resúmenes empleando la técnica de TextRank, que, como estudiamos, una de las técnicas de resumen extractivo más populares. El código Python empleado era el siguiente:

```
import gensim

from gensim.summarization import summarize


summary = summarize(text, ratio=0.2)

print(summary)
```

Como resumen solo se seleccionaba una frase, la segunda del texto:

Es considerado desde la génesis del siglo XX como uno de los mayores pintores que participaron en los variados movimientos artísticos que se propagaron por el mundo y ejercieron una gran influencia en otros grandes artistas de su tiempo.

Sin embargo, esta funcionalidad ha sido eliminada en las versiones más recientes de Gensim, con lo que hemos tenido que eliminar esta sección del ejercicio.

Para hacer resúmenes con diferentes técnicas se puede usar también el paquete Python llamado Sumy, que implementa numerosos algoritmos.

Para instalar las dependencias, primero ejecuta el siguiente código en una celda de código:

```
!pip install sumy
```

Y luego, en una celda diferente, copia y ejecuta el siguiente código:

```
import sumy

import nltk

nltk.download('punkt')


from sumy.parsers.plaintext import PlaintextParser

from sumy.nlp.tokenizers import Tokenizer

parser = PlaintextParser.from_string(text, Tokenizer('spanish'))
```

Por ejemplo, para obtener un resumen con la técnica TextRank:

```
from sumy.summarizers.text_rank import TextRankSummarizer


summary = TextRankSummarizer()(parser.document, sentences_count=1)

for sentence in summary:

    print(sentence)
```

El algoritmo selecciona una única frase (como se indica en el parámetro de la función), que no parece generar un buen resumen del texto en este caso:

Fue miembro del Partido Comunista de España y del Partido Comunista Francés hasta su muerte, acaecida el 8 de abril de 1973 a los noventa y un años de edad, en su casa llamada «Notre-Dame-de-Vie» de la localidad francesa de Mougins.

De manera similar, por ejemplo, para obtener un resumen con la técnica LexRank:

```
from sumy.summarizers.lex_rank import LexRankSummarizer

summary = LexRankSummarizer()(parser.document, sentences_count=1)

for sentence in summary:

  print(sentence)
```

En este caso, el algoritmo selecciona la primera frase del texto, que tiene un cierto sentido:

Pablo Ruiz Picasso (Málaga, 25 de octubre de 1881–Mougins, 8 de abril de 1973) fue un pintor y escultor español, creador, junto con Georges Braque, del cubismo.

También se puede utilizar el algoritmo LSA:

```
from sumy.summarizers.lsa import LsaSummarizer

summary = LsaSummarizer()(parser.document, sentences_count=1)

for sentence in summary:

  print(sentence)
```

Es considerado desde la génesis del siglo XX como uno de los mayores pintores que participaron en los variados movimientos artísticos que se propagaron por el mundo y ejercieron una gran influencia en otros grandes artistas de su tiempo.

Otra técnica es la de Luhn, que genera el mismo resumen que TextRank:

```
from sumy.summarizers.luhn import LuhnSummarizer

summary = LuhnSummarizer()(parser.document, sentences_count=1)

for sentence in summary:

    print(sentence)
```

Fue miembro del Partido Comunista de España y del Partido Comunista Francés hasta su muerte, acaecida el 8 de abril de 1973 a los noventa y un años de edad, en su casa llamada «Notre-Dame-de-Vie» de la localidad francesa de Mougins.

Por último, empleando la técnica de KL-Sum se obtienen los resultados siguientes:

```
from sumy.summarizers.kl import KLSummarizer

summary = KLSummarizer()(parser.document, sentences_count=2)

for sentence in summary:

    print(sentence)
```

Pablo Ruiz Picasso (Málaga, 25 de octubre de 1881–Mougins, 8 de abril de 1973) fue un pintor y escultor español, creador, junto con Georges Braque, del cubismo.

Sus trabajos están presentes en museos y colecciones de toda Europa y del mundo.

Compara las diferentes técnicas e intenta determinar cuál genera el mejor resumen para el texto dado.

