USANDO EMBEDDINGS PARA ENCONTRAR RESPUESTAS EN DOCUMENTOS

Ignacio Despujol Zabala

Transcripciones

hola en este último vídeo del módulo seis vamos a ver una solución aproximada aún conocido problema de teoría de grafos el problema del viajante y esta solución aproximada la vamos a llamar una solución de peso bajo vamos a enunciar un problema que nos va a permitir motivarlo supongamos que tenemos aquí varias poblaciones que hemos elegido unas cuantas de la provincia de córdoba y suponemos que viajan debe partir de aquí hinojosa del duque y de visitar varias poblaciones que tenemos aquí cuál es el camino más corto para visitar todas las poblaciones preestablecidas y volver al lugar de origen pues bien este problema es un problema clásico de teoría de grafos y se conoce como el problema del viajante o del viajante de comercio que en inglés esto belén sesma problem y el objetivo es que el viajante visite una las ciudades preestablecidas una única vez y vuelva al punto de partida recorriendo la minima distancia posible para recorrer la minima distancia habrá que determinar el orden que ha de ir visitando estas ciudades la modelización que hacemos del problema es similar a la que hemos visto en otros problemas los vértices se corresponden con las ciudades o poblaciones y las aristas con las carreteras que las conectan existe una solución de coste computacional una solución exacta al problema pero el coste computacional es muy elevado a este problema se dice que es un problema np completo es decir no hay un algoritmo que coste col polinómico lo resuelva o por lo menos

Troceo

LANGCHAIN
STANZA

```
Chunks:  10

CHUNK ID:  96375264-e711-0545-a6a9-3aeb11fdfec4-es-1-10
page_content='vamos a ver ahora cuáles son los principales

CHUNK ID:  96375264-e711-0545-a6a9-3aeb11fdfec4-es-2-10
page_content='de o la banda de comunicaciones tenemos aquí

CHUNK ID:  96375264-e711-0545-a6a9-3aeb11fdfec4-es-3-10
page_content='material estos coeficientes son son distintos

CHUNK ID:  96375264-e711-0545-a6a9-3aeb11fdfec4-es-4-10
page_content='por ciento podemos ver en primer lugar que es

CHUNK ID:  96375264-e711-0545-a6a9-3aeb11fdfec4-es-5-10
page_content='átomo de silicio rodeado por dos de germanio
```

Prompt

¿Qué se hace si el costo de ir a un vértice intermedio es infinito en el algoritmo de Floyd Warshall?

hackathon-pln-es/paraphrase-spanish-distilroberta

hiiamsid/sentence_similarity_spanish_es

Modelo sentence transformer

Modelo sentence transformer (embeddings)

```
[[ 5.06358333e-02 -1.05640627e-01  7.34398142e-02 -3.73049751e-02
  -8.24301779e-01  1.05691671e-01 -3.94928396e-01  4.73771065e-01
  -1.55304847e-02 -2.09006771e-01 -1.03202112e-01 -8.04044366e-01
   2.53202379e-01  6.84906900e-01  4.48702693e-01 -6.44352019e-01
  -8.80799115e-01 -6.34729862e-01 -2.61860132e-01  5.97549558e-01
  -1.08259313e-01 -2.62562722e-01  6.13226831e-01 -5.73833764e-01
  -7.92396814e-02  2.50171274e-01  5.34025431e-01  3.64548415e-01
   3.79988104e-01 -2.29440153e-01  3.81207541e-02  1.10504672e-01
   5.53961098e-01  2.90008307e-01 -5.92698574e-01 -3.03615719e-01
  -6.11763597e-01 -1.40954301e-01 -3.67093116e-01 -7.56292269e-02
  -5.08366585e-01  7.84000397e-01  2.94597775e-01 -1.80093378e-01
   3.60840969e-02 -3.04007202e-01  1.35826471e-03  3.93313825e-01
  -2.17118766e-02 -4.82794523e-01 -4.04380888e-01  2.69106384e-02
  -1.35055929e-01  4.02209461e-02  2.91002184e-01 -1.95927978e-01
  -3.52636218e-01 -1.78240523e-01  5.25177360e-01  1.40406899e-02
   4.71500129e-01 -2.41142496e-01 -1.05385780e-01  5.46730496e-03
  -1.58865787e-02  1.60915311e-02 -5.63011110e-01 -2.96117455e-01
  -4.65531498e-01  3.50438207e-02  8.80551934e-02 -6.65815532e-01
   2.65740663e-01  1.05463387e-02 -2.38081321e-01  2.29002118e-01
  -4.31801826e-01 -1.20894289e+00  1.84754774e-01  3.28569040e-02
   1.21093867e-02 -5.58392346e-01 -7.00938776e-02 -3.29579189e-02
   3.38207543e-01 -9.18567955e-01 -4.63521421e-01  8.16378474e-01
   2.38938242e-01 -3.51232231e-01 -1.16267294e-01 -2.08648685e-02
  -5.42877674e-01  1.49845332e-01 -7.88723081e-02 -8.36026073e-01
   4.29221720e-01  7.84379125e-01 -5.52735746e-01  6.87306225e-01
   1.02310288e+00 -1.73760831e-01 -2.57695109e-01 -2.26044357e-01
   4.24143970e-01  2.87447751e-01  6.61724865e-01  3.85540158e-01
```

