



# Szövegosztályozás Spark klaszterrel az ELKH Cloudon



Kacsuk Zoltán

HdM IAAI kutató, TK CAP  
és TK POLTEXT külsős  
kutatási partner



# Research context

Intersection of two research projects at the Centre for Social Sciences, Institute for Political Science:

- **Text Mining of Political and Legal Texts (POLTEXT) Incubator Project** (Principal Investigator: Miklós Sebők)
- **Hungarian Comparative Agendas Project (CAP) project** (Project leaders: Zsolt Boda & Miklós Sebők)

# Research problem

- **Quantitative analysis** of qualitative data
  - Classifying articles according to policy topics
  - Topics: education to defense, Comparative Agendas Project
- **Gold standard:** double-blind human coding by well-trained researchers
- What if this is **unfeasible?**
  - Article counts of over 100.000
  - **Cost and training** of human coders for this scale



# The project

- **Hungarian country project** of the Comparative Agendas Project ([cap.tk.mta.hu](http://cap.tk.mta.hu))
  - Media module – 3 daily newspapers
- For this pilot project:
  - Left-liberal **Népszabadság (NS)**: Over 50 000 front-page articles (1990-2014)
  - Centre-Right **Magyar Nemzet (MN)**: 35 021 articles (2002-2014)
- Hand-coding was unfeasible for our purposes
- Solution: **text mining + machine learning**

# A machine learning solution

- **Text as Data** – qualitative data is converted to quantitative (matrices)
- How to categorize articles into pre-defined classes: **Dictionary-based** or **supervised learning**
- For the latter a sufficiently **large human-coded *training/test set*** is needed



ELKH Cloud

Part 1

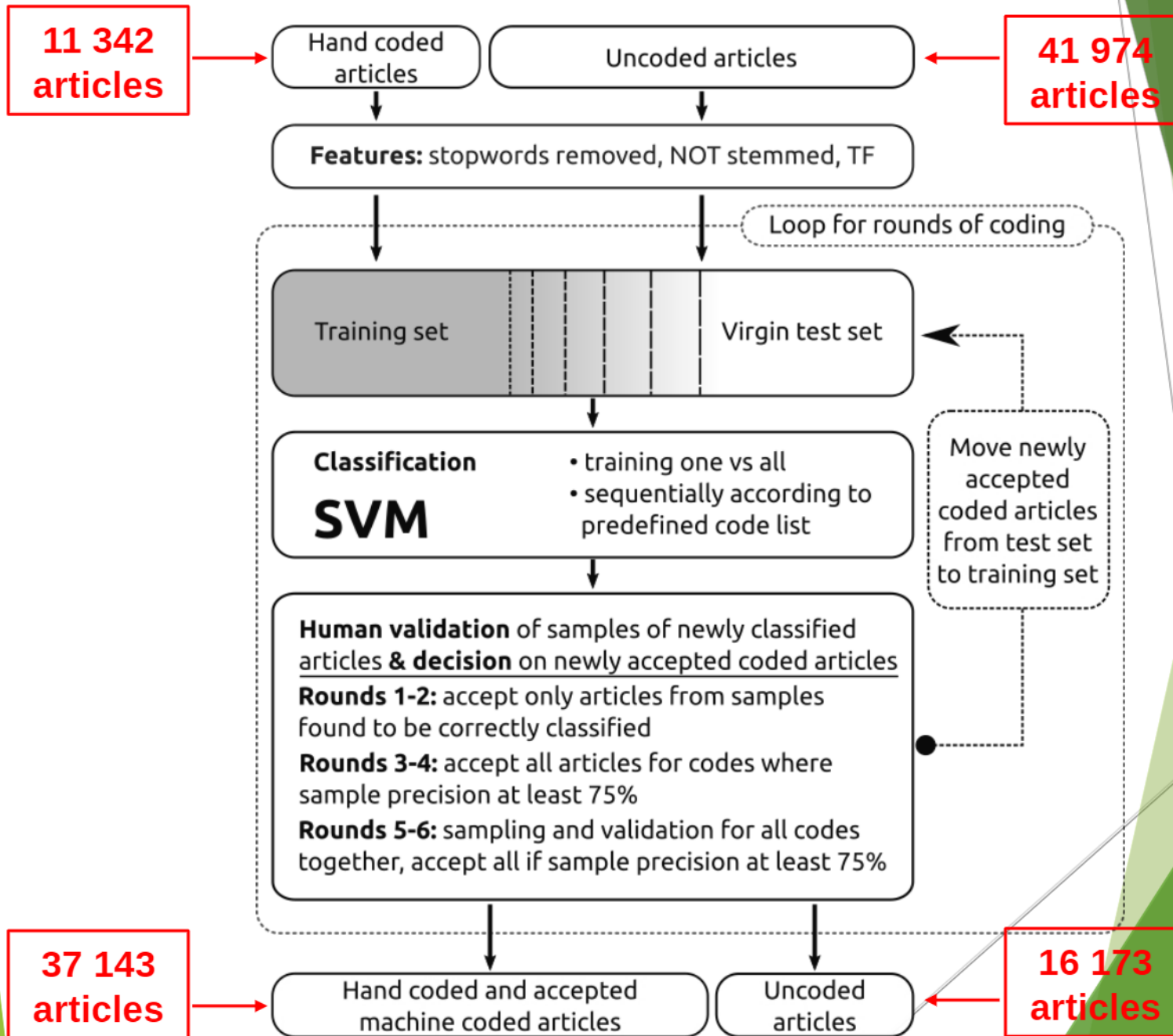
# **CREATING A MACHINE CODED TRAINING SET FOR THE LEFT-WING DAILY NÉPSZABADSÁG (NS)**

# The Hybrid Binary Snowball (HBS) process

- We need to keep human **coding costs as low as possible**, while extracting the largest possible gain per invested human coding hour
- We simplify multi-class classification by rephrasing it as a **series of pairwise comparisons**
- We apply a snowball method to **augment the training set with machine-classified observations**



# Coding NS articles



# Infrastructural bottlenecks 1:

## Memory

- Our desktop workstation had **only 32 GB** RAM
- Encountered **problems**:
  - Could not work on the **whole virgin data** set
  - Could not run **certain configurations**, for example: Term frequency - Inverse document frequency (Tf-Idf) weighting
- Even the **solutions were problems**:
  - Virgin data was partitioned up for processing
  - This would impact Tf-Idf weighting significantly
- **Real solution** going forward:
  - Using larger capacity single virtual instances or a **cluster in the cloud**

