Transformer models

JISU KIM, PH.D

JUNE 2024

Materials

Slides + Codes are available here:

https://github.com/jisukimmmm/NCCR_MWQTA_2024

Outline

1. Transformers, what are they and what can they do?
2. Introducing transformer-based models
3. How do Transformer models work?
4. What are they composed of?
5. Bias and limitations
6. Using Transformer models

For Day 4, the materials are based on : https://huggingface.co/course

Transformers, what are they?

Transformer models are a type of neural network designed for handling sequences of data, like sentences.

They use a mechanism called **self-attention** to determine the importance of each word in a sentence relative to the others, allowing them to understand context better.

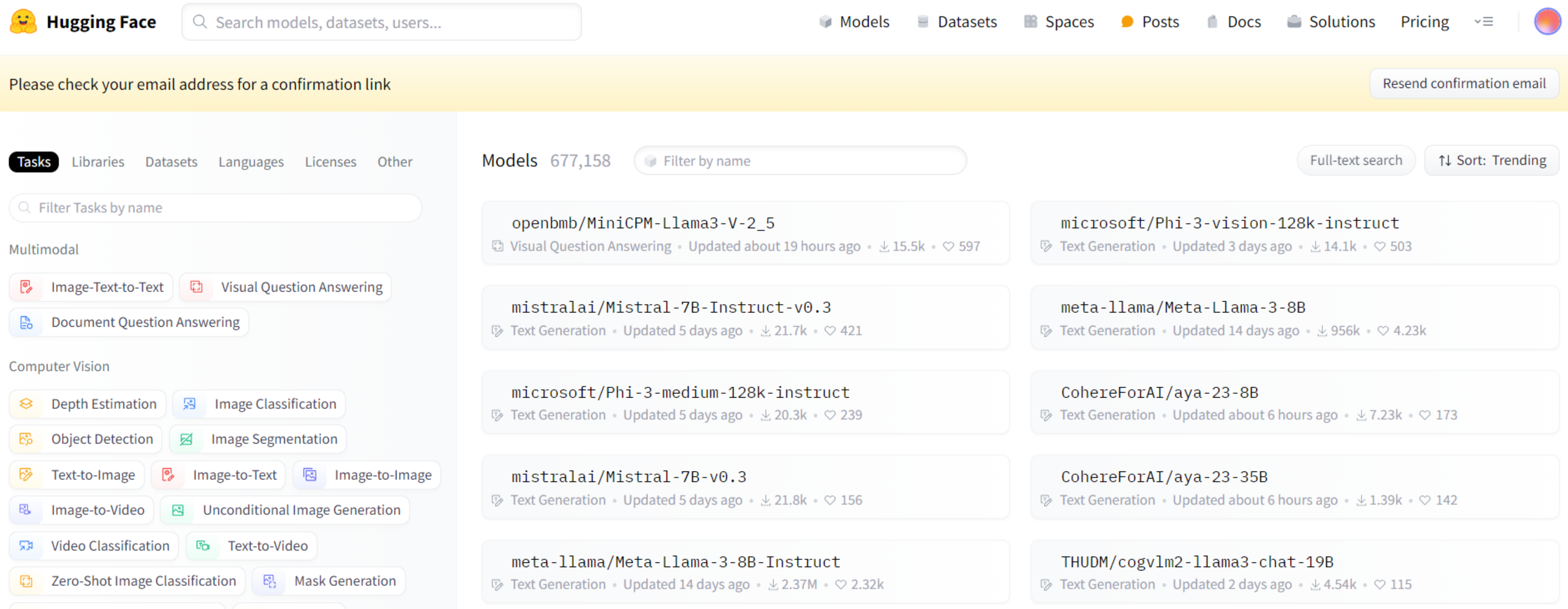
Transformers consist of an **encoder** that processes the input and a **decoder** that generates the output.

This architecture allows them to perform tasks like translating languages, summarizing text, generating coherent sentences, recognizing speech, and even analyzing images.

They are **faster** and more **scalable** than previous models because they process entire sequences simultaneously rather than one step at a time.

Model hub

Transformer models are used to solve all kinds of NLP tasks! The Model Hub contains thousands of pretrained models that anyone can download and use. You can also upload your own models to the Hub!→ The Transformers library provides the functionality to create and use those shared models.



Please check your email address for a confirmation link

Multimodal

Image-Text-to-Text · Visual Question Answering · Document Question Answering

Computer Vision

Depth Estimation · Image Classification · Object Detection · Image Segmentation · Text-to-Image · Image-to-Text · Image-to-Image · Image-to-Video · Unconditional Image Generation · Video Classification · Text-to-Video · Zero-Shot Image Classification · Mask Generation

Models 677,158

openbmb/MiniCPM-Llama3-V-2_5
Visual Question Answering • Updated about 19 hours ago • 15.5k • 597

microsoft/Phi-3-vision-128k-instruct
Text Generation • Updated 3 days ago • 14.1k • 503

mistralai/Mistral-7B-Instruct-v0.3
Text Generation • Updated 5 days ago • 21.7k • 421

meta-llama/Meta-Llama-3-8B
Text Generation • Updated 14 days ago • 956k • 4.23k

microsoft/Phi-3-medium-128k-instruct
Text Generation • Updated 5 days ago • 20.3k • 239

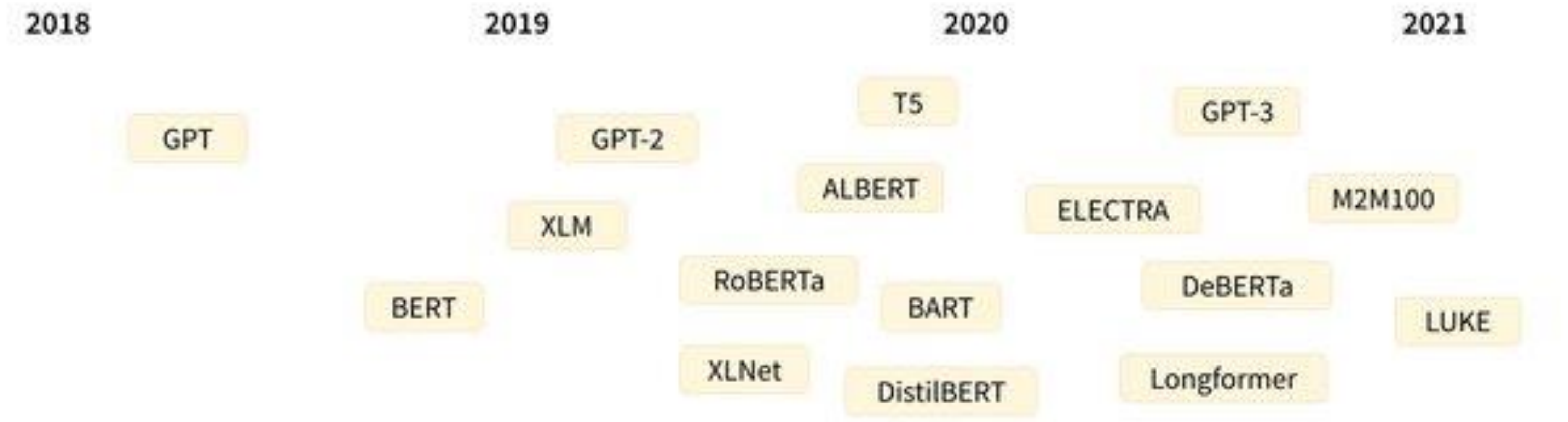
CohereForAI/aya-23-8B
Text Generation • Updated about 6 hours ago • 7.23k • 173

mistralai/Mistral-7B-v0.3
Text Generation • Updated 5 days ago • 21.8k • 156

CohereForAI/aya-23-35B
Text Generation • Updated about 6 hours ago • 1.39k • 142

meta-llama/Meta-Llama-3-8B-Instruct
Text Generation • Updated 14 days ago • 2.37M • 2.32k