Función de distancia L2^2 Cosine similarity

Número de resultados

Distancia máxima

position	videoID	totalchunks	chunknumber	videopercentage	inversedistance	ponderation
1	4cdb1530-8dc8-11e5-b5c5-79269adc4720	7	4	0.428571	0.601626	0.601626
2	786647e0-8d2a-11e5-b5c5-79269adc4720	10	3	0.2	0.600426	0.600426
3	786647e0-8d2a-11e5-b5c5-79269adc4720	10	6	0.5	0.58887	0.58887
4	5b986a3e-ba74-8b47-a109-a42f77af4899	8	2	0.125	0.583737	0.583737
5	3b14c260-8dc8-11e5-b5c5-79269adc4720	9	5	0.444444	0.582847	0.582847
6	3b14c260-8dc8-11e5-b5c5-79269adc4720	9	2	0.111111	0.581607	0.581607

Suma distancias inversas embeddings (u otra función)

	videoID	position	inversedistance	starttime
4	3b14c260-8dc8-11e5-b5c5-79269adc4720	5	0.582847	235.556
5	3b14c260-8dc8-11e5-b5c5-79269adc4720	6	0.581607	58.8889
13	3b14c260-8dc8-11e5-b5c5-79269adc4720	14	0.543915	176.667
30	3b14c260-8dc8-11e5-b5c5-79269adc4720	31	0.514448	117.778

Modelo sentence transformer

```
[[ 2.02749535e-01 -2.94297844e-01 -1.35566741e-01 -3.56267273e-01
  -7.41501331e-01  5.73129714e-01  7.80099928e-02  6.19875610e-01
  -2.77250353e-02 -1.59624025e-01 -7.47680888e-02 -5.59022248e-01
   2.68425405e-01  1.11099458e+00 -5.49181029e-02 -5.85959494e-01
  -1.12856007e+00 -9.01477396e-01 -2.00370416e-01 -1.04925588e-01
  -2.15428591e-01  5.83995879e-02  1.02414325e-01 -8.94946873e-01
  -1.06905031e+00  1.48028970e-01  2.39937112e-01 -7.21366107e-02
   7.90857255e-01  1.46516770e-01  2.80191034e-01 -1.92026779e-01
   6.55784965e-01  2.03917935e-01 -2.05507055e-01 -4.23334509e-01
  -5.49599886e-01  8.93159807e-02  3.94596845e-01 -3.65348160e-01
  -2.13531941e-01  1.05895400e+00  4.22934294e-01  8.03557131e-03
   3.50571543e-01 -4.50815171e-01 -4.93423522e-01 -1.33019462e-01
  -9.83991846e-03 -5.83450913e-01 -4.31719691e-01  3.61981004e-01
  -2.85323232e-01 -2.19812065e-01  2.16965415e-02  2.15310216e-01
  -2.96250917e-02  9.66454148e-01  4.09537345e-01 -4.26953137e-01
   9.80897546e-01  3.08119237e-01  1.81322739e-01 -7.93146901e-03
  -1.12200342e-01  7.66798258e-02 -4.38248634e-01  5.81914604e-01
  -6.34084821e-01 -8.27846408e-01 -1.30282030e-01 -8.45160723e-01
   1.54547710e-02  1.81473479e-01 -2.89977361e-02  3.28688085e-01
  -1.04429448e+00 -2.12244678e+00  2.48529345e-01 -7.75075108e-02
  -6.89957514e-02 -6.40023768e-01 -3.53493035e-01  1.21361531e-01
   3.20549123e-02 -6.76555216e-01 -1.03989470e+00  7.19095469e-01
  -2.63877809e-01  3.52476835e-01 -4.24150705e-01  1.20350122e-01
  -1.56736806e-01  4.05614346e-01  5.64359576e-02 -3.99387836e-01
  -2.41383135e-01  8.69309187e-01 -6.62261903e-01  8.22840691e-01
   7.54021466e-01 -1.09781399e-01  2.56675690e-01  7.45241791e-02
  -2.46534020e-01  5.41768074e-01  7.22605705e-01  2.08181068e-01
```

768 números

Variables a considerar

- ¿Pasamos a minúsculas los textos y el prompt?
- ¿Cómo troceamos las transcripciones?
- ¿Qué modelo sentence-transformer usamos?
- ¿Hacemos ajuste fino del modelo?¿cómo?
- ¿Qué función de distancia usamos?
- ¿Cuántos embeddings tenemos en cuenta y con qué distancia máxima?
- ¿Seleccionamos el vídeo solo por un embedding o sumamos las distancias?
- ¿Aplicamos alguna función de corrección al sumar?
- ¿Cómo probamos cada configuración?
- ¿Qué base de datos de vectores escogemos?

