

An Econ Use Case for AI / ML

From Historical Newspapers to Strike Data

Attila Gáspár

ELTE Centre for Economic and Regional Studies

AI Seminars / December 2 2025.

Motivation

- Show a concrete **economics use case** for AI / ML
- Use case: historical political economy
- Goal: go from **raw scans** to a **clean strike dataset**
- Focus: what AI / ML tools help, and how much work it takes



Example Project (Példaprojekt)

Topic: The impact of agricultural mechanisation (mezőgazdaság gépesítése) on political behaviour.

- Long-run question: how technology changes **conflict** and **collective action**
- Data source on mechanisation:
 - 1895 “Gazdacím-tár” (Hungarian agricultural directory)
 - Processed for us by collaborators
- Outcome: **strikes** (sztrájkok) by locality and year

The Econometric Task (Simplified)

- For the talk: think of a simple regression
- Dependent variable (DV): indicator for a strike in a locality-year
- Key independent variable (ID): measure of mechanisation from 1895 directory
- Control variables, fixed effects, etc. are **not** the focus here
- The focus is: **How do we get the strike data at scale?**

Data Product: Strikes in Népszava

- We build a dataset of **strikes** from Népszava daily newspaper
- Each observation: one strike event, with
 - Date
 - Location (mai helyszín neve)
 - Industry code (iparági kód)
 - Occupation code (foglalkozás kód)
- Final goal: merge this with mechanisation and other historical data

Népszava pages



ELŐFILETÉSI DÍJ:
a kedvi, országos és szombati bérháza:
Árakat 14.450,-, 15.000,-, 16.000,-, 17.000,-, 18.000,-
Távolsági árakat 14.450,-, 15.000,-, 16.000,-, 17.000,-, 18.000,-
Gyorsítás a csatornához bérháza:
Árakat 14.450,-, 15.000,-, 16.000,-, 17.000,-, 18.000,-

Az elnököt ér Julius havában a budapesti vörösvízi kisgyűlést többmárti nyilvános vasúti munkás-gyűlés meghirdette a szakszervezet törleszt, hogy az elhangzott pántaszaktól irányba fogalva, a kerületi miniszter el jönne. A politikai válság folytán az egy oldalról hosszú ideig lebetszen volt, és így a szakszervezeti tanács csak most vont abbah a helyzetben, hogy a kezességi gyűlést megelőzzen.

Estét sudrezsés hő 6-án, délután
2 órakor, Budapesten a Tattersallban
(Közép-kerepesi-ut 7. sz. alatt)

országos
vasuti munkás-gyilkos-
tartások. **Hipirend:**
1. június 1-én minden vasúti munkásra.
2. Joggal szolgálhatnak a vasúti munkások
kölcsönökükben.

A „megbízhatók.”

A „megbízható” minősé — tartja a gyors — a beszéged, a színjátékod, a megennyúszerűd, a társszín nem töred, a mosásod, ami nagyobb kincsílmányod a

gyűjtem. A „magasított” – a tár-
hársnyi gyök – az a munka, aki
szükséges a felfoghatóságot, aki
szükséges a hosszú és a keresztet,
szükséges az osztályt az ordig, még mi
nem. „Megbízhat” – az a munka-
sorja Pöcsey, a munkások barátja
kik hasít. „Megbízhat” – mondja
az örököslő munkás – csak családtagok
valóbanak, emeljük az arculatot, örökké
megfoghatók maradékukon esélyük
gyar rökkáru függetlensége mellett
láthat. A gyáraknak pedig csak

Budapest, 1904. kezd. március 1.

XII. évfolyam.

NÉPSZAVA

ELŐFILETÉSI DÍJ:
a kedi, cesttori és személyi bérletekre:
Igényre 14.400,-, 12.000,-, 10.000,-, 8.000,-, 5.000,-, 3.000,-, 1.000,-
személyszállítási igényre 10.000,-, 8.000,-, 6.000,-, 4.000,-, 2.000,-
Csoportos a cesttori bérletekre:
Igényre 4.000,-, 3.000,-, 2.000,-, 1.000,-

szükséges, hogy a munkahelyet elérni. Néha jogosultság a miniszteri szinten. Szíves, ha a tanács jókat akar, az titkos jóváhagyást.

Ezután tudja, hogy a „perelhető” vagy „nem” csak érdekes, amely mindenben tükrülhet a politikai helyzetben. Nyilván körülönbelül, hogy „nem” a meghosszabbításnak megfelelően nem engedélyezhető. Aztán a tanács eldönti, hogy a jogosultság a rendelkezésre álló időszakban lehetséges-e a meghosszabbítás. Ez a rész a tanácsnak a jogosultságban történő részletezéséhez köthető. Táncsics László, garantára de szemmelvételben részt vevő 1926-ban írt az akkor jogszabálytól eltérően:

„Álarcot vesz tehit, hogy eltársa, annyi nam hogy puulni se tudjon s megalakulják. Igen felszínűek lesznek azok, akik a statisztikai méréseket követően megállapítják, hogy mindenki körülük van hibásító, de nem mindenki hibásítja őket.” (Hetzl 229.)