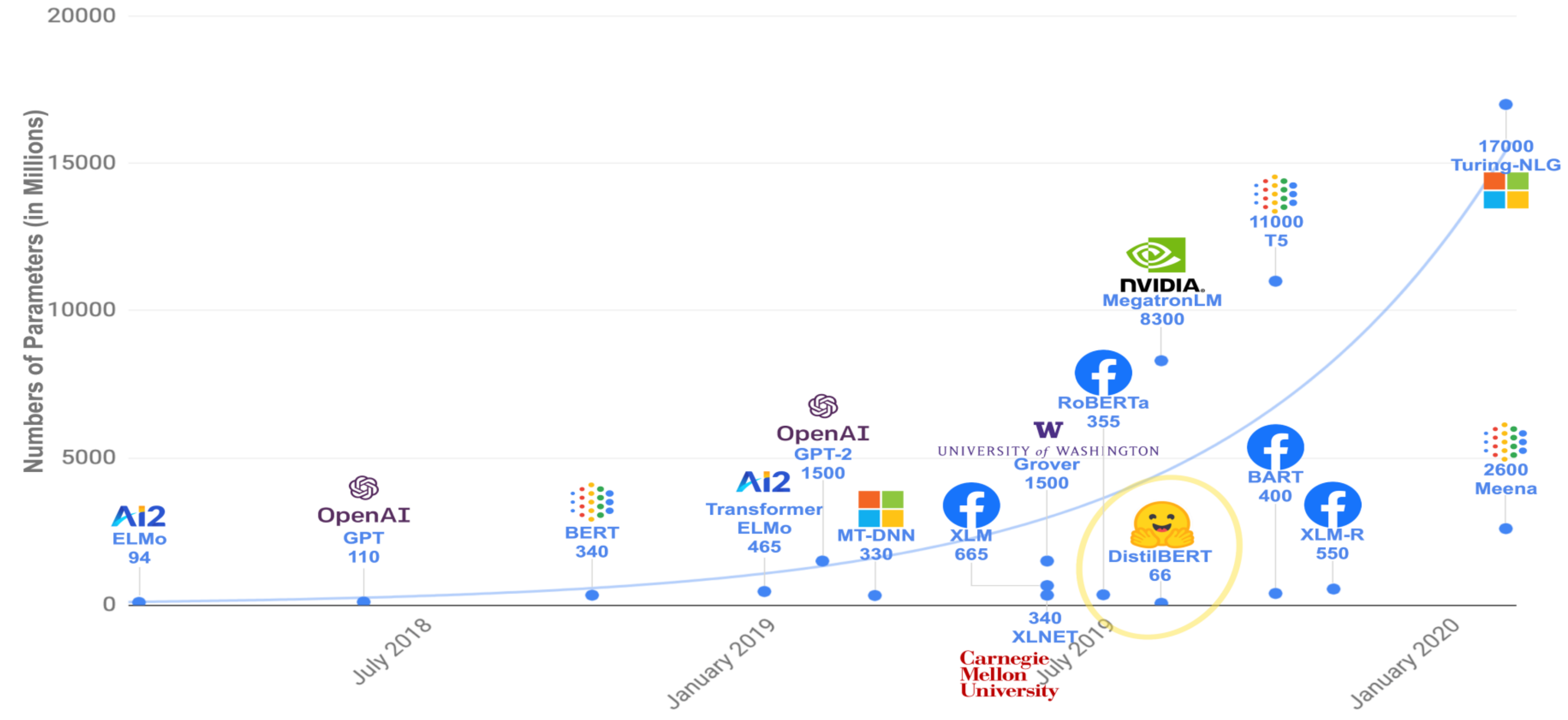
THUDM/cogvlm2-llama3-chat-19B
Text Generation • Updated 2 days ago • 4.54k • 115



Source. HuggingFace

Transformers are big models

Apart from a few outliers (like DistilBERT), the general strategy to achieve better performance is by **increasing the models' sizes as well as the amount of data they are pretrained on.**



Language models

They all have been trained as language models.

This means they have been trained on large amounts of raw text in a self-supervised fashion.

Self-supervised learning is a type of training in which the objective is automatically computed from the inputs of the model. That means that humans are not needed to label the data!

This type of model develops a statistical understanding of the language it has been trained on, but it’s not very useful for specific practical tasks. Because of this, the general pretrained model then goes through a process called ***transfer learning***[1]. During this process, the model is fine-tuned in a supervised way — that is, using human-annotated labels — on a given task.

[1] Transfer learning, used in machine learning, is the reuse of a pre-trained model on a new problem.

Three groups

Transformer models can be grouped into three categories:

- GPT-like (also called auto-regressive Transformer models)
- BERT-like (also called auto-encoding Transformer models)
- BART/T5-like (also called sequence-to-sequence Transformer models)

Transformers, what can they do?

Natural Language Processing (NLP)

a. Language Translation

Example: Google's Neural Machine Translation (GNMT) and OpenAI's GPT models can translate text between different languages with high accuracy.

b. Text Generation

Example: GPT-3 and GPT-4 can generate coherent and contextually relevant text, write essays, stories, and even code.

c. Text Summarization

Example: Models like BART and T5 can summarize long documents into concise summaries, helping in information extraction and content creation.

d. Question Answering

Example: Models like BERT, RoBERTa, and T5 can answer questions based on a given context, enabling applications like chatbots and virtual assistants.

Introducing transformer-based models

BERT (Bidirectional Encoder Representations from Transformers):

- BERT has been widely used for various NLP tasks, including topic classification.
- Pre-trained BERT models, such as bert-base-uncased, bert-large-uncased, etc., can be fine-tuned for topic classification on specific datasets.
- Libraries like Hugging Face's Transformers provide easy-to-use interfaces for fine-tuning BERT for topic classification.

RoBERTa (Robustly optimized BERT approach):

- RoBERTa builds upon BERT's architecture with modifications and training enhancements, leading to improved performance.
- Similar to BERT, pre-trained RoBERTa models can be fine-tuned for topic classification tasks.

Introducing transformer-based models

DistilBERT:

- DistilBERT is a smaller and faster variant of BERT, trained to retain much of BERT's performance while being more resource-efficient.
- It can be a good choice for topic classification tasks where computational resources are limited.

GPT (Generative Pre-trained Transformer) series (e.g., GPT-2, GPT-3):

- While GPT models are primarily used for text generation tasks, they can also be fine-tuned for classification tasks, including topic classification.
- GPT models are trained in an autoregressive manner and have demonstrated strong performance on a wide range of NLP tasks.