Separación en trozos

```
model.get_max_seq_length()
```

LangChain

LangChain is a framework for developing applications powered by language models. We believe that the most powerful and differentiated applications will not only call out to a language model via an api, but will also:

1. Be data-aware: connect a language model to other sources of data
2. Be agentic: Allow a language model to interact with its environment

As such, the LangChain framework is designed with the objective in mind to enable those types of applications.

There are two main value props the LangChain framework provides:

1. Components: LangChain provides modular abstractions for the components neccessary to work with language models. LangChain also has collections of implementations for all these abstractions. The components are designed to be easy to use, regardless of whether you are using the rest of the LangChain framework or not.
2. Use-Case Specific Chains: Chains can be thought of as assembling these components in particular ways in order to best accomplish a particular use case. These are intended to be a higher level interface through which people can easily get started with a specific use case. These chains are also designed to be customizable.

https://docs.langchain.com/docs/

Stanza

Stanza: A Python NLP Library for Many Human Languages

Run Stanza Tests passing | pypi v1.5.0 | conda v1.5.0 | python 3.6 | 3.7 | 3.8 | 3.9 | 3.10

The Stanford NLP Group's official Python NLP library. It contains support for running various accurate natural language processing tools on 60+ languages and for accessing the Java Stanford CoreNLP software from Python. For detailed information please visit our official website.

🔥 A new collection of **biomedical** and **clinical** English model packages are now available, offering seamless experience for syntactic analysis and named entity recognition (NER) from biomedical literature text and clinical notes. For more information, check out our Biomedical models documentation page.

References

If you use this library in your research, please kindly cite our ACL2020 Stanza system demo paper:

```
@inproceedings{qi2020stanza,
    title={Stanza: A {Python} Natural Language Processing Toolkit for Many Human Languages},
    author={Qi, Peng and Zhang, Yuhao and Zhang, Yuhui and Bolton, Jason and Manning, Christopher D.},
    booktitle = "Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: S
    year={2020}
}
```

https://github.com/stanfordnlp/stanza

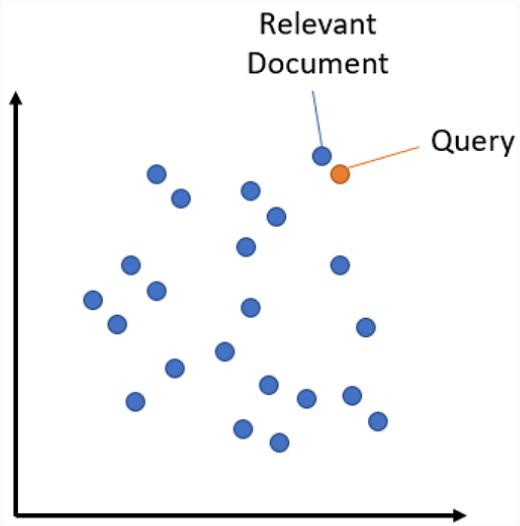
Sentence transformers

- Se generan a partir de modelos BERT, añadiendo una capa de concentración
- Se pueden entrenar de 0, hacer ajustes finos o ajustar los de otro idioma
- Se pueden usar modelos multilíngües
- Hay 2 en español: `paraphrase-spanish-distilroberta, sentence_similarity_spanish_es`
- Para hacer ajuste fino hay que crear pares de frases y anotar su similaridad
- Se puede usar gpt4 para crear frases similares a las de las transcripciones

https://www.sbert.net/examples/applications/semantic-search/README.html

Sentence Transformers

At search time, the query is embedded into the same vector space and the closest embeddings from your corpus are found. These entries should have a high semantic overlap with the query.



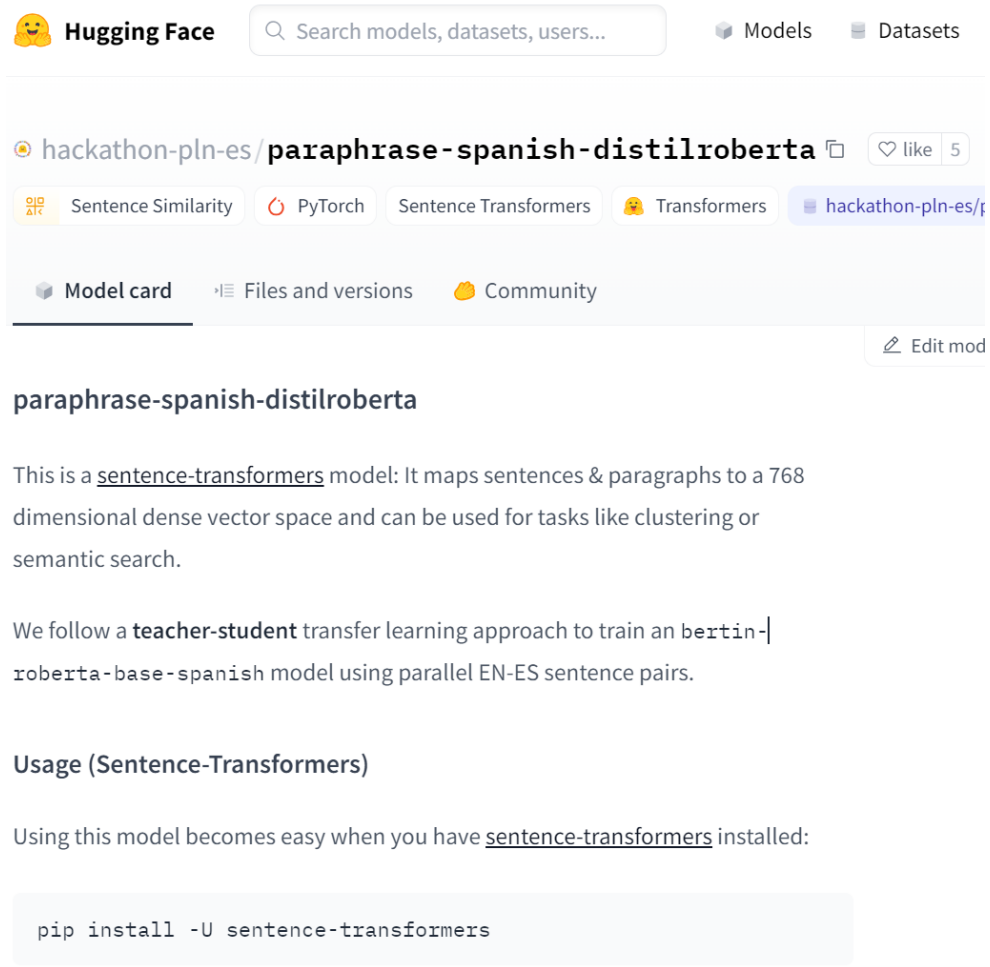
Symmetric vs. Asymmetric Semantic Search