és a haszifával), az olcsottnak és enyhéen drámaiknak tekintettek. 1823-as könyvben először az ország, hogy vezeti ferfai, "bölcsítői" minőségeket vagy tanítók szerepét a franciaok elismerések elutasították. A művészeti tevékenységek és az oktatás jogai jóval korábban meghatározottak voltak, mint a politikai jogok. Az 1848-as forradalom után minden bizonymával, ha valós energiával, hogy ezt a magasrangúat a nemzeti jogrendszerben helyezze el.

Egyetérzetet mindenki! Ebből a hír-országbeli magis bel csinálnak Európát. Mondjuk mi, a „felfüggesztés és a használhatóság”

An Econ Use Case for AI / ML

AI Seminars / December 2 2025.

6 / 25

44. *materias* 8.

MEPESZAV

五
八

A smaragdos minden hőszig előt pénzjárati tárca állt. A legmagasabb támogatás f. 100 millió Ft. Azonban az adózás után a támogatás f.

Várhelyihez csapott. A csatolt ügynök, hogy csak írt a hírben, hogy a kormány elutasította a kölcsönök megtérítését, de föld magának hengér kirendeléssel. De ekkor már két hal a szenátusban, aki nem támogatta az összesen 10 millió forintos kölcsönök megtérítését, mert mindenki szerint ezeket nem kell visszatéríteni.

AKMOZGAL

What is Népszava?

- Socialist daily newspaper in Hungary
- Important for labour movement and workers' politics
- Rich coverage of **strikes, labour conflicts, and politics**
- Digitised and available via **Arcanum** (online archive)

adt.arcanum.com – Népszava

Historical Context

- Period of interest: around **1904**
- Political crisis and intense social conflict
- **Arató-sztrájkok** (harvest strikes)
- Our project:
 - Use AI / ML to systematically extract these events
 - Connect them to mechanisation and other local characteristics

Step 0: Download PDFs

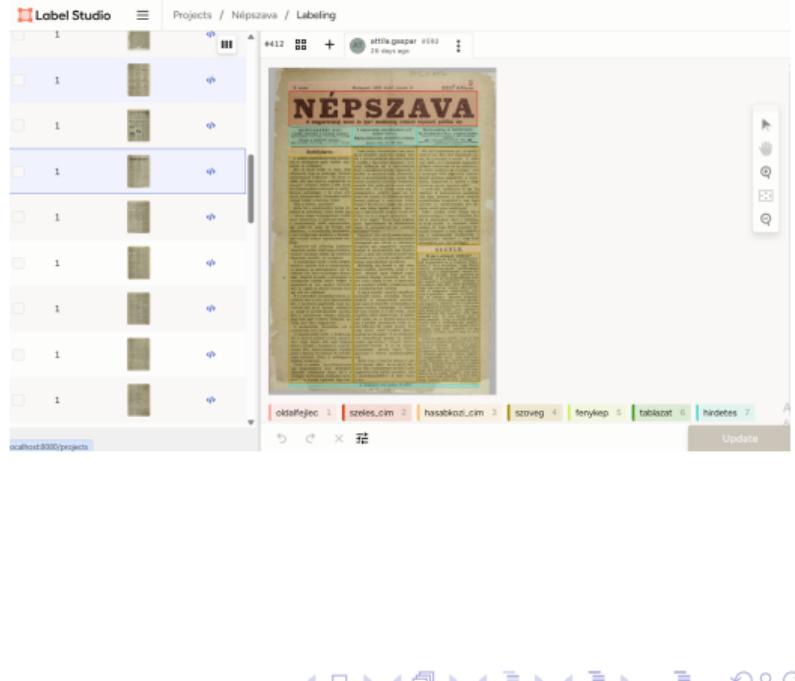
- Download relevant issues from **Arcanum**
- Could be automated via scripts or APIs
- In practice: for this prototype, we did it manually (shoutout to Mátyás Tamás)

Layout Model Training: Overview

- Need to detect **sections** (rovatok) in the newspaper pages
- Why?
 - Articles about strikes appear in specific sections
 - We want to extract only these relevant sections
- Approach:
 - ① Generate training data
 - ② Label sections (rovatok)
 - ③ Train a **layout model** (LayoutParser)

Generating Training Data

- Take about **50 pages** of *Népszava*
- Convert PDF pages to **JPG** images
- Use **LabelStudio** to annotate:
 - Headers (fejlécek)
 - Section titles (rovatcímek)
 - Main text blocks
 - Advertisements (hirdetések)
- Output: labelled images + JSON annotation files



What is LayoutParser? (1/2)

- Python library for **document layout analysis**
- Built on top of modern computer vision models
- Can detect:
 - Paragraphs, titles, figures, tables
 - Arbitrary custom classes (e.g., newspaper sections)
- Works well with historical documents after **fine-tuning**

LayoutParser

What is LayoutParser? (2/2)