# Infrastructural bottlenecks 2: **Time**

- **Huge numbers of small operations** add up quickly
- If process runtimes become too long, project execution becomes unfeasible
- Solution: **parallelizing** the execution of operations

Part 2

**USING THE CODED LEFT-WING DAILY ARTICLES  
TO TEACH THE ALGORITHM HOW TO CODE THE  
CONSERVATIVE DAILY MAGYAR NEMZET (MN)**

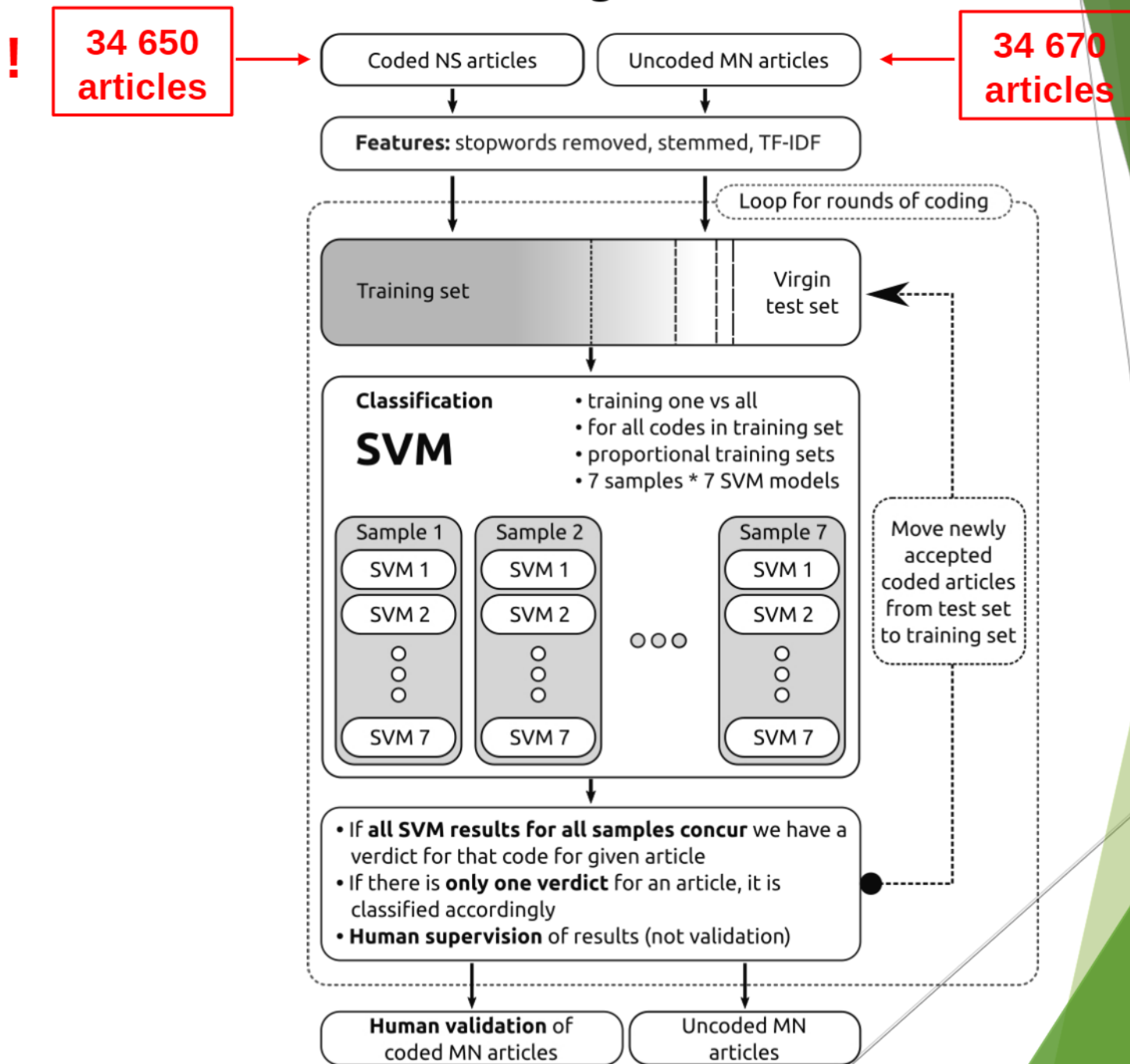
# Apache Spark cluster

- With the help of the Laboratory of Parallel and Distributed Systems at the Institute for Computer Science and Control (**SZTAKI LPDS**)
- Apache Spark cluster running on five virtual instances in the **SZTAKI ELKH Cloud**
- All five virtual instances had 8 virtual processors and 32 GBs of RAM each, and were running Ubuntu 16.04.
- Four instances acted as worker nodes and one as the master node of the Spark cluster. Each Spark session was running with 32 VCPUs (but **default parallelism set to 24**) and **96 GBs of RAM** total on the four worker nodes combined.