Introducing transformer-based models

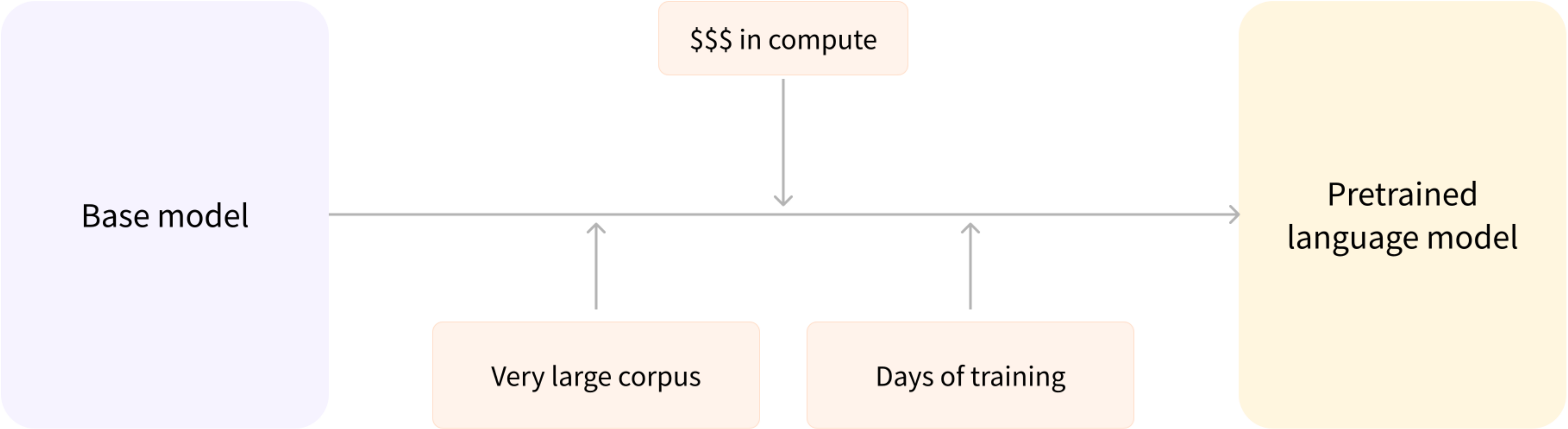
CamemBERT:

- CamemBERT is a French version of BERT, pre-trained on a large French corpus.
- It can be fine-tuned for topic classification tasks in French text.

And of course there are more: e.g., XLNet, Electra, T5, …

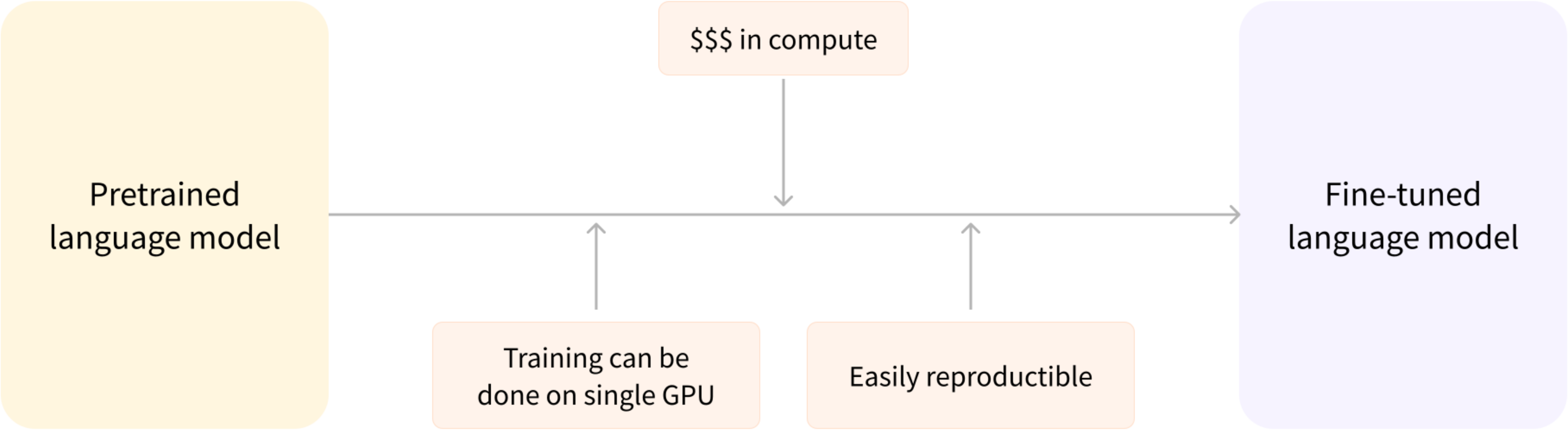
How do Transformer models work?

1. **Pretraining**: is the act of training a model from scratch: the weights are randomly initialized, and the training starts without any prior knowledge. This pretraining is usually done on very large amounts of data. Therefore, it requires a very large corpus of data, and training can take up to several weeks.



How do Transformer models work?

2. **Fine-tuning**: is the training done **after** a model has been pretrained. To perform fine-tuning, you first acquire a pretrained language model, then perform additional training with a dataset specific to your task.


Pretrained language model
$$$ in compute
Training can be done on single GPU
Easily reproductible
Fine-tuned language model

Fine-tuning vs Transfer learning

Fine-tuning adapts a pre-trained model on a specific task with modest changes, while **transfer learning** employs knowledge gained from a pre-trained model to enhance performance on a different but related task.

For example, one could leverage a pretrained model trained on the English language and then fine-tune it on an arXiv corpus, resulting in a science/research-based model.

→ the knowledge the pretrained model has acquired is “transferred,” hence the term *transfer learning*.

What are they composed of?

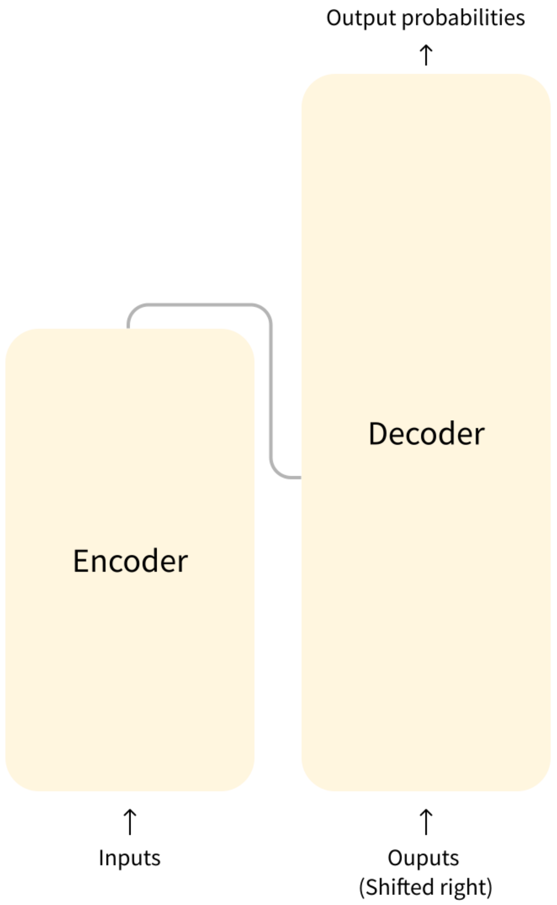
The model is primarily composed of 2 blocks:

Encoder (left): The encoder receives an input and builds a representation of it (its features). This means that the model is optimized to acquire understanding from the input.

Decoder (right): The decoder uses the encoder’s representation (features) along with other inputs to generate a target sequence. This means that the model is optimized for generating outputs.

Each of these parts can be used independently, depending on the task:

- **Encoder-only models**: Good for tasks that require understanding of the input, such as sentence classification and named entity recognition. E.g., BERT
- **Decoder-only models**: Good for generative tasks such as text generation. E.g., GPT
- **Encoder-decoder models** or **sequence-to-sequence models**: Good for generative tasks that require an input, such as translation or summarization. E.g., BART



Attention layers[1]

This layer will tell the model to pay specific attention to certain words in the sentence you passed it (and more or less ignore the others) when dealing with the representation of each word. It works like a spotlight, helping the model focus on the most important parts of a sentence when understanding or generating text.