A **critical distinction** for your setup is *symmetric* vs. *asymmetric semantic search*:

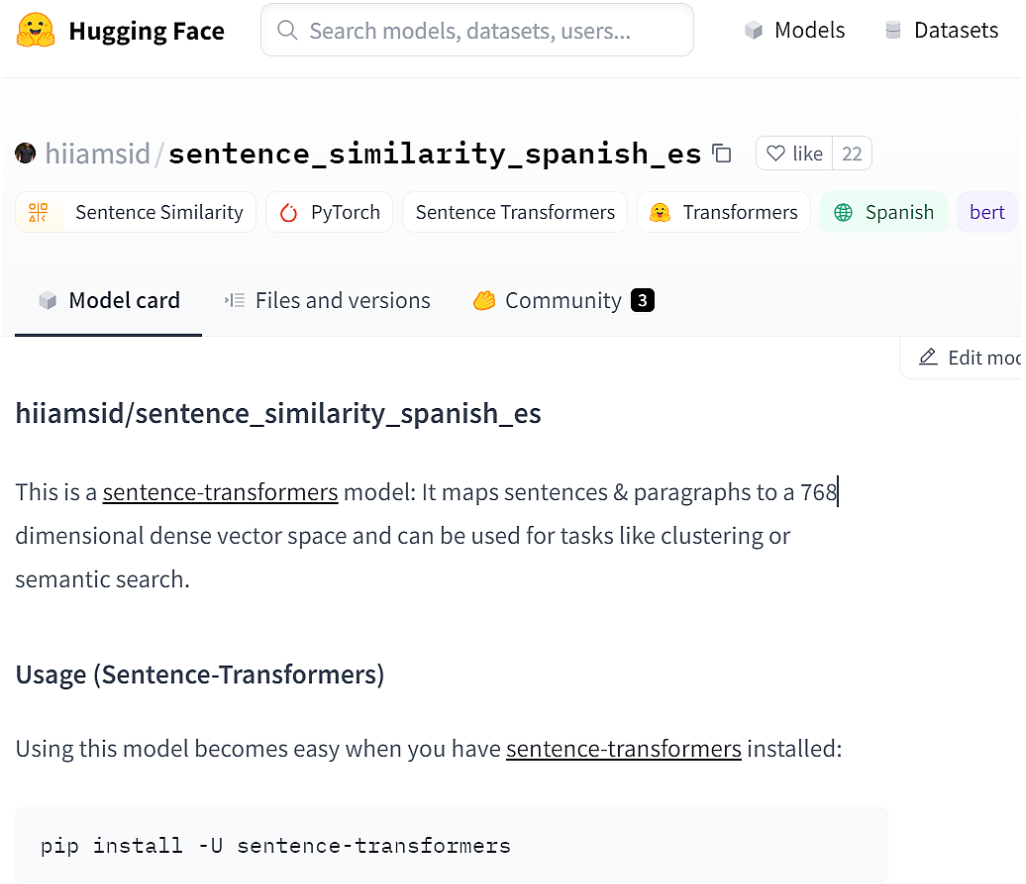
- For **symmetric semantic search** your query and the entries in your corpus are of about the same length and have the same amount of content. An example would be searching for similar questions: Your query could for example be *"How to learn Python online?"* and you want to find an entry like *"How to learn Python on the web?"*. For symmetric tasks, you could potentially flip the query and the entries in your corpus.
- For **asymmetric semantic search**, you usually have a **short query** (like a question or some keywords) and you want to find a longer paragraph answering the query. An example would be a query like *"What is Python"* and you wand to find the paragraph *"Python is an interpreted, high-level and general-purpose programming language. Python's design philosophy ..."*. For asymmetric tasks, flipping the query and the entries in your corpus usually does not make sense.