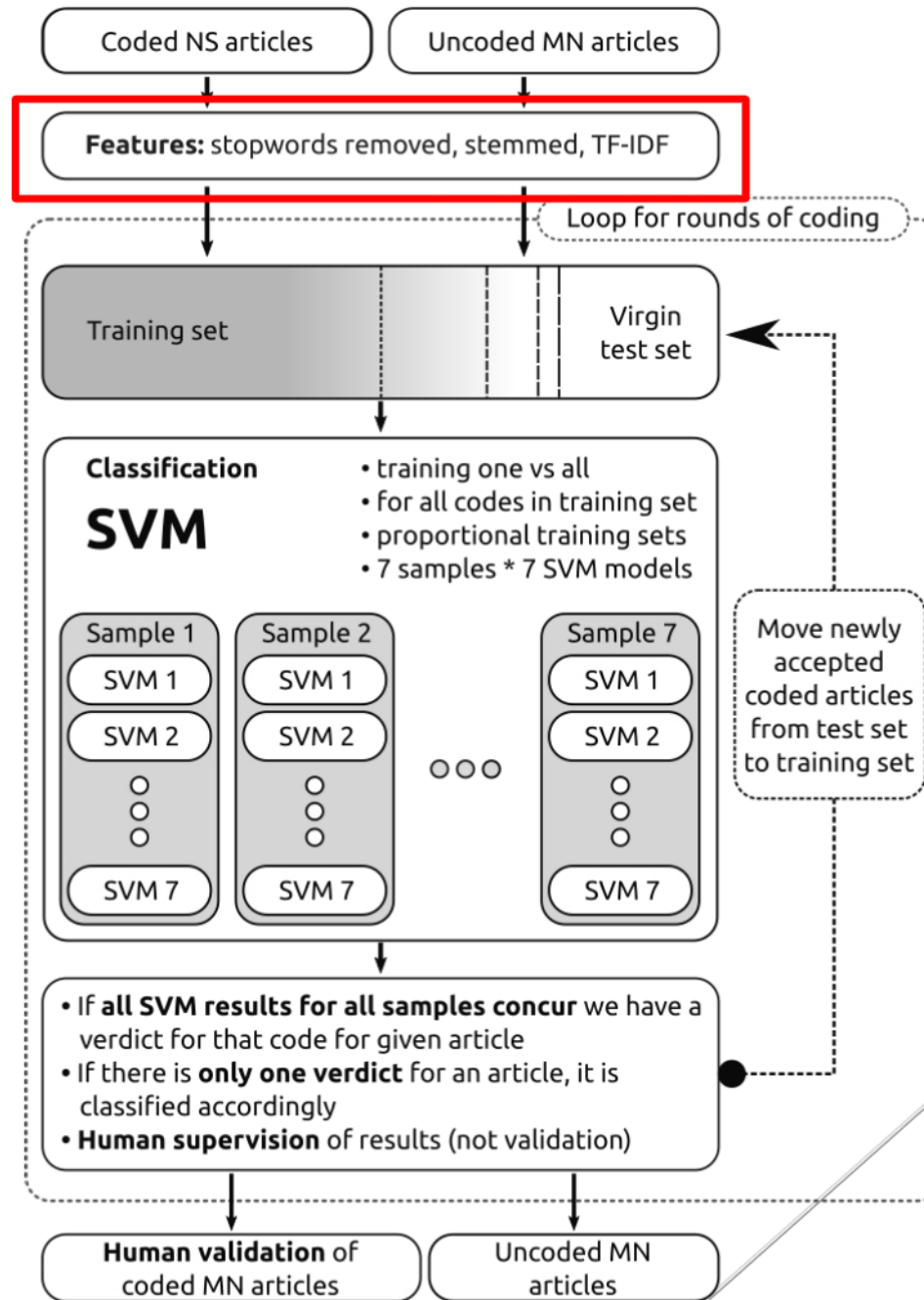
# Manifold increase in speed

- **Old desktop setup:** roughly **3 days** for a full round of coding (33 code categories)
- **Spark cluster:** ca. **30 minutes** for a full round of coding
- This increase in speed enabled:
  - **1) Rapid prototyping**
  - **2) Complex classification workflow**