Además, puedes probar con otros textos, por ejemplo, una noticia de prensa. TextRank es una técnica independiente del idioma, así que puedes elegir textos en otro idioma, por ejemplo, inglés. Para el resto de algoritmos, lo único que cambia es la segmentación del texto, así que elige otro idioma para el *tokenizer*, por ejemplo:

```
parser = PlaintextParser.from_string(text, Tokenizer('english'))
```

2. Resúmenes abstractivos

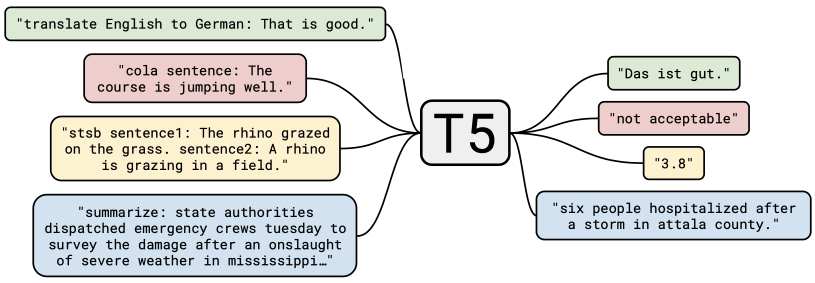
Para realizar extracción de resúmenes abstractivos, vamos a emplear técnicas de *deep learning* basadas en transformers.

El primer paso es instalar las dependencias en nuestro sistema de la biblioteca de Python:

```
!pip install transformers
```

```
from transformers import pipeline
```

En vez de entrenar un modelo propio, que requeriría un gran volumen de corpus de entrenamiento y horas y horas de proceso, vamos a utilizar un modelo ya entrenado de Google, llamado T5, por "Text-To-Text Transfer Transformer". Es un modelo multipropósito, preentrenado con un corpus de código abierto llamado Colossal Clean Crawled Corpus (C4), que obtiene resultados excelentes en diferentes tareas de procesamiento del lenguaje natural como la traducción automática, la similitud semántica o la extracción de resúmenes, y es lo suficientemente flexible como para poder ajustarse a otras tareas.



Google T5

Para saber más sobre Google T5, puedes pinchar en este enlace.

Hay diferentes versiones con un número creciente de parámetros. Vamos a utilizar el modelo más reducido ("t5-small", con 60 millones de parámetros) para generar el resumen del texto anterior. Copia y ejecuta el siguiente código en Python.

```
summarizer = pipeline("summarization",
                      model="t5-small")

summary = summarizer(text,
                     max_length=150,
                     min_length=30,
                     do_sample=False)
print(summary)
```

Pincha aquí para acceder al código

```
summarizer = pipeline("summarization",
                      model="t5-small")

summary = summarizer(text,
                     max_length=150,
                     min_length=30,
                     do_sample=False)

print(summary)
```


El resultado es muy sencillo, no hay reescritura, sino una selección de la frase más relevante:

Pablo Ruiz Picasso (Málaga, 25 de octubre de 1881–Mougins, 8 de abril de 1973) fue un pintor y escultor espaol, creador, junto con Georges Braque, del cubismo.

Es un modelo multilingüe, por ello es posible generar resúmenes en español, aunque con sus fallos, por ejemplo, la omisión de la letra “ñ”.

Si se utiliza un modelo mayor (“t5-base”, 220 millones de parámetros), los resultados mejoran. Cambia en el código anterior “t5-small” por “t5-base” y ejecútalo. Los resultados son mejores, hay una reescritura/simplificación de la frase más larga, aunque se observan fallos con la letra “ñ” y los caracteres acentuados, y una curiosa mezcla de palabras en español e inglés.

Pablo Ruiz Picasso was a pintor y escultor espaol, creador, junto con Georges Braque, del cubismo . he was considered desde la génesis del siglo XX como uno de los mayores pintores que participaron en los variados movimientos artsticos .

Otro modelo preentrenado para la traducción es Bart, basado en una arquitectura estándar secuencia-a-secuencia con un codificador bidireccional (como BERT) y un decodificador de izquierda a derecha (como GPT).

Para saber más sobre Bart, puedes pinchar en este enlace.

Cambia en el código anterior poniendo “facebook/bart-large-cnn” como valor del modelo y pulsa ejecutar. El resultado es mejor que el anterior, aunque también hay una mezcla entre español e inglés:

Pablo Ruiz Picasso (Málaga, 25 de octubre de 1881–Mougins, 8 de abril de 1973) fue un pintor y escultor español, creador, junto with Georges Braque, del cubismo. Es considerado desde la génesis del siglo XX como uno of los mayores pintores que participaron en los variados movimientos artísticos.

Prueba, si quieres, con otros textos y otros idiomas, y analiza los resultados obtenidos.

Recursos

Enlaces de Interés

- **Resumir (DRAE)**
  https://dle.rae.es/resumir
- **Google Colaboratory**
  https://colab.research.google.com/
- **Gensim**
  https://radimrehurek.com/project/gensim/
- **Google T5**
  https://github.com/google-research/text-to-text-transfer-transformer
- **Bart**
  https://huggingface.co/transformers/model_doc/bart.html