Lectorsync 1.0

ENHANCING YOUR LECTURE STORAGE USING AI AND NATURAL LANGUAGE PROCESSING

Gelai Serafico, Sebastián Farje, Gabriel Haftel, Pablo Chamorro, Christian Ranon, Alexander Benady

Crucial Talking Points

DISCUSSION FLOW

- WHAT IS IT?
  - Practical Uses
- HOW DOES IT WORK?
  - Data Creation
  - Transcriptor
  - Summariser
  - Translator
  - Classifier
- SHOWCASE
- CONCLUSION

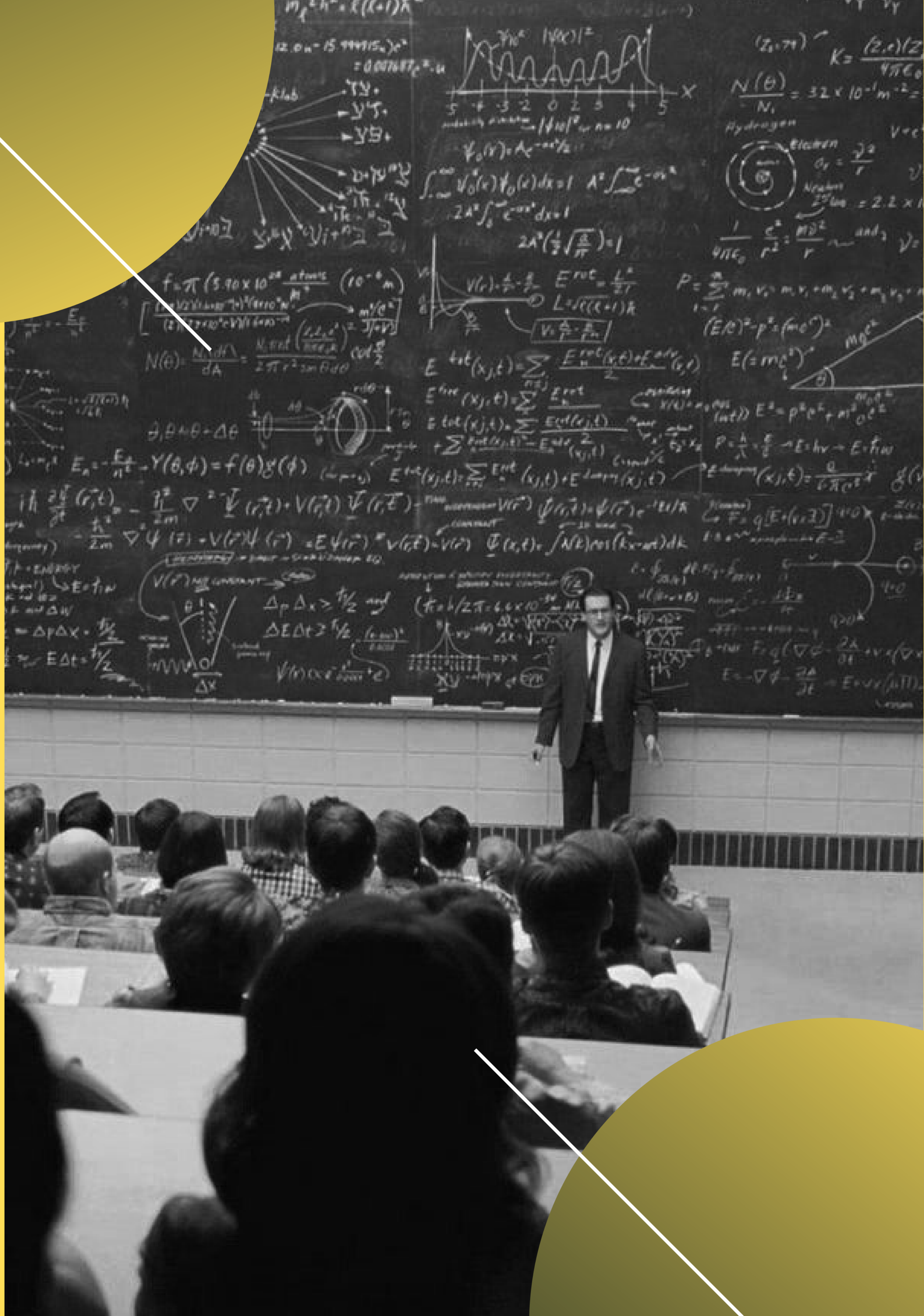
OUR GOAL?

MAKING REVIEWING AND REMEMBERING EASY

Taking concurrent notes can be tiring and distracting.

By leveraging the latest in AI and NLP, we created a system that:

- Transcribes
- Translates
- Summarizes
- Classifies



Practical uses?

BUSINESS CASES

STUDENTS AND EDUCATIONAL INSTITUTIONS

Summarize lectures and notes both for ease in study and instituional oversight.

MULTI-LINGUAL CORPORATIONS AND MULTINATIONALS

Easy sharing of meeting minutes in multiple languages.