# Coding MN articles

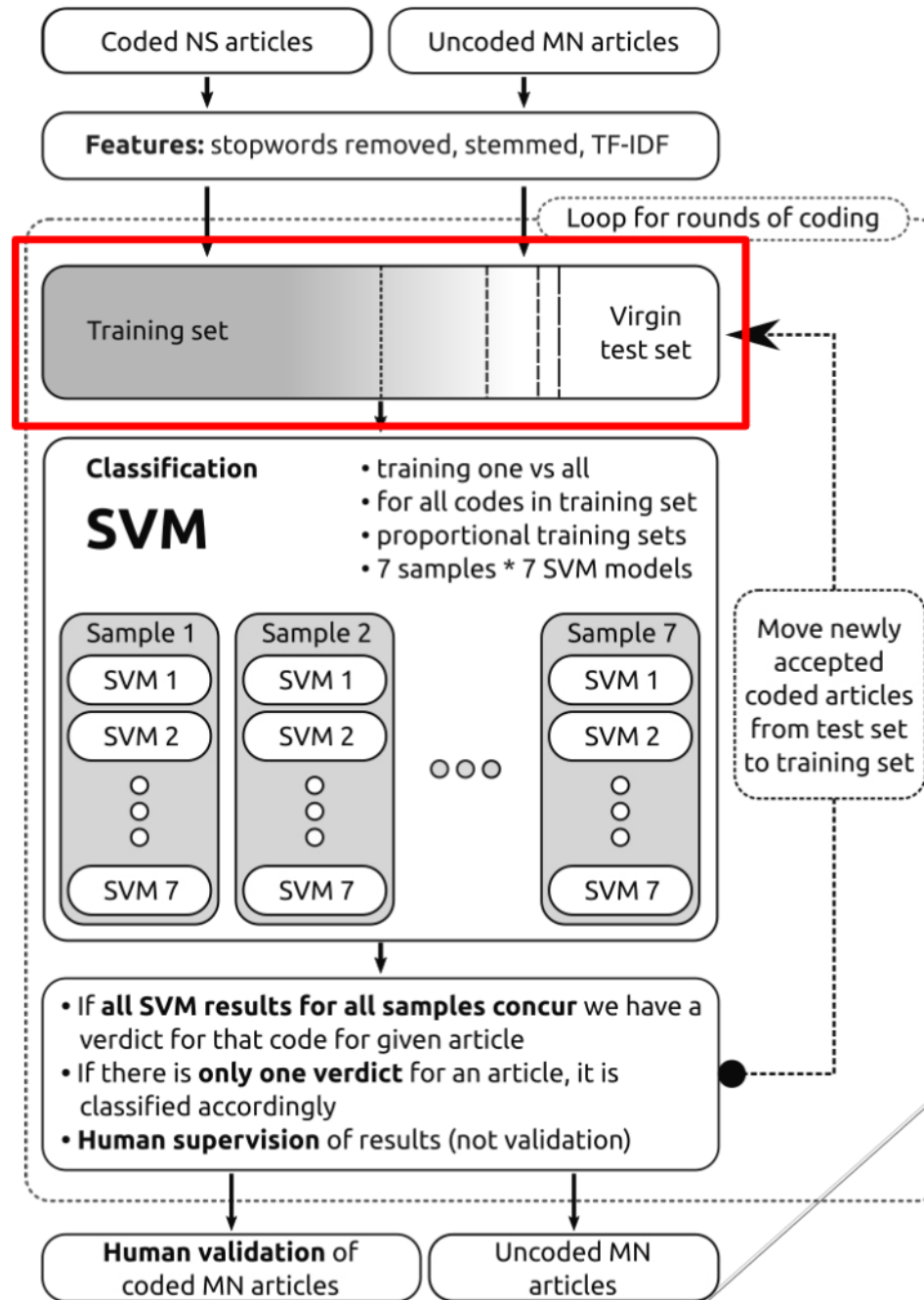


# Coding MN articles

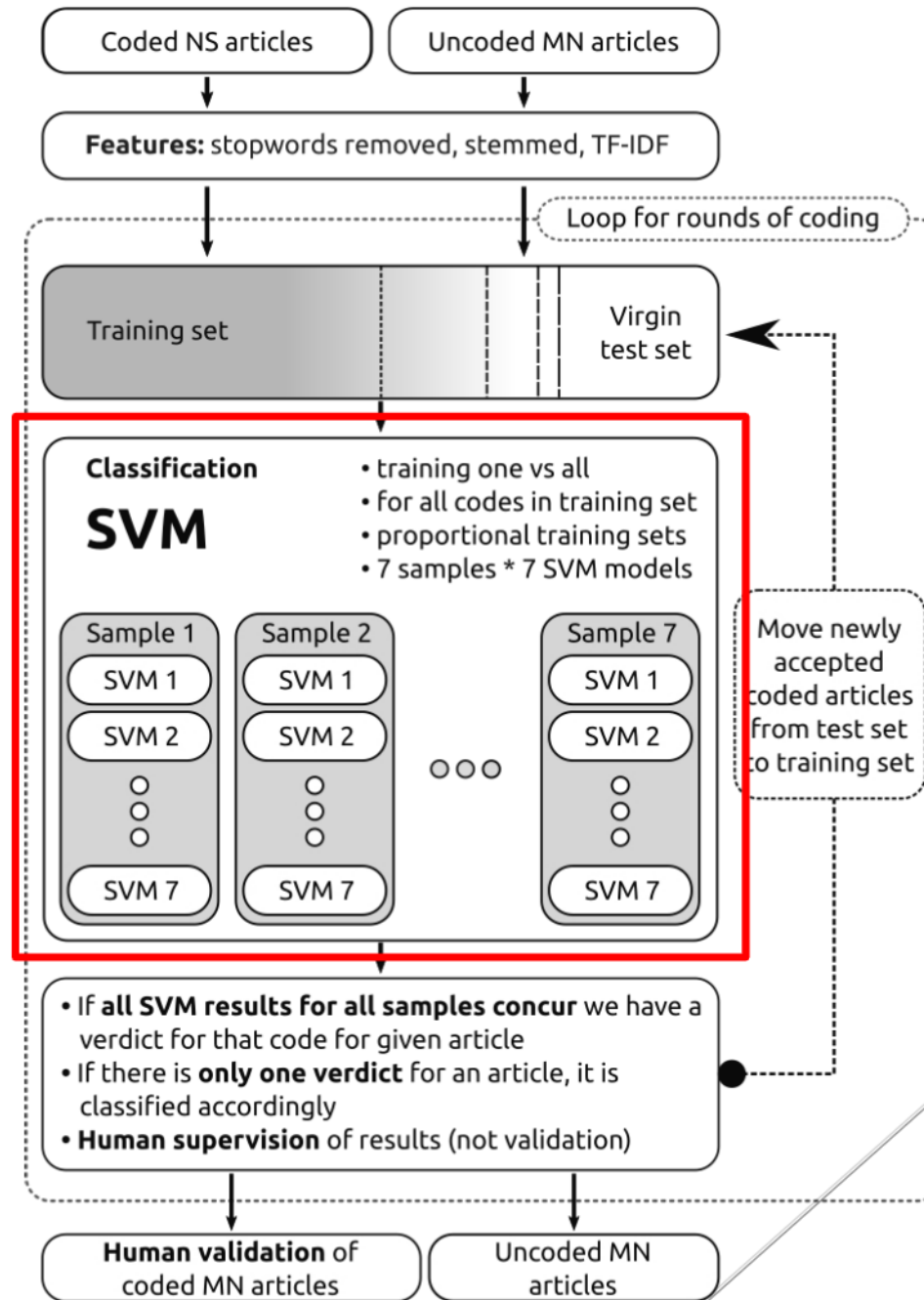




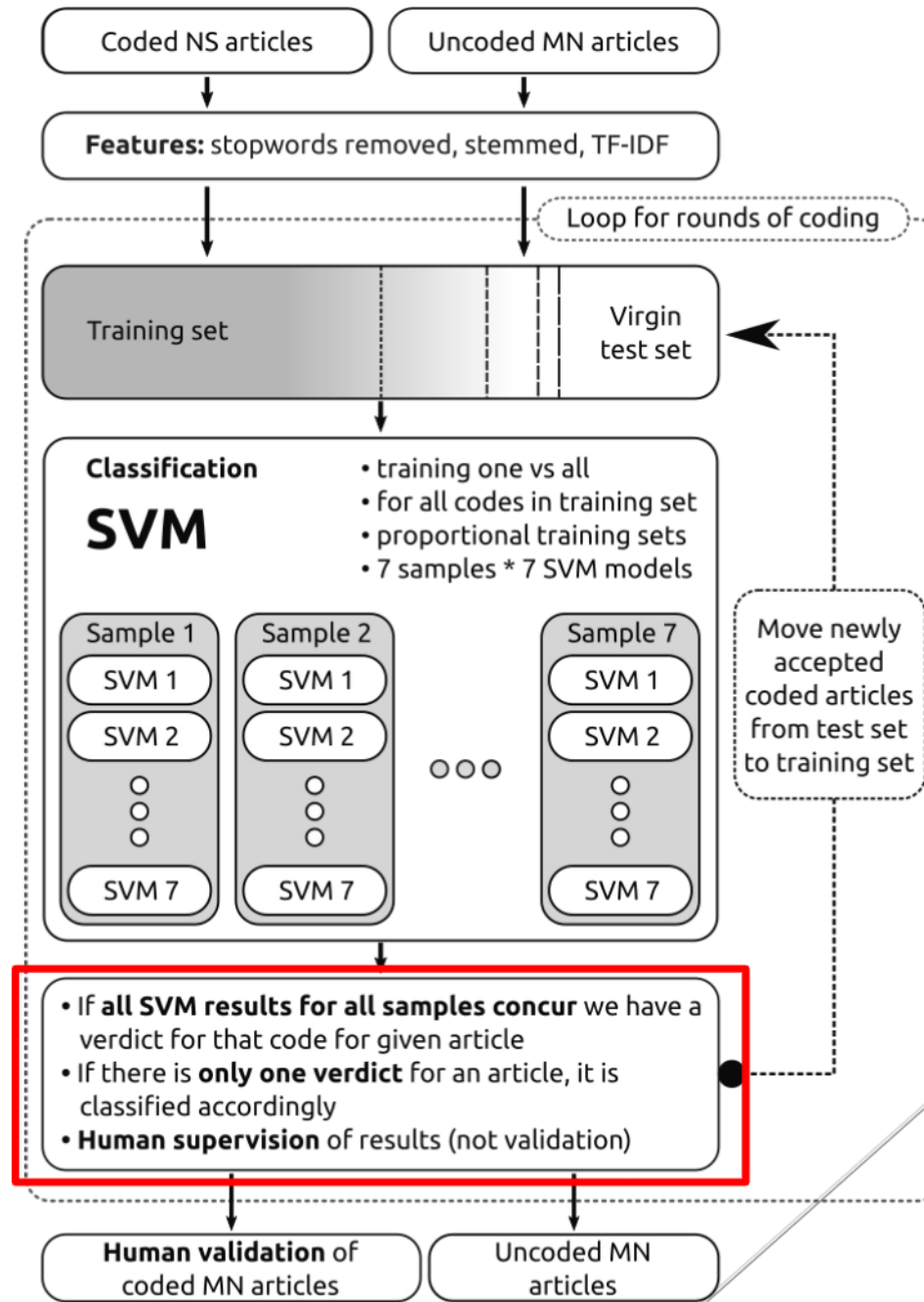
# Coding MN articles



# Coding MN articles



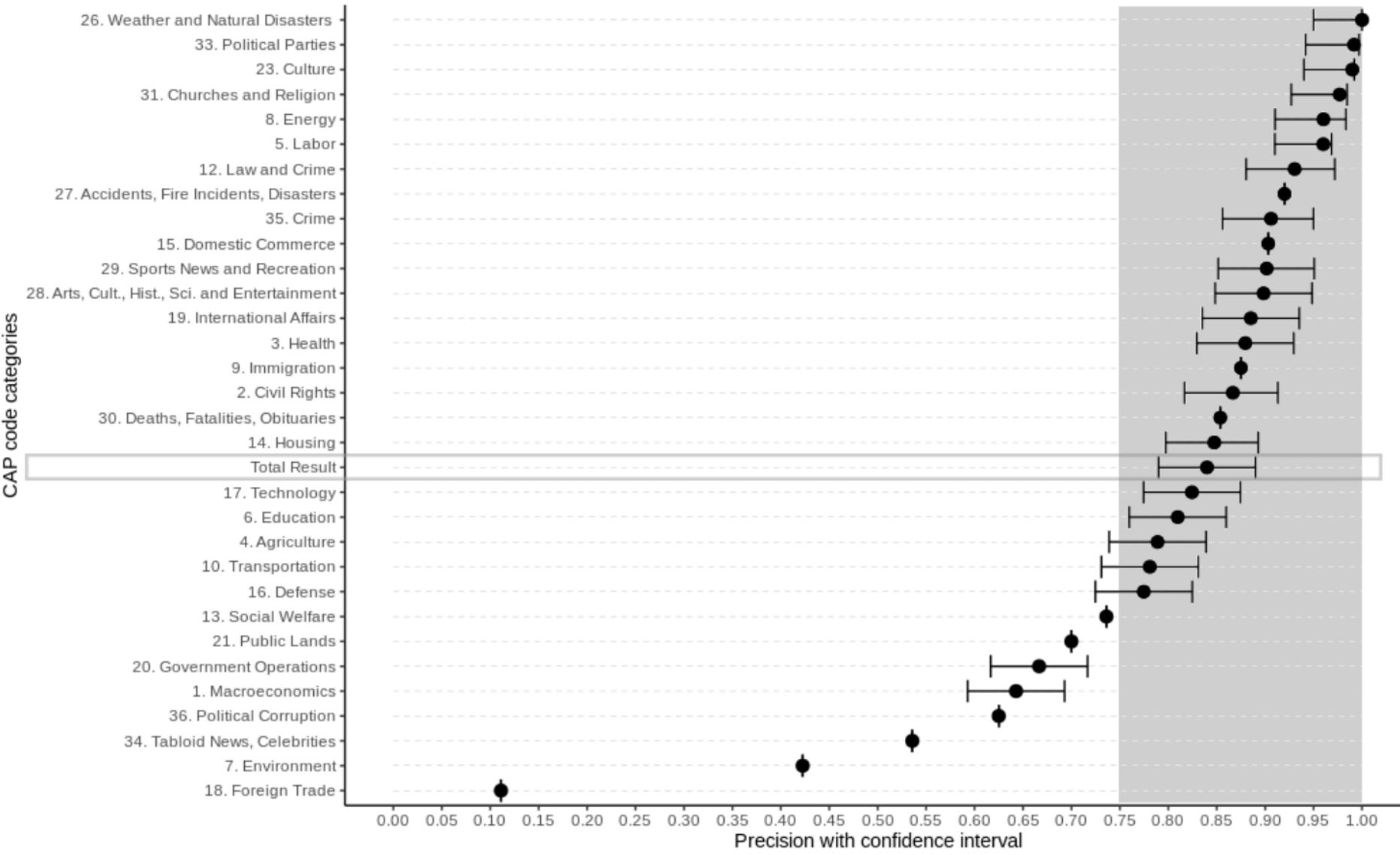
# Coding MN articles



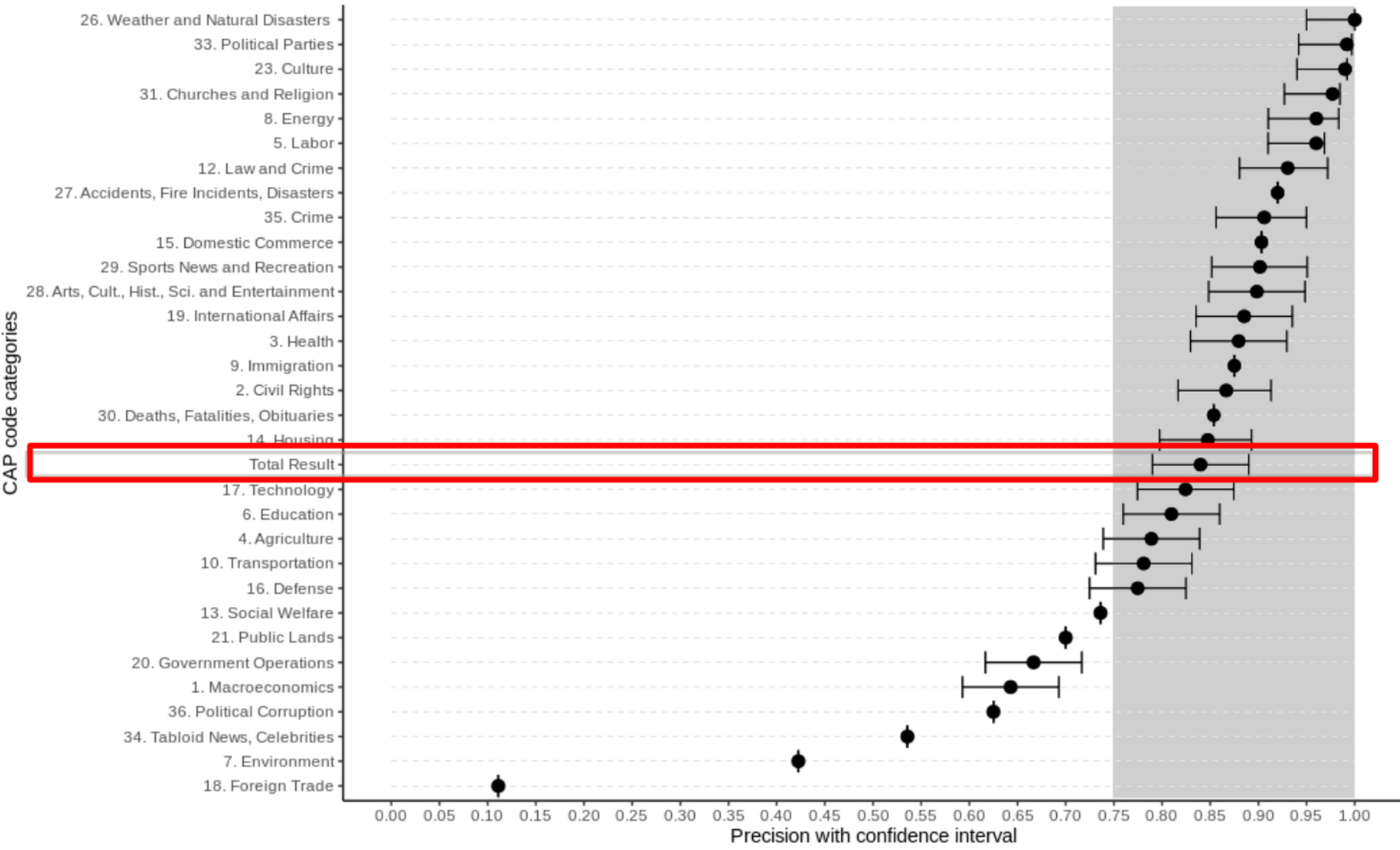


Major Topic	Coded Articles	Sample Size	Precision
1. Macroeconomics	3833	350	0.64
2. Civil Rights	207	135	0.87
3. Health	700	249	0.88
4. Agriculture	408	199	0.79
5. Labor	127	100	0.96