https://www.sbert.net/examples/applications/semantic-search/README.html

Sentence Transformers en español

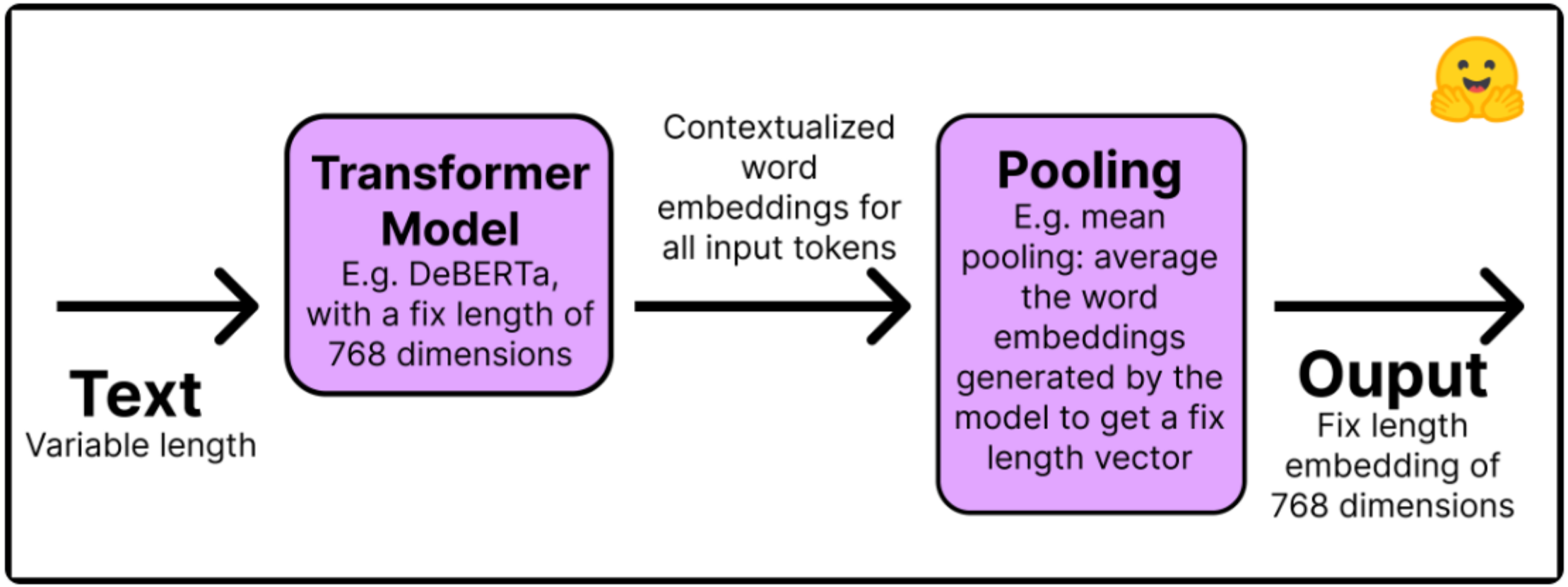


https://huggingface.co/hackathon-pln-es/paraphrase-spanish-distilroberta



https://huggingface.co/hiiamsid/sentence_similarity_spanish_es

Training Sentence Transformers



https://huggingface.co/blog/how-to-train-sentence-transformers

Fine tuning Sentence Transformers

Datasetstructures to train your SentenceTransformers model		
Datasetstructure	**Example datasets(repo id in Hugging Face Hub)**	**Loss functions(imported from sentence_transformers)**
Pair of sentences and a label indicating how similar they are	snli	ContrastiveLoss; SoftmaxLoss; CosineSimilarityLoss
Pair of positive (similar) sentences without a label	embedding-data/flickr30k_captions_quintets; embedding-data/coco_captions_quintets	MultipleNegativesRankingLoss; MegaBatchMarginLoss
Single sentence with an integer label	trec; yahoo_answers_topics	BatchHardTripletLoss; BatchAllTripletLoss; BatchHardSoftMarginTripletLoss; BatchSemiHardTripletLoss
Triplet (anchor, positive, negative) sentences	sembedding-data/QQP_triplets	TripletLoss

https://huggingface.co/blog/how-to-train-sentence-transformers

El modelo de Openai

To see embeddings in action, check out our code samples

- Classification
- Topic clustering
- Search
- Recommendations

Browse Samples

How to get embeddings

To get an embedding, send your text string to the embeddings API endpoint along with a choice of embedding model ID (e.g., `text-embedding-ada-002`). The response will contain an embedding, which you can extract, save, and use.