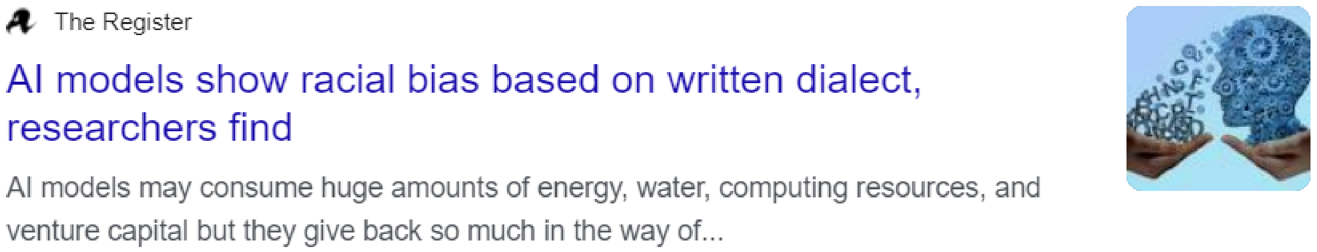
E.g., Imagine you are trying to understand a sentence like "The cat sat on the mat." To make sense of the word "cat," you might want to look at words around it, like "sat" and "mat.“

→ Attention layers do this automatically. They look at each word in the sentence and figure out how much attention to pay to every other word.

[1] Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).

Biases and limitations

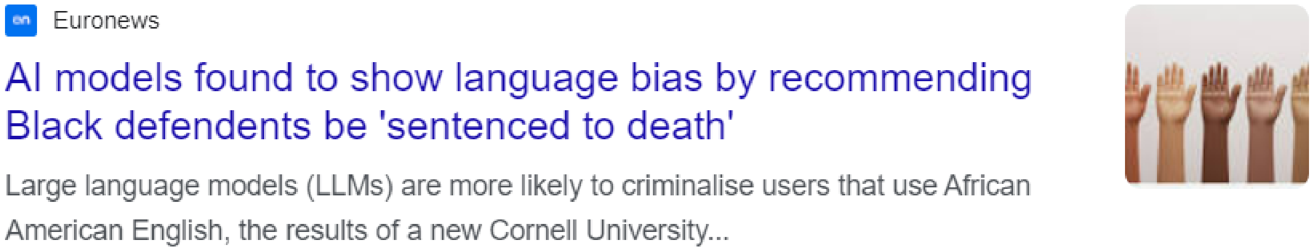
- It could very easily generate sexist, racist, or homophobic content
- Fine-tuning the model on your data won’t make this intrinsic bias disappear.



The Register

AI models show racial bias based on written dialect, researchers find

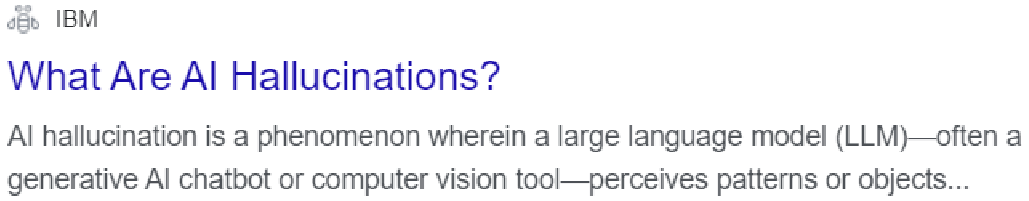
AI models may consume huge amounts of energy, water, computing resources, and venture capital but they give back so much in the way of...



Euronews

AI models found to show language bias by recommending Black defendents be 'sentenced to death'

Large language models (LLMs) are more likely to criminalise users that use African American English, the results of a new Cornell University...



IBM

What Are AI Hallucinations?

AI hallucination is a phenomenon wherein a large language model (LLM)—often a generative AI chatbot or computer vision tool—perceives patterns or objects...

Using Transformer models

Libraries and software-

- Scikit-learn (scikit-learn.org)
- TensorFlow (tensorflow.org)
- Keras (keras.io)
- PyTorch (pytorch.org)

→ You can use Google Colab (a hosted Jupyter Notebook service) if you want to access to computing resources, including GPUs. It is also well suited to ML, and DS.

TensorFlow

Overview: TensorFlow is an open-source deep learning framework developed by Google Brain. It supports both static and dynamic computational graphs, though it was originally designed with static graphs in mind.

Key Features:

- Static computation graph (with eager execution mode available).
- High performance on both CPU and GPU.
- Extensive ecosystem with tools like TensorBoard for visualization and TensorFlow Serving for deployment.

Transformers: TensorFlow is also used for developing and deploying transformer models. The TensorFlow implementation of transformers is available through the transformers library from Hugging Face, as well as TensorFlow's own implementations such as TensorFlow Hub, which provides pre-trained models that can be fine-tuned.

Keras

Overview: Keras is an open-source neural network library written in Python. It is designed to be user-friendly and modular, often serving as an interface for other deep learning libraries like TensorFlow and Theano. Since TensorFlow 2.0, Keras is integrated as tf.keras within TensorFlow.

Key Features:

- High-level API for easy model building.
- User-friendly and modular, making it accessible for beginners.
- Seamlessly integrates with TensorFlow.

Transformers: With Keras, transformer models can be easily constructed using its high-level APIs. Keras provides layers and functions that simplify the creation of transformer models. For instance, the Hugging Face Transformers library also supports TensorFlow/Keras, allowing users to load pre-trained transformer models in a Keras-compatible format.

PyTorch

Overview: PyTorch is an open-source deep learning library developed by Facebook's AI Research lab (FAIR). It is known for its dynamic computational graph, which allows for more flexibility in model building and debugging.

Key Features:

- Dynamic computation graph (eager execution).
- Strong GPU acceleration with CUDA support.
- Extensive library of pre-built models and tools.

Transformers: PyTorch is widely used for implementing transformer models. Libraries like Hugging Face's Transformers are built on top of PyTorch, providing pre-trained transformer models like BERT, GPT, and T5, which can be fine-tuned for various NLP tasks.

Summary Table

Feature	PyTorch	TensorFlow	Keras
Ease of Use	High	Moderate	Very High
Flexibility	Very High	High	Moderate
Debugging	Very Easy (Dynamic Graphs)	Moderate (Static Graphs, Eager Mode)	Very Easy
Deployment	Moderate (Improving)	Very High (Comprehensive Tools)	High (via TensorFlow)
Community	Large (Research Focus)	Large (Industry and Research)	Large (Beginner to Intermediate Focus)
Ecosystem and Tools	Moderate	Very High (Extensive)	High (via TensorFlow)
Use Case Examples	Research, Prototyping, Custom Models	Production, Mobile, Web, Large-scale ML	Rapid Prototyping, Easy Model Building

Pipeline()

The most basic object in the Transformers library is the **pipeline()** function. It connects a model with its necessary preprocessing and postprocessing steps, allowing us to directly input any text.

A pipeline selects a particular pretrained model that has been fine-tuned for a task. The model is downloaded and cached when you create the classifier object.

Pipeline allows you to sequentially apply a list of transformers to preprocess the data and, if desired, conclude the sequence with a final predictor for predictive modeling.