6. Education	436	205	0.81
7. Environment	71	71	0.42
8. Energy	542	226	0.96
9. Immigration	8	8	0.88
10. Transportation	459	210	0.78
12. Law and Crime	570	230	0.93
13. Social Welfare	72	72	0.74
14. Housing	168	118	0.85
15. Domestic Commerce	93	93	0.90
16. Defense	342	182	0.77
17. Technology	196	131	0.82
18. Foreign Trade	27	27	0.11
19. International Affairs	7617	366	0.89
20. Government Operations	1247	294	0.67
21. Public Lands	10	10	0.70
23. Culture	124	100	0.99
26. Weather and Natural Disasters	201	133	1.00
27. Accidents, Fire Incidents, Disasters	50	50	0.92
28. Arts, Cult., Hist., Sci. and Entertainment	677	246	0.90
29. Sports News and Recreation	385	193	0.90
30. Deaths, Fatalities, Obituaries	41	41	0.85
31. Churches and Religion	194	130	0.98
33. Political Parties	661	244	0.99
34. Tabloid News, Celebrities	28	28	0.54
35. Crime	339	181	0.91
36. Political Corruption	8	8	0.63
<b>Total Result</b>	<b>19841</b>	<b>4630</b>	<b>0.84</b>

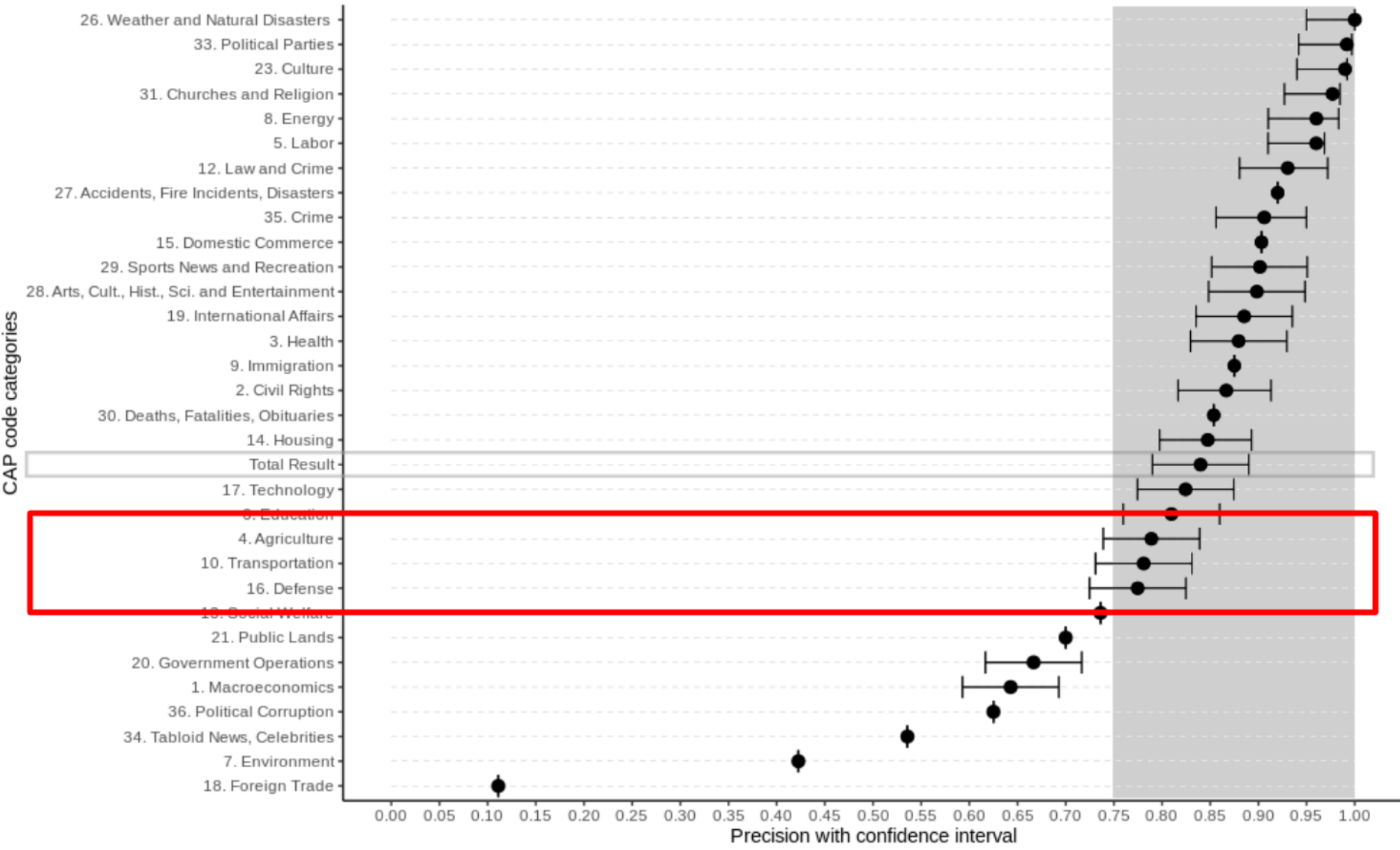
# Precision of MN corpus coding by CAP code category