Example requests:

Example: Getting embeddings — python — Copy

```
response = openai.Embedding.create(
    input="Your text string goes here",
    model="text-embedding-ada-002"
)
embeddings = response['data'][0]['embedding']
```

Model	Usage
Ada v2	$0.0001 / 1K tokens

- 4 caracteres por token
- Sobre 0,08€ para los 371 vídeos del set de prueba
- Sobre 5€ para los 25.000 vídeos de Polimedia

https://platform.openai.com/docs/guides/embeddings/what-are-embeddings?lang=python

Embeddings a considerar

0 raw results had a bigger distance than the set threshold, 45 results returned.

	position	videoID	totalchunks	chunknumber	videopercentage	inversedistance	ponderation
0	1	4cdb1530-8dc8-11e5-b5c5-79269adc4720	7	4	0.428571	0.601626	0.601626
1	2	786647e0-8d2a-11e5-b5c5-79269adc4720	10	3	0.2	0.600426	0.600426
2	3	786647e0-8d2a-11e5-b5c5-79269adc4720	10	6	0.5	0.58887	0.58887
3	4	5b986a3e-ba74-8b47-a109-a42f77af4899	8	2	0.125	0.583737	0.583737
4	5	3b14c260-8dc8-11e5-b5c5-79269adc4720	9	5	0.444444	0.582847	0.582847
5	6	3b14c260-8dc8-11e5-b5c5-79269adc4720	9	2	0.111111	0.581607	0.581607
6	7	7ada8093-2b03-4346-9427-1233149b1bf7	10	3	0.2	0.581419	0.581419
7	8	e76946cf-aa89-394e-b180-133a1bd9fe15	6	3	0.333333	0.578708	0.578708
8	9	4cdb1530-8dc8-11e5-b5c5-79269adc4720	7	3	0.285714	0.575897	0.575897
9	10	0b9d9453-8830-8c40-b050-f20bbe2da998	9	5	0.444444	0.571778	0.571778
10	11	4150f213-d0a6-374f-8a98-5c632656b33e	13	2	0.0769231	0.57171	0.57171
11	12	754309f8-d338-8144-a358-c0c4c18dc5e6	14	1	0	0.56282	0.56282
12	13	8de4621f-130c-3f40-877a-bea215e32116	6	6	0.833333	0.553318	0.553318
13	14	3b14c260-8dc8-11e5-b5c5-79269adc4720	9	4	0.333333	0.543915	0.543915
14	15	0b9d9453-8830-8c40-b050-f20bbe2da998	9	7	0.666667	0.543387	0.543387
15	16	754309f8-d338-8144-a358-c0c4c18dc5e6	14	4	0.214286	0.539775	0.539775
16	17	786647e0-8d2a-11e5-b5c5-79269adc4720	10	2	0.1	0.539238	0.539238
17	18	7f33b9f3-5537-7544-b8fd-80b6133a4649	5	1	0	0.538402	0.538402
18	19	827c9f0c-32c9-a442-a515-d3580ac724aa	10000	10000	0.9999	0.53817	0.53817
19	20	f1bd97b7-49db-5c44-a6c9-530185dcfec1	10000	10000	0.9999	0.533503	0.533503
20	21	827c9f0c-32c9-a442-a515-d3580ac724aa	10	5	0.4	0.53292	0.53292

Selección del vídeo

Video 3b14c260-8dc8-11e5-b5c5-79269adc4720
Link: https://media.upv.es/#/portal/video/3b14c260-8dc8-11e5-b5c5-79269adc4720