For example:

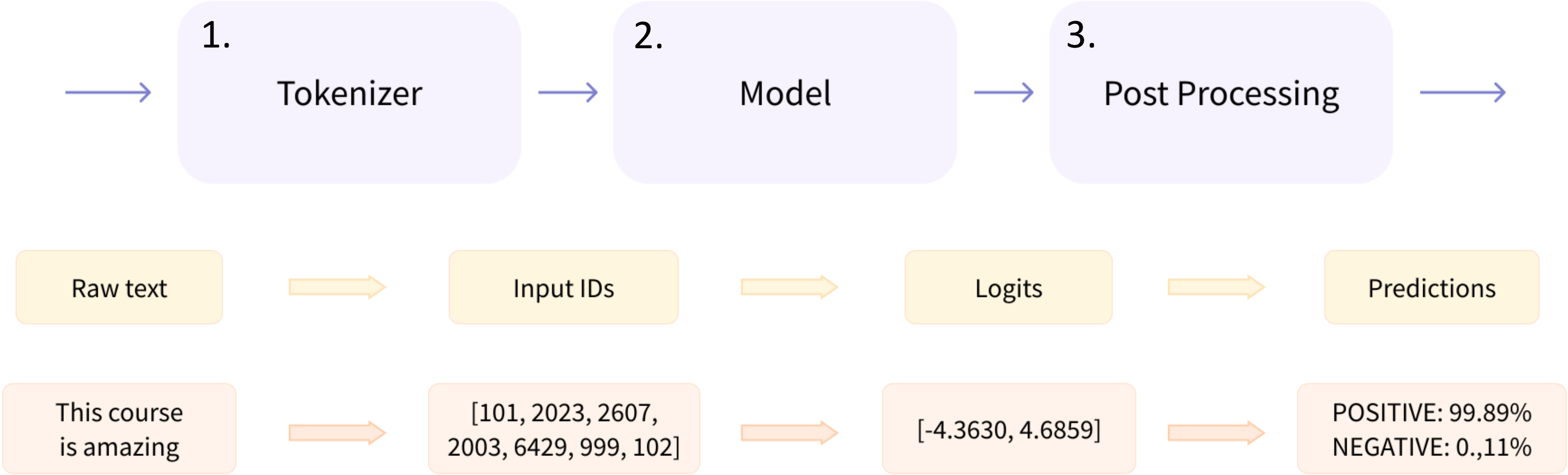
```
from transformers import pipeline


classifier = pipeline("sentiment-analysis")
classifier("I've been waiting for a HuggingFace course my whole life.")
```

3 steps in pipeline()

There are three main steps involved when you pass some text to a pipeline:

- The text is preprocessed into a format the model can understand.
- The preprocessed inputs are passed to the model.
- The predictions of the model are post-processed, so you can make sense of them.



1. Tokenizer

2. Model

3. Post Processing

Raw text → Input IDs → Logits → Predictions

This course is amazing → [101, 2023, 2607, 2003, 6429, 999, 102] → [-4.3630, 4.6859] → POSITIVE: 99.89%
NEGATIVE: 0.,11%

Handling multiple sequences

How do we handle multiple sequences?

→ Using ***Batching:*** It is the act of sending multiple sentences through the model, all at once. If you only have one sentence, you can just build a batch with a single sequence:

How do we handle multiple sequences *of different lengths*?

→ Using **Padding:** It makes sure all our sentences have the same length by adding a special word called the *padding token* to the sentences with fewer values. For example, if you have 10 sentences with 10 words and 1 sentence with 20 words, padding will ensure all the sentences have 20 words.

Is there such a thing as too long a sequence?

→ With Transformer models, there is a limit to the lengths of the sequences we can pass the models. Most models handle sequences of up to 512 or 1024 tokens, and will crash when asked to process longer sequences.

Zero-shot classification

It allows you to specify which labels to use for the **classification**, so you don’t have to rely on the labels of the pretrained model.

It is called *zero-shot* because you don’t need to fine-tune the model on your data to use it. It can directly return probability scores for any list of labels you want!

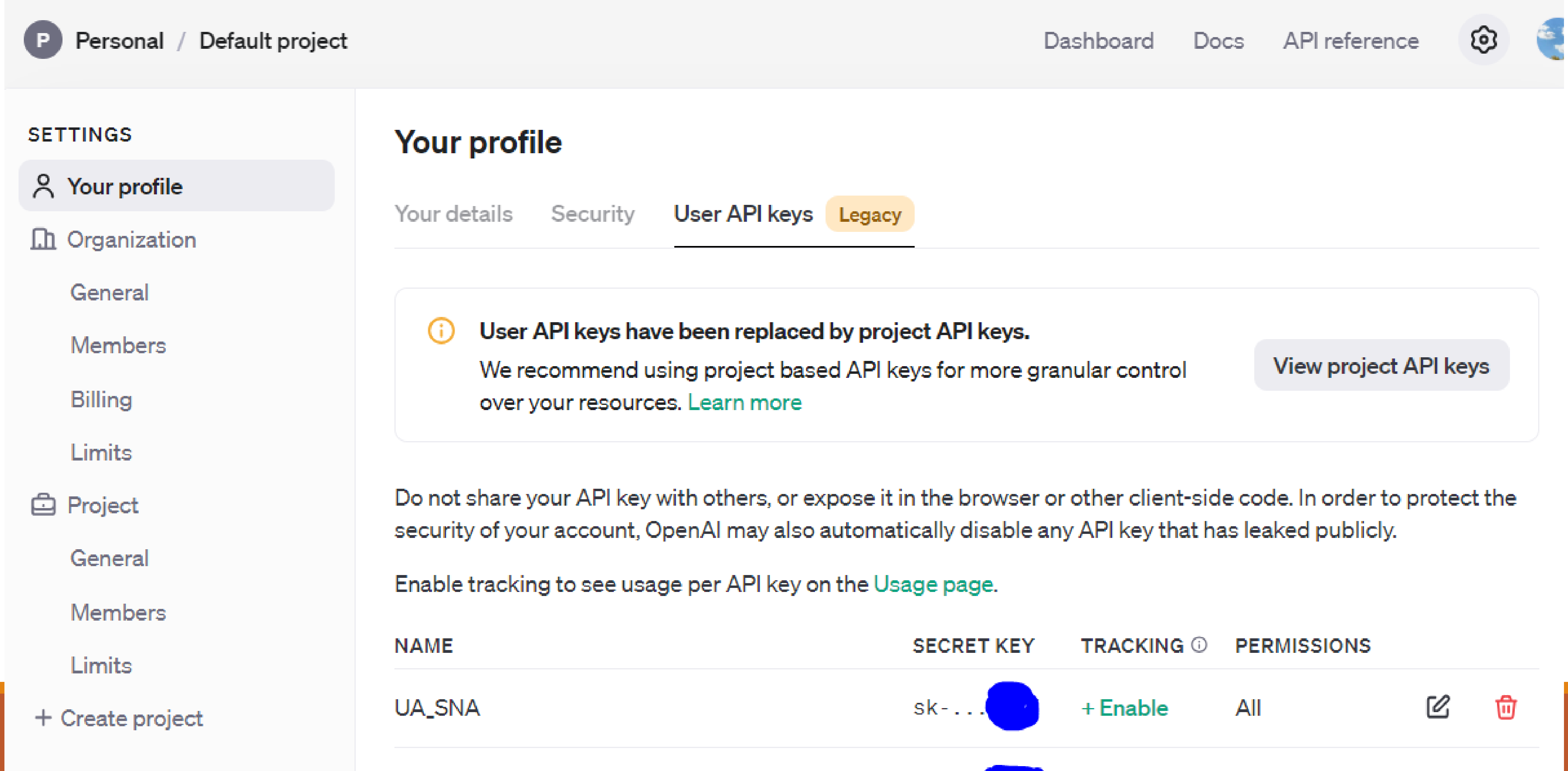
```
from transformers import pipeline
```

Example:

```
    classifier = pipeline("zero-shot-classification")
    classifier( "This is a course about the Transformers library",
                candidate_labels=["education", "politics", "business"])
    {'sequence': 'This is a course about the Transformers library',
     'labels': ['education', 'business', 'politics'],
     'scores': [0.844, 0.112, 0.043]}
```

Using GPT API

https://platform.openai.com/docs/overview



Your profile

User API keys have been replaced by project API keys.