- Use the **LayoutParser Trainer** on a Linux server
- Create a **Docker container** that:
 - Installs all dependencies
 - Mounts training data
 - Logs model checkpoints
- Training details (example):
 - Runs on Koren Miklós's server
 - Nominally ≈ 8 hours, but convergence is faster

Running the Layout Model on All PDFs

- Upload **all** relevant *Népszava* PDFs to the server
- For each page:
 - Convert to image
 - Run the trained layout model
 - Save predictions
- Output per page:
 - JSON with detected blocks:
 - header, section title, text, ads, etc.
 - JPG image with bounding boxes

Parsed Page and JSON Output



```
{ "label": "szoveg",  
  "points": [  
    [ 229.65077209472656,  
      28.679264068603516  
    ],  
    [ 1156.51708984375,  
      4370.50244140625  
    ]  
  ],  
  "group_id": null,  
  "shape_type": "rectangle",  
  "flags": {},  
  "column_number": 1,  
  "row_number": 1  
},  
{ "label": "oldalfejlec",  
  "points": [  
    [ 279.60626220703125,  
      124.34354400634766  
    ],  
    [ 2837.328369140625,  
      202.19850158691406  
    ]  
  ],  
  "group_id": null,  
  "shape_type": "rectangle",  
  "flags": {},  
  "column_number": 0  
}
```

Python Tool: From Blocks to Text

- Write a Python tool that:
 - ① **Adjusts bounding boxes**
 - Clean up slightly “hand-drawn” boxes from the model
 - ② Runs **OCR** (Tesseract) on:
 - Text blocks
 - Headers
 - Section titles
 - ③ Determines the correct **reading order**:
 - Which column?
 - Top-to-bottom ordering within columns

Python Tool: Exporting the Output

- After ordering the blocks, we can:
 - Export a **continuous text** version per section
 - Save detailed **JSON**:
 - block type, coordinates, OCR text
 - page number, issue date
 - Optionally save **plain text** (TXT) for quick inspection

LLM Cleaning Tool in Python: Overview

- Next step: filter and structure the text using **LLMs**
- Python pipeline:
 - ① Iterate over section titles (rovatcímek)
 - ② Use a **cheap model** to classify section type
 - ③ Use a **more expensive model** to extract strike events
- Output: strike events in a **consistent JSON schema**

LLM Cleaning: Details

- Step 1: cheap model
 - Check if section title is e.g. “*Tőke és munka*” or “*Földmívelés*”
 - If yes, save the section content for detailed analysis
- Step 2: expensive model
 - Ask: Are there **strikes** in this section?
 - If yes, extract each strike as JSON:
 - modern place name, place code, industry code, occupation code, date, etc.

Final JSON

```
{  
  "event_date": null,  
  "industry_txt": "shoe manufacturing at Moskovits Farkas shoe factory",  
  "industry_SIC": "3140",  
  "participants_txt": "workers of Moskovits Farkas shoe factory in Nagyvárad",  
  "participants_ISCO": "8150",  
  "firm_name": "Moskovits Farkas shoe factory",  
  "location_txt": [  
    "Nagyvárad"  
,  
    "location_official": [  
      "Oradea"  
,  
      "location_geonames_id": [  
        671768  
,  
        "strike_status": "ongoing",  
        "description_en": "Workers at Moskovits Farkas shoe factory in Nagyvárad are on strike"  
      },  
    ]  
  ]  
}
```

Geocoding and Mapping

- LLM output:
 - Good at **modern place names**
 - Weaker on **geo-codes**
- Solution:
 - Use **GeoNames API** to geocode place names
 - Combine LLM guesses with external geospatial database
- Final step: last Python tool:
 - Generate a **HTML map** with all strikes
 - Interactive: click on a point for details

Github repo (with map)

Tools Used (AI / ML Stack)

- **VS Code + GitHub Copilot**, Claude Sonnet for coding help
 - Cost: 100 USD / year
- **LayoutParser**
 - Cost: free (open source)
- **Tesseract OCR**
 - Cost: free (open source)
- **GPT-4.1** and **GPT-5.1** via API
 - Cost: 10-15 USD / run on 4 full years worth of strike data
- **ChatGPT Plus** for presentation help
 - Cost: 20 USD / month

- **Labeling**

- Annotating ~50 pages in LabelStudio
- One-off cost, but essential for model performance

- **Coding** (mostly vibe coding)

- Layout model training scripts, Docker setup
- Python tools for OCR, reading order, LLM calls, geocoding, mapping

- **Testing and debugging**

- Checking a sample of pages and strike events by hand
- Iterating on prompts and model settings

Takeaways

- AI / ML allows us to:
 - Turn **historical PDFs** into structured datasets
 - Scale up research on **political behaviour** and **labour conflict**
- But:
 - Human input (labeling, checking) remains crucial
 - Infrastructure and coding still take time
- Next steps:
 - Refine geocoding and coding schemes
 - Run the full econometric analysis
 - Generalise the pipeline to other newspapers / countries

Thank you!



Github repo of project