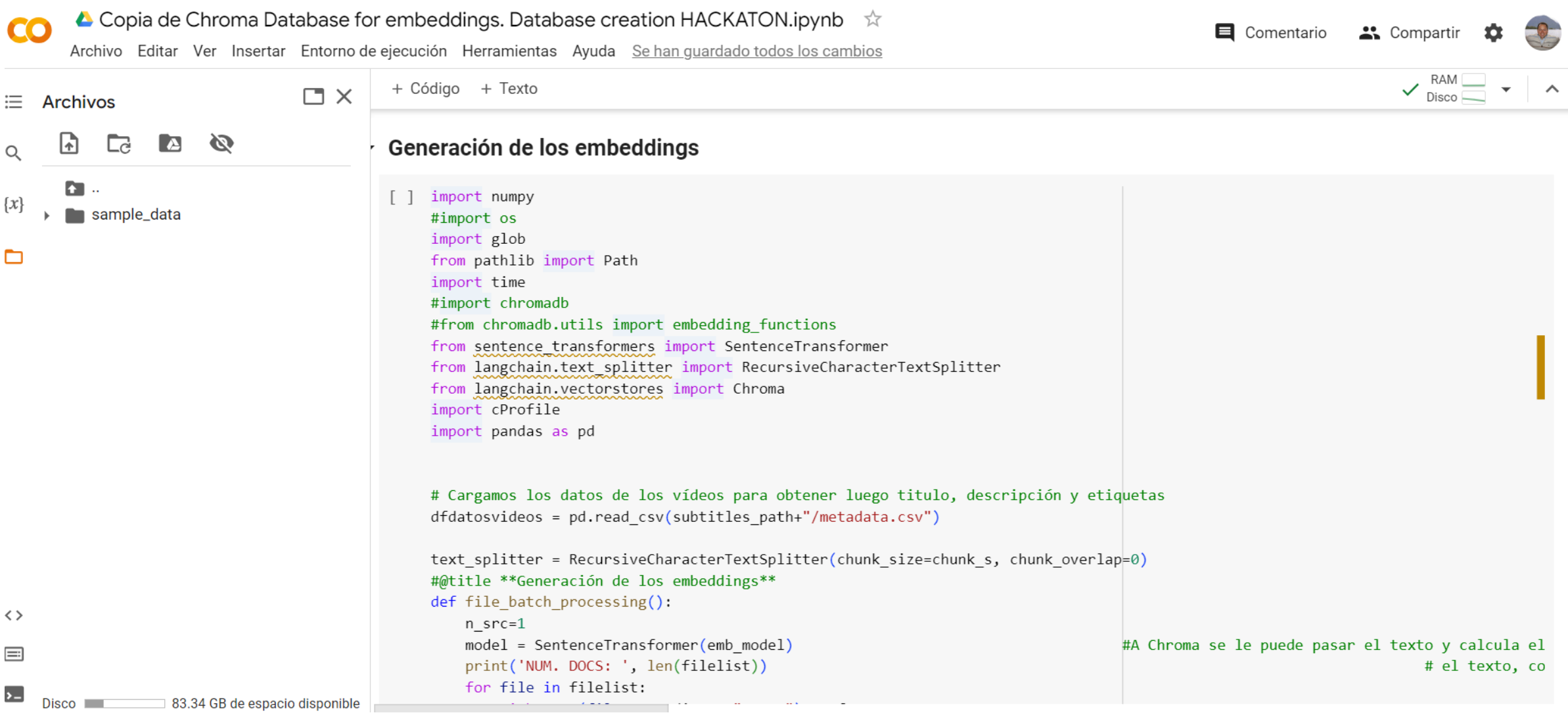
	videoID	position	inversedistance	starttime
4	3b14c260-8dc8-11e5-b5c5-79269adc4720	5	0.582847	235.556
5	3b14c260-8dc8-11e5-b5c5-79269adc4720	6	0.581607	58.8889
13	3b14c260-8dc8-11e5-b5c5-79269adc4720	14	0.543915	176.667
30	3b14c260-8dc8-11e5-b5c5-79269adc4720	31	0.514448	117.778

Probar las configuraciones

genera 10 preguntas de respuesta corta y sus respuestas sobre el texto que se adjunta a continuación, devuelve el resultado en formato csv, con los campos pregunta, respuesta. Las preguntas deben incluir todo el contexto necesario para entenderlas sin tener acceso al texto. El texto es "