We recommend using project based API keys for more granular control over your resources. Learn more

Do not share your API key with others, or expose it in the browser or other client-side code. In order to protect the security of your account, OpenAI may also automatically disable any API key that has leaked publicly.

Enable tracking to see usage per API key on the Usage page.

NAME	SECRET KEY	TRACKING ⓘ	PERMISSIONS
UA_SNA	sk-...[redacted]	+ Enable	All
	[redacted]		

Some migration studies literature using NLP

Tun-Mendicuti, A., Kim, J., & Mulder, C. H. (2024, May). Understanding opinions towards migrants in transit: An analysis of tweets on Migrant Caravans in the US and Mexico. In *Proceedings of the 16th ACM Web Science Conference* (pp. 1-10).

Mittal, J., Belorkar, A., Jakhetiya, V., Pokuri, V., & Guntuku, S. C. (2023, April). Language on reddit reveals differential mental health markers for individuals posting in immigration communities. In *Proceedings of the 15th ACM Web Science Conference 2023* (pp. 153-162).

Arcila-Calderón, C., Blanco-Herrero, D., Frías-Vázquez, M., & Seoane-Pérez, F. (2021). Refugees welcome? Online hate speech and sentiments in twitter in Spain during the reception of the boat Aquarius. Sustainability, 13(5), 2728.

Öztürk, N., & Ayvaz, S. (2018). Sentiment analysis on Twitter: A text mining approach to the Syrian refugee crisis. *Telematics and Informatics*, *35*(1), 136-147.

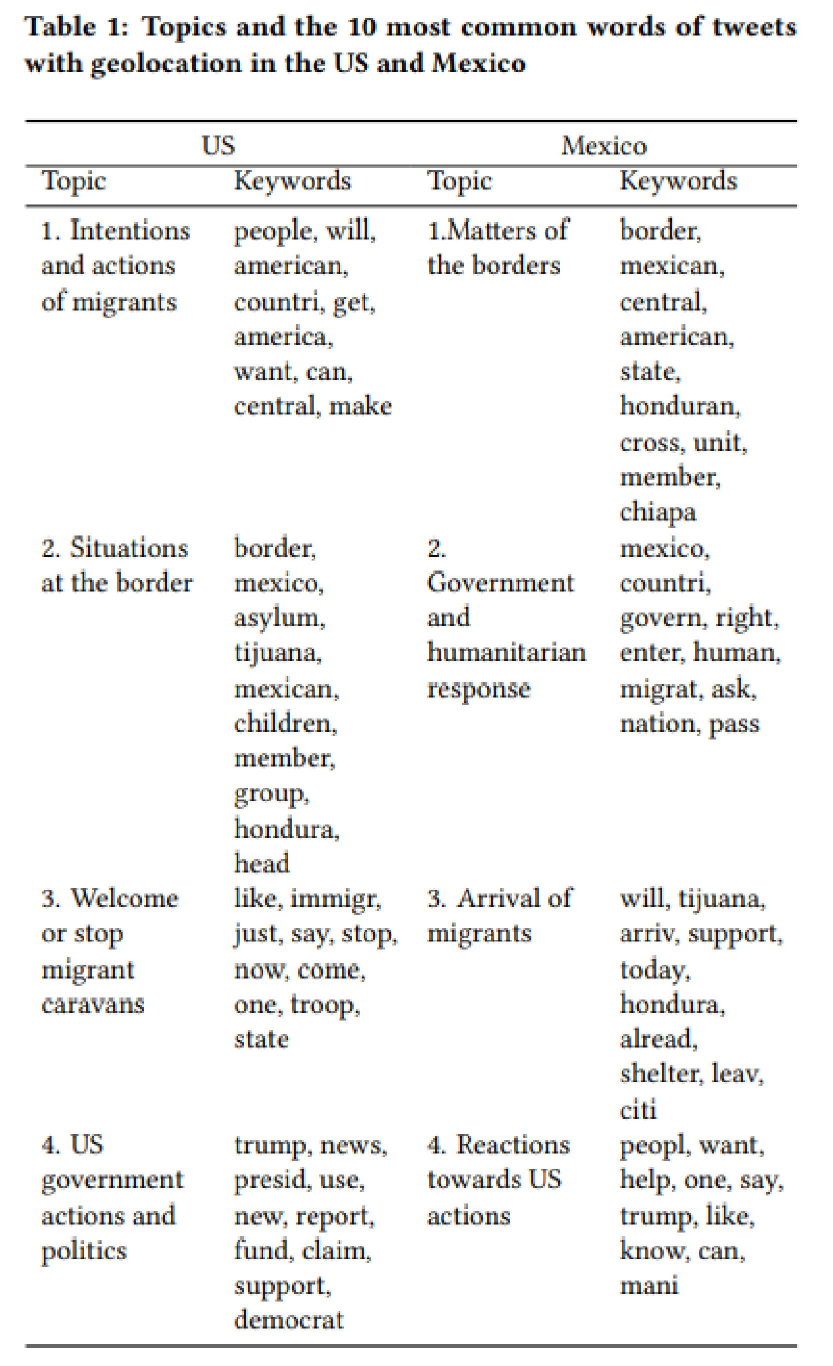
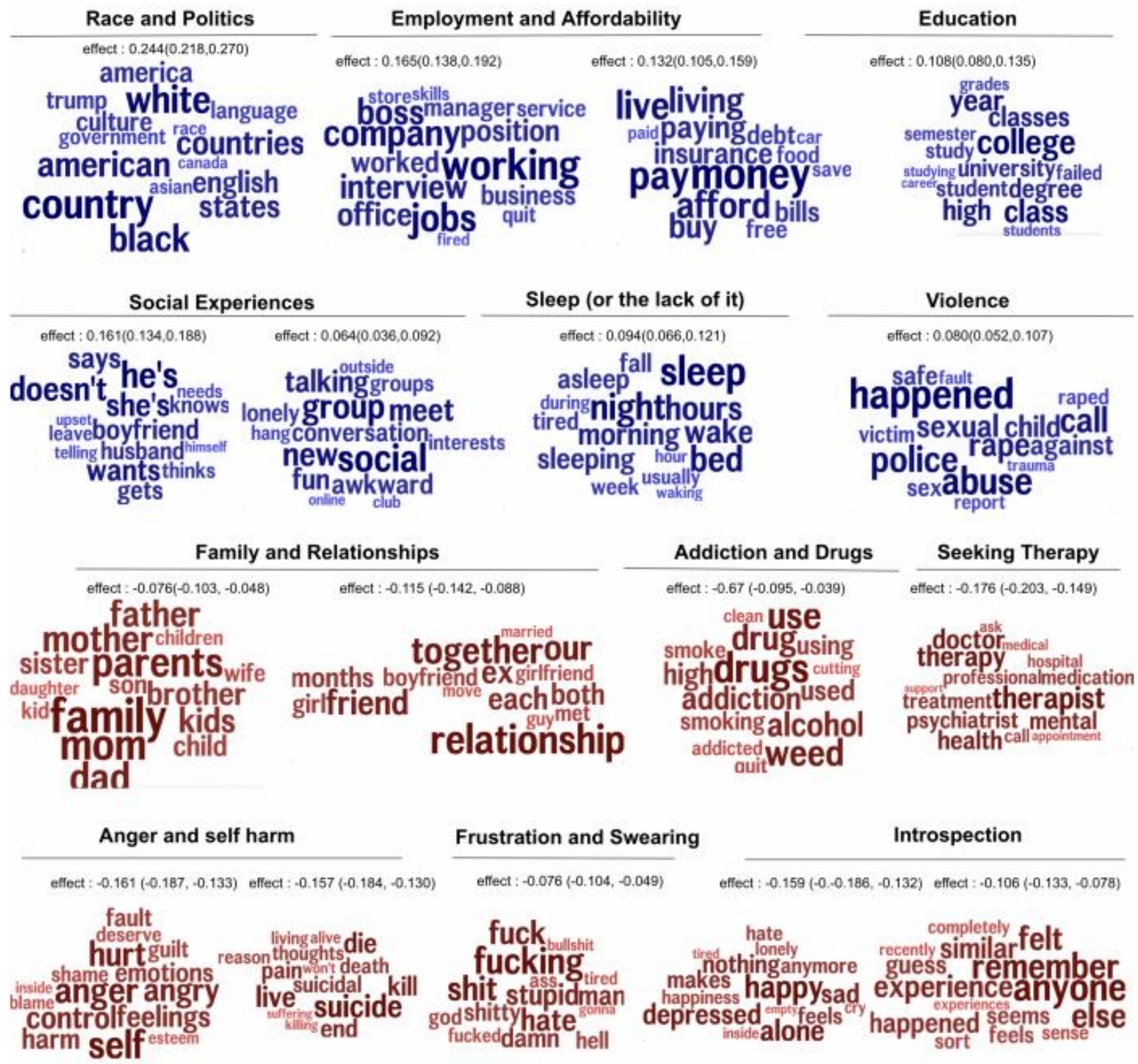