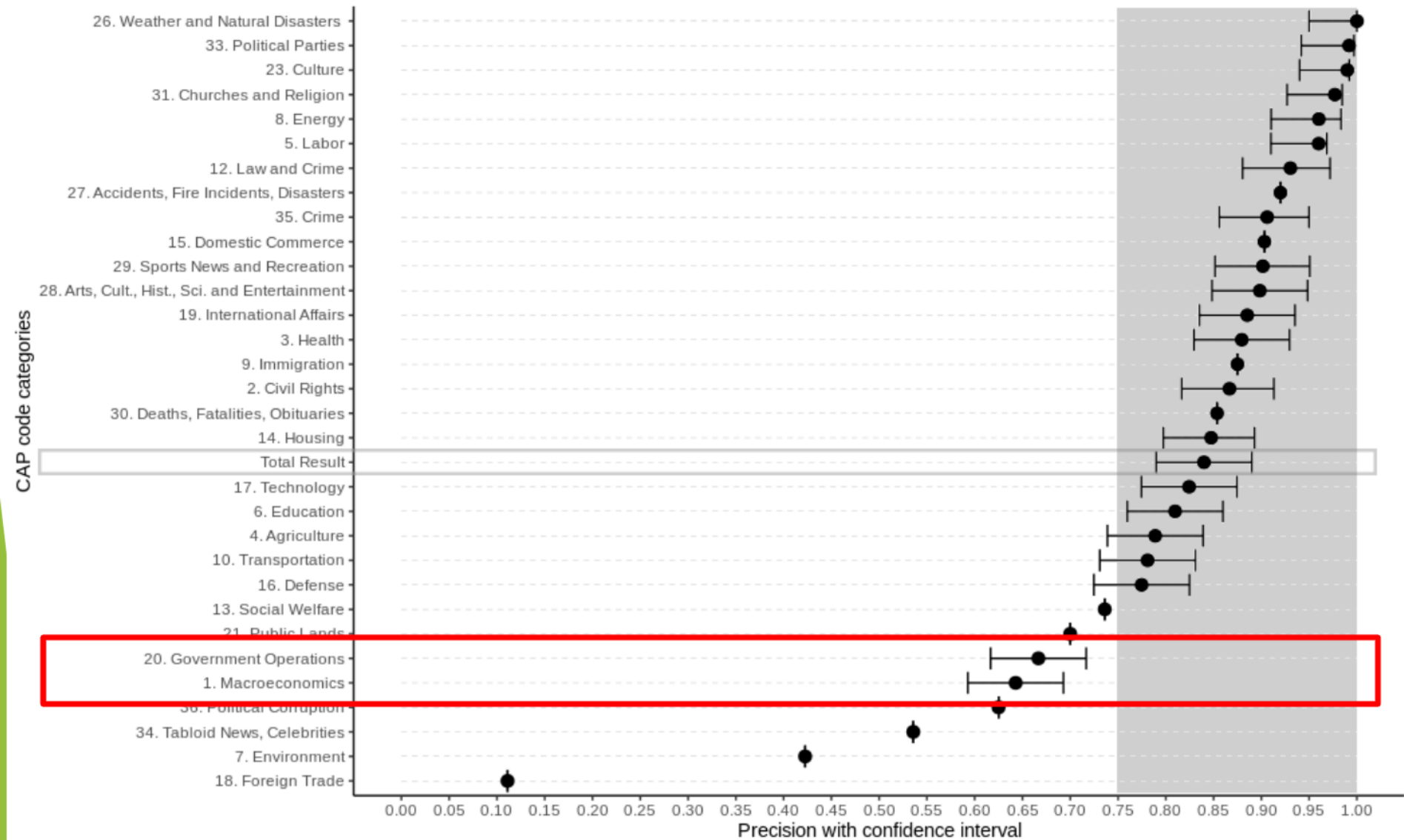
# Precision of MN corpus coding by CAP code category