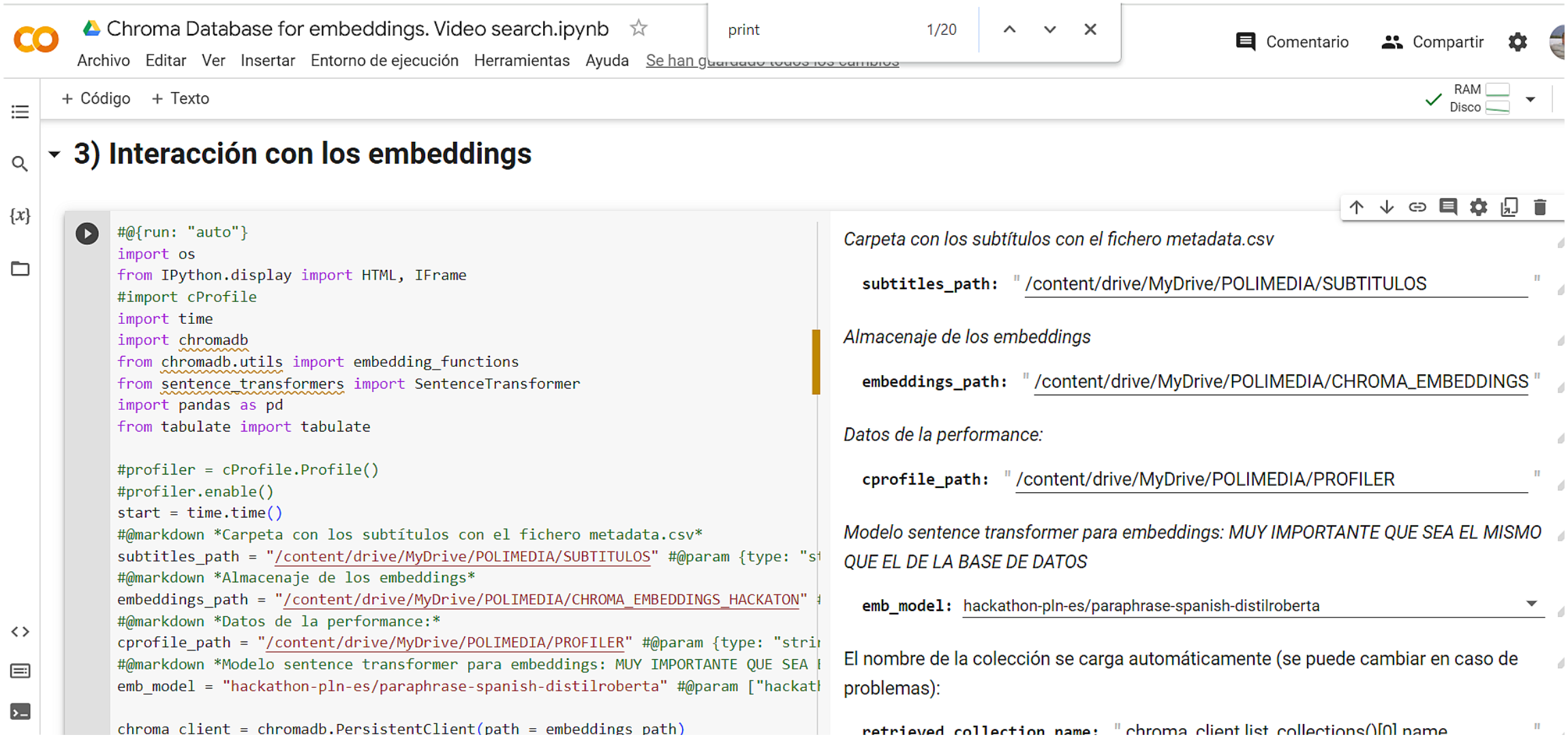
pregunta	respuesta	video
¿Cuál es el objetivo en el problema del viajante?	Que el viajan	0b9d9453-8830-8c40-b050-f20bbe2da998
¿Cómo se modeliza el problema del viajante?	Los vértices s	0b9d9453-8830-8c40-b050-f20bbe2da998
¿Por qué es un problema NP completo el problema del viajante?	Porque no ha	0b9d9453-8830-8c40-b050-f20bbe2da998
¿Cómo se calcula el costo computacional del problema del viajante?	El costo com	0b9d9453-8830-8c40-b050-f20bbe2da998
¿Qué es un ciclo hamiltoniano de peso bajo?	Un ciclo que	0b9d9453-8830-8c40-b050-f20bbe2da998
¿Qué condiciones son necesarias para aplicar el algoritmo del problema del viajante?	Necesitamos	0b9d9453-8830-8c40-b050-f20bbe2da998
¿Cómo se aplica el algoritmo del viajante para encontrar un ciclo de peso bajo?	Se elige un ve	0b9d9453-8830-8c40-b050-f20bbe2da998
¿Cuál es el enfoque principal del algoritmo de Floyd Warshall?	"Encontrar e	0b9f0878-0c02-6646-ad5a-ab2709b5dafb
¿Cómo se representan los caminos que no existen en el algoritmo de Floyd Warshall?	"Se represer	0b9f0878-0c02-6646-ad5a-ab2709b5dafb
¿Qué tipo de problemas no puede resolver el algoritmo de Floyd Warshall?	"No puede r	0b9f0878-0c02-6646-ad5a-ab2709b5dafb
¿Cómo se consideran los costes negativos en el contexto del algoritmo de Floyd Warshall	"Los costes r	0b9f0878-0c02-6646-ad5a-ab2709b5dafb
¿Qué tipo de información se puede dar con una matriz en el algoritmo de Floyd Warshall	"La matriz p	0b9f0878-0c02-6646-ad5a-ab2709b5dafb
¿Cuál es la idea central en la que se basa el algoritmo de Floyd?	"La idea es e	0b9f0878-0c02-6646-ad5a-ab2709b5dafb
¿Qué sucede si se encuentra un camino más corto al pasar por un vértice intermedio en e	"Si se encue	0b9f0878-0c02-6646-ad5a-ab2709b5dafb
¿Qué sucede si el camino más corto pasa por el vértice de origen o de destino en el algori	"Estos casos	0b9f0878-0c02-6646-ad5a-ab2709b5dafb
¿Qué se hace si el costo de ir a un vértice intermedio es infinito en el algoritmo de Floyd V	"Este caso n	0b9f0878-0c02-6646-ad5a-ab2709b5dafb
¿Qué causa la atenuación en muchas de las fibras ópticas instaladas actualmente?	La atenuació	0c5e963b-6a1f-fe48-9c8e-510714d77998
¿Dónde se encuentra la resonancia de absorción del enlace h o radical h?	La resonanci	0c5e963b-6a1f-fe48-9c8e-510714d77998
¿Qué ocurre cuando existen procesos no lineales en el mecanismo de absorción de la fibr	Los procesos	0c5e963b-6a1f-fe48-9c8e-510714d77998
¿De qué depende la altura de los picos de absorción en una fibra óptica?	La altura de l	0c5e963b-6a1f-fe48-9c8e-510714d77998

Generación de embeddings



https://colab.research.google.com/drive/1ykqaPyqqP68pUrZPvdQNj7rSgQnh1XKE?usp=sharing

Prueba de embeddings



https://colab.research.google.com/drive/1VPbi9-UcBy2IwxSvwZXOVSZVIwOhDdCe?usp=sharing

Bases de datos de embeddings

- Pinecone (de pago)
- Milvus (tiene versión para instalar)
- Weviate (tiene versión para instalar)
- Vespa (tiene versión para instalar)
- Quadrant (tiene versión para instalar)
- ChromaDB (vale para pruebas)
- FAISS con SQL