Table 1: Topics and the 10 most common words of tweets with geolocation in the US and Mexico

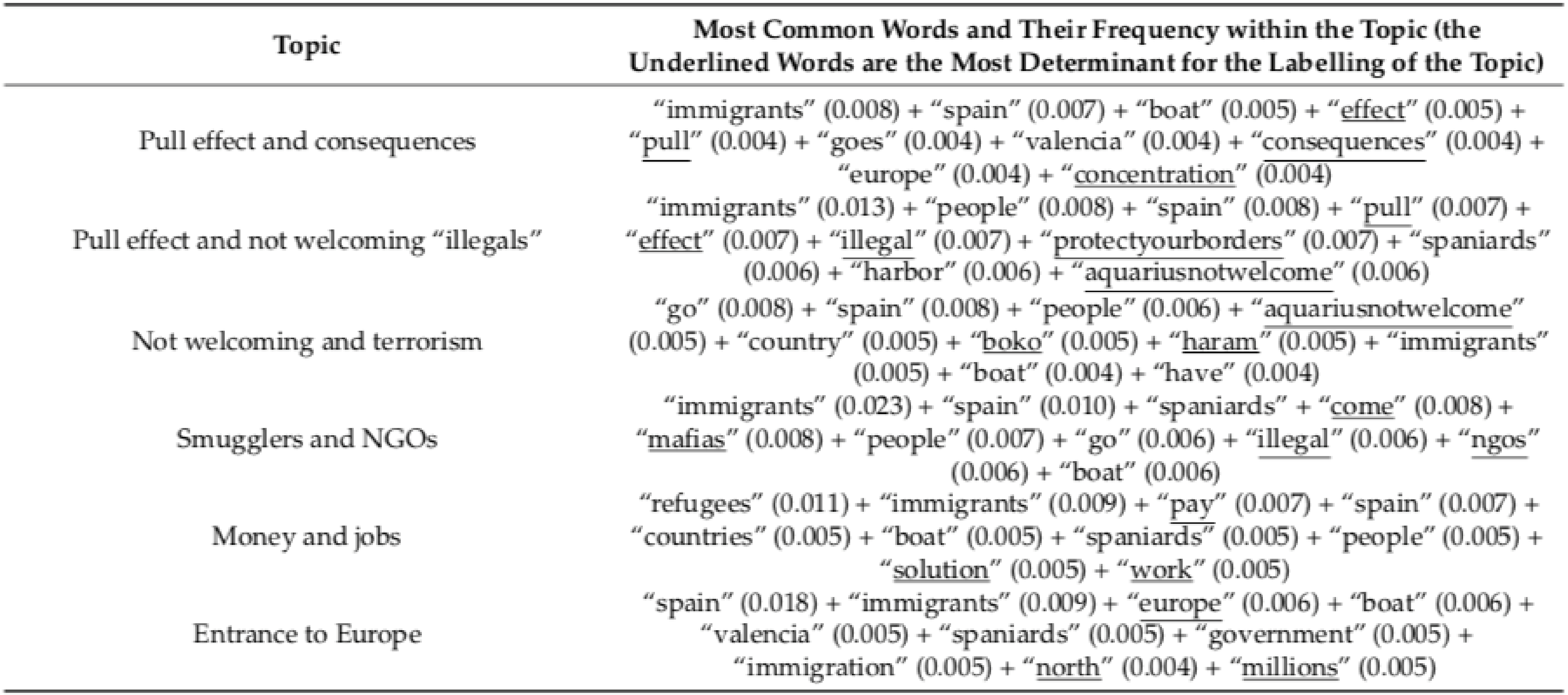
US		Mexico	
Topic	Keywords	Topic	Keywords
1. Intentions and actions of migrants	people, will, american, countri, get, america, want, can, central, make	1.Matters of the borders	border, mexican, central, american, state, honduran, cross, unit, member, chiapa
2. Situations at the border	border, mexico, asylum, tijuana, mexican, children, member, group, hondura, head	2. Government and humanitarian response	mexico, countri, govern, right, enter, human, migrat, ask, nation, pass
3. Welcome or stop migrant caravans	like, immigr, just, say, stop, now, come, one, troop, state	3. Arrival of migrants	will, tijuana, arriv, support, today, hondura, alread, shelter, leav, citi
4. US government actions and politics	trump, news, presid, use, new, report, fund, claim, support, democrat	4. Reactions towards US actions	peopl, want, help, one, say, trump, like, know, can, mani

Tun-Mendicuti, etl al., (2024, May). Understanding opinions towards migrants in transit: An analysis of tweets on Migrant Caravans in the US and Mexico. In *Proceedings of the 16th ACM Web Science Conference* (pp. 1-10).



Mittal, J., et al., (2023, April). Language on reddit reveals differential mental health markers for individuals posting in immigration communities. In *Proceedings of the 15th ACM Web Science Conference 2023* (pp. 153-162).

Hate speech towards migrants/refugees? What are the main topics behind the messages in Twitter in Spain with hate speech against refugees?