# Precision of MN corpus coding by CAP code category

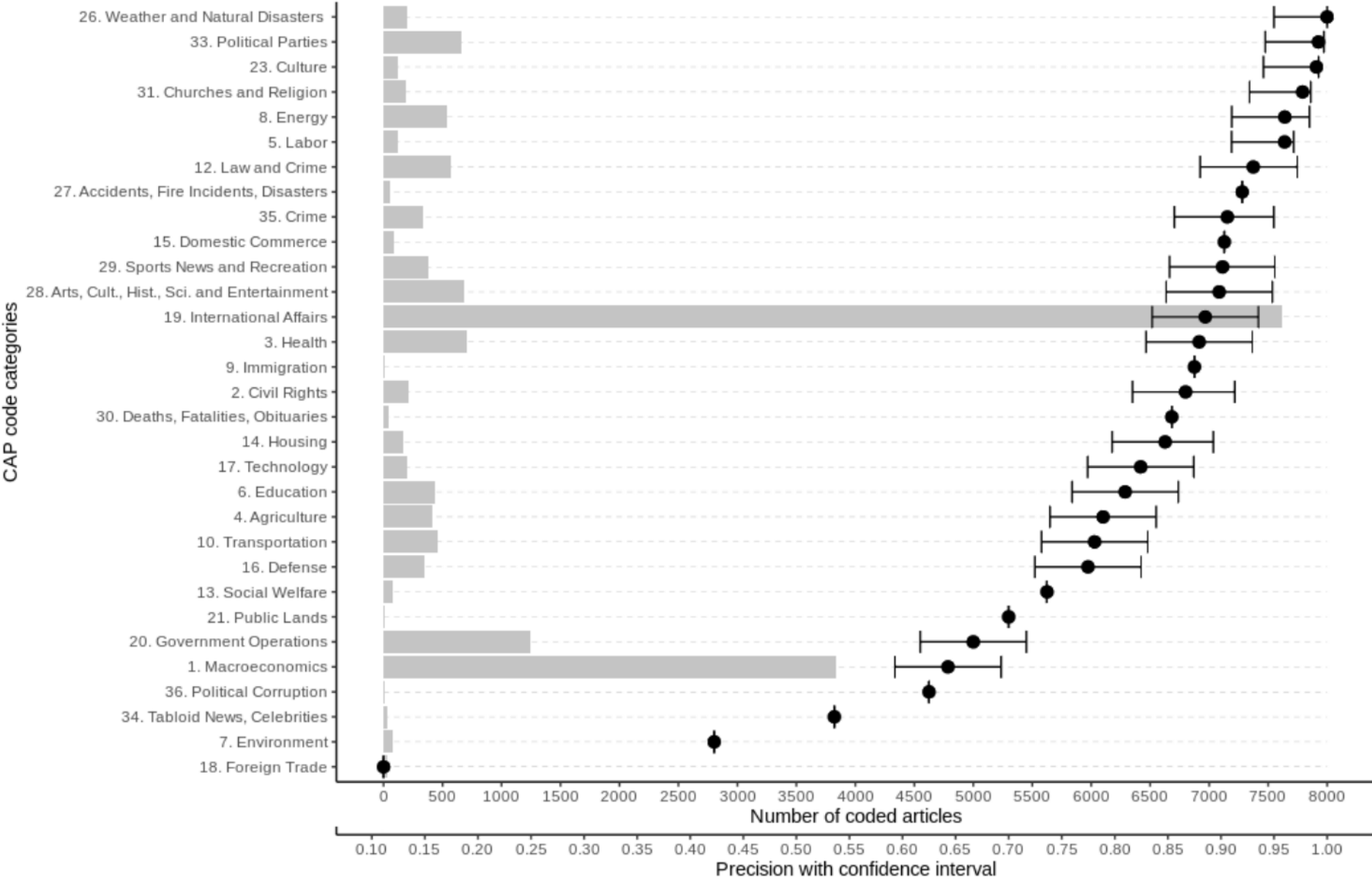


# Precision of MN corpus coding by CAP code category

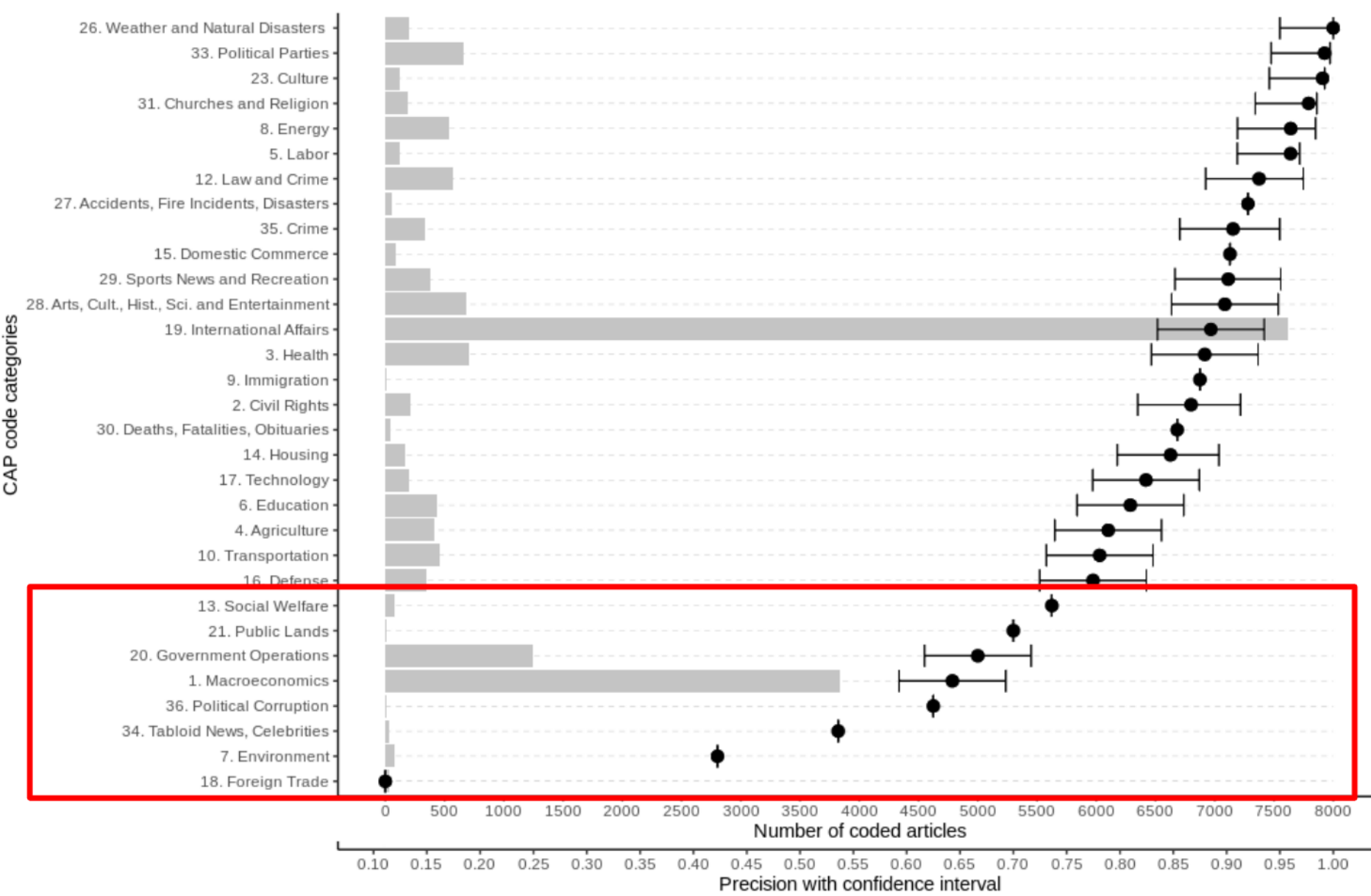