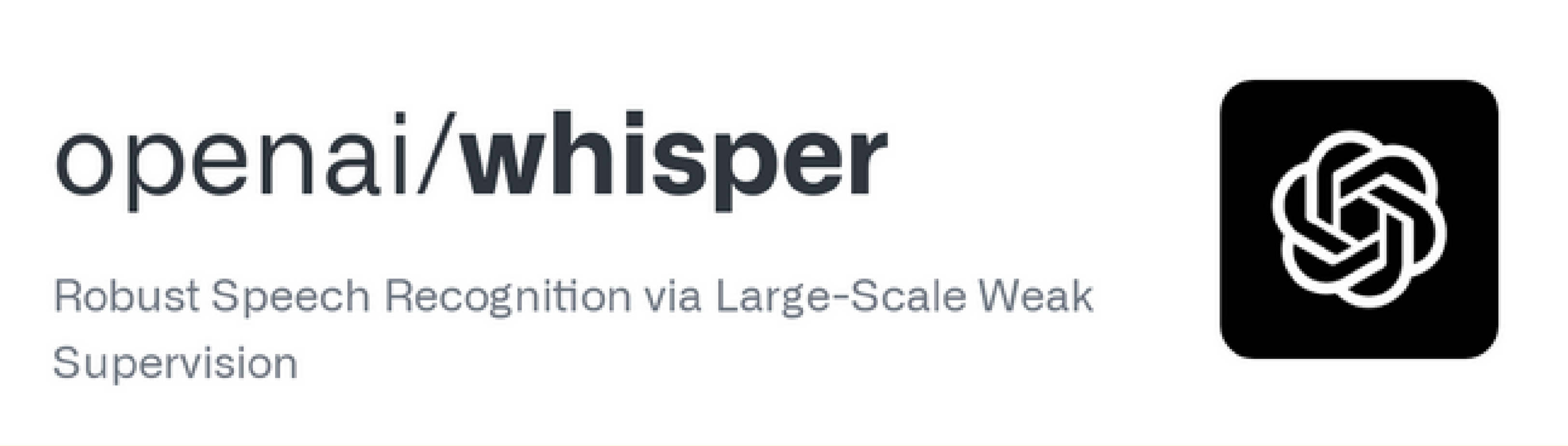
How does it work, exactly?

A TOUR OF THE SYTEM AND
OUR CREATION PROCESS.

Data Generation

- Developed our own **synthetic dataset** due to lack of suitable pre-existing data.
- Selected 1,**000 lecture topics** (five fields, each with eight subjects, each subject with 25 session lectures).
- Used GPT-3.5 API to generate the **text** of the 1,000 unique lecture titles.
- Used GPT-4 to generate 120-word **summaries** for each lecture
- **Translated** summaries into Spanish using DeepL API.

Transcriber



OPENAI WHISPER LARGE V3

Pre-trained model for automatic speech recognition designed not to need fine tuning. Uses seq2seq model base trained on 680k hours of labeled data and further improved on 1 million hours of data.

Other models tested:

- NVIDIA NeMo Canary 1B
- MetaVoice 1B

Summarizer

PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization

Table of Results

Model	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-Lsum	Gen Len
Pegasus	43.8758	20.1981	30.8852	40.6891	78.6333
XLMPROPHET	42.4824	18.5202	32.1804	39.1061	128.0000
Prophet	40.7825	16.0289	30.2197	37.1005	127.6666
MT5	22.1939	13.1137	19.5421	20.8396	20.0000

- Trained on CNN-daily Mail, Fine-Tuned on Our Dataset
- Produces shorter, concise summaries compared to Prophet and XLMProphet.
- Evaluated on the ROGUE metric given the nature of our task (shortening our texts but distilling it into an informationally rich summary).

Translator Fine-tuning Helsinki OPUS MT EN-ES model

HELSINKI MODEL

Originally trained using the amazing framework of Marian NMT and OPUS dataset

OUR FINE TUNING

Trained the model on our own english to Spanish translations of notes summaries. We decided to do simple train to avoid any potential overfitting because th eoriginal model was already performing well and used our own data, because eng-spa available data mainly came from web-crawls in governemnt and legislative sites ->catastrophic forgetting..

```python
from transformers import MarianMTModel, MarianTokenizer

model_name = "sfarjebespalaia/enestranslatorforsummaries"
tokenizer = MarianTokenizer.from_pretrained(model_name)
model = MarianMTModel.from_pretrained(model_name)

src_text = [
    "By understanding Kafka's core concepts and architecture, you'll be well-equipped to leverage its capabilities in your projects.",
    "Thank you for your attention, and I'll now open the floor for any questions you may have."
]

# Prepare the text data into the format that the model expects
tokenized_text = tokenizer(src_text, return_tensors="pt", padding=True)

# Generate the translation using the model
translated = model.generate(**tokenized_text)

# Decode the translated text
for t in translated:
    print(tokenizer.decode(t, skip_special_tokens=True))
```

```
generation_config.json: 100% 288/288 [00:00<00:00, 11.5kB/s]
Al comprender los conceptos y la arquitectura fundamentales de Kafka, estarás bien equipado para aprovechar sus capacidades en tus proyecto
Gracias por su atención, y ahora abriré la palabra para cualquier pregunta que pueda tener.
```

Gradio App Showcase

Conclusions: Challenges & future improvements

CHALLENGES

- Had to create synthetic data
- Computing Resources
- Texts for summarizer training where too short. Transcriptions for actual lectures are longer

FUTURE-IMPROVEMENTS

- Use real anotated data
- Add a determined structure to the summary
- Apply more languages


注数据
'passages": [{
"p":"鸦片战争博物馆始建于1957年。",
"a":"1957年"}],
"q":"鸦片战争博物馆始建于什么时候？" }
nnotation
'passages": [{
"p":"The Opium War Museum was built in 1957.",
"a":"1957"}],
"q":"When was the Opium War Museum first built?" }