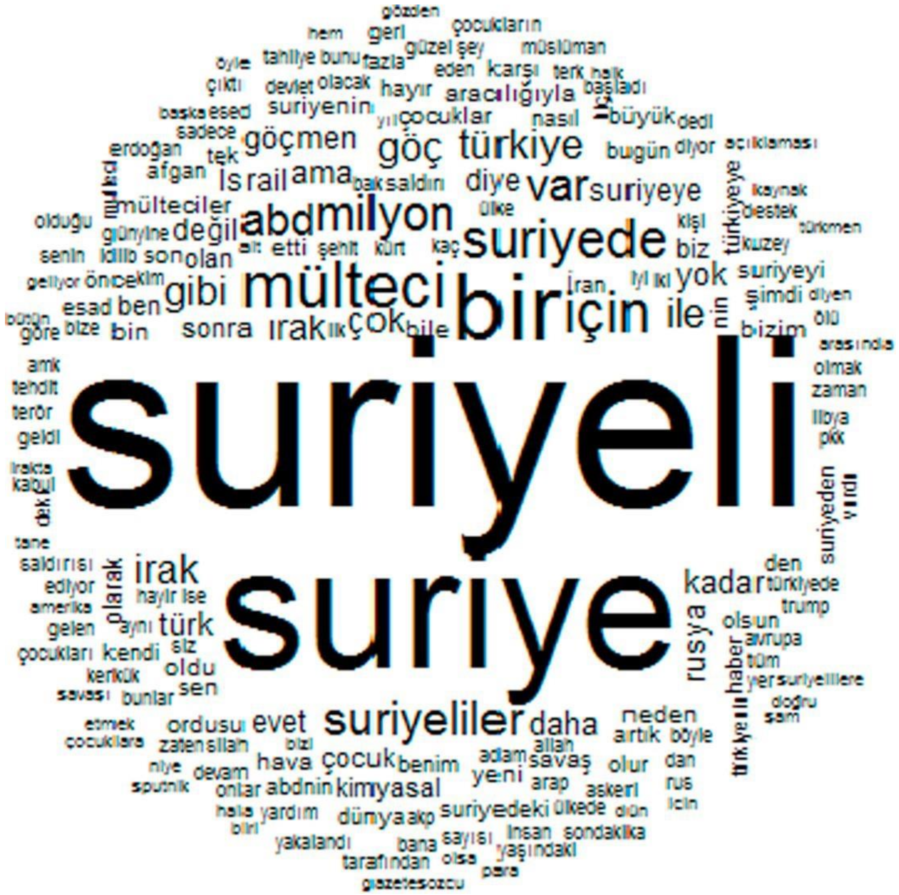
Topic	Most Common Words and Their Frequency within the Topic (the Underlined Words are the Most Determinant for the Labelling of the Topic)
Pull effect and consequences	"immigrants" (0.008) + "spain" (0.007) + "boat" (0.005) + "effect" (0.005) + "pull" (0.004) + "goes" (0.004) + "valencia" (0.004) + "consequences" (0.004) + "europe" (0.004) + "concentration" (0.004)
Pull effect and not welcoming "illegals"	"immigrants" (0.013) + "people" (0.008) + "spain" (0.008) + "pull" (0.007) + "effect" (0.007) + "illegal" (0.007) + "protectyourborders" (0.007) + "spaniards" (0.006) + "harbor" (0.006) + "aquariusnotwelcome" (0.006)
Not welcoming and terrorism	"go" (0.008) + "spain" (0.008) + "people" (0.006) + "aquariusnotwelcome" (0.005) + "country" (0.005) + "boko" (0.005) + "haram" (0.005) + "immigrants" (0.005) + "boat" (0.004) + "have" (0.004)
Smugglers and NGOs	"immigrants" (0.023) + "spain" (0.010) + "spaniards" + "come" (0.008) + "mafias" (0.008) + "people" (0.007) + "go" (0.006) + "illegal" (0.006) + "ngos" (0.006) + "boat" (0.006)
Money and jobs	"refugees" (0.011) + "immigrants" (0.009) + "pay" (0.007) + "spain" (0.007) + "countries" (0.005) + "boat" (0.005) + "spaniards" (0.005) + "people" (0.005) + "solution" (0.005) + "work" (0.005)
Entrance to Europe	"spain" (0.018) + "immigrants" (0.009) + "europe" (0.006) + "boat" (0.006) + "valencia" (0.005) + "spaniards" (0.005) + "government" (0.005) + "immigration" (0.005) + "north" (0.004) + "millions" (0.005)

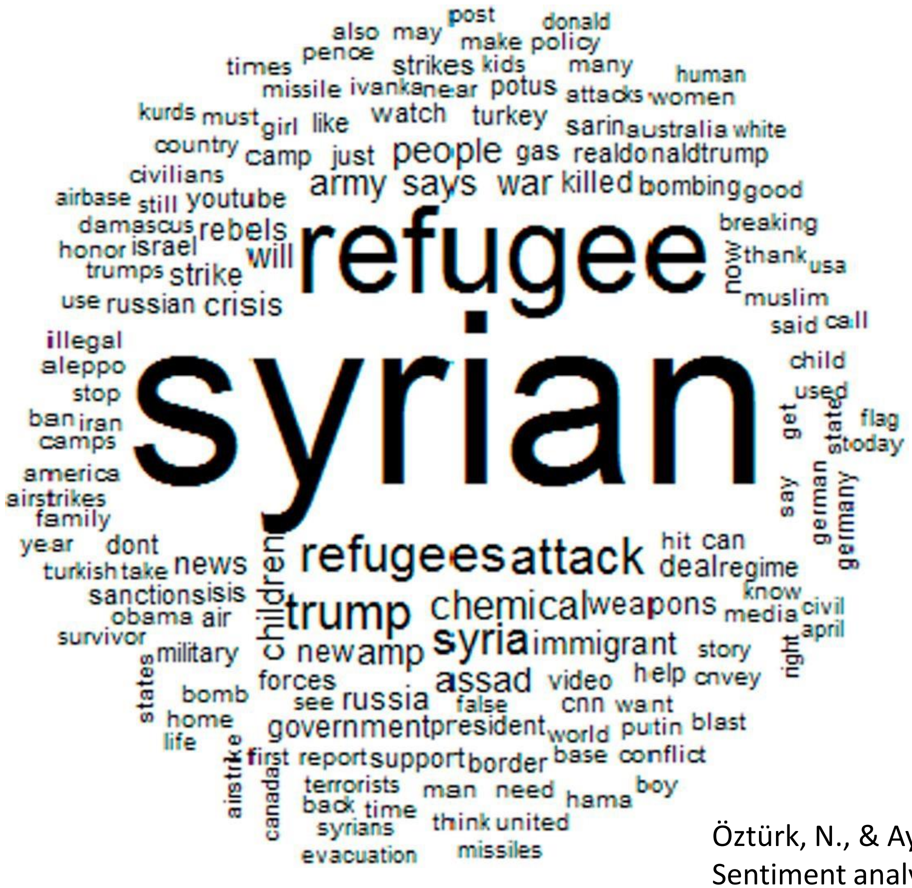
Arcila-Calderón, C., et al., (2021). Refugees welcome? Online hate speech and sentiments in twitter in Spain during the reception of the boat Aquarius. Sustainability, 13(5), 2728.

Explores sentiments towards Syrian refugee crisis in a comparative way.

Turkish data showed more positive sentiment toward refugees in balanced distribution.

English speaking community shared sharply more neutral and negative opinions.





Öztürk, N., & Ayvaz, S. (2018). Sentiment analysis on Twitter: A text mining approach to the Syrian refugee crisis. *Telematics and Informatics*, *35*(1), 136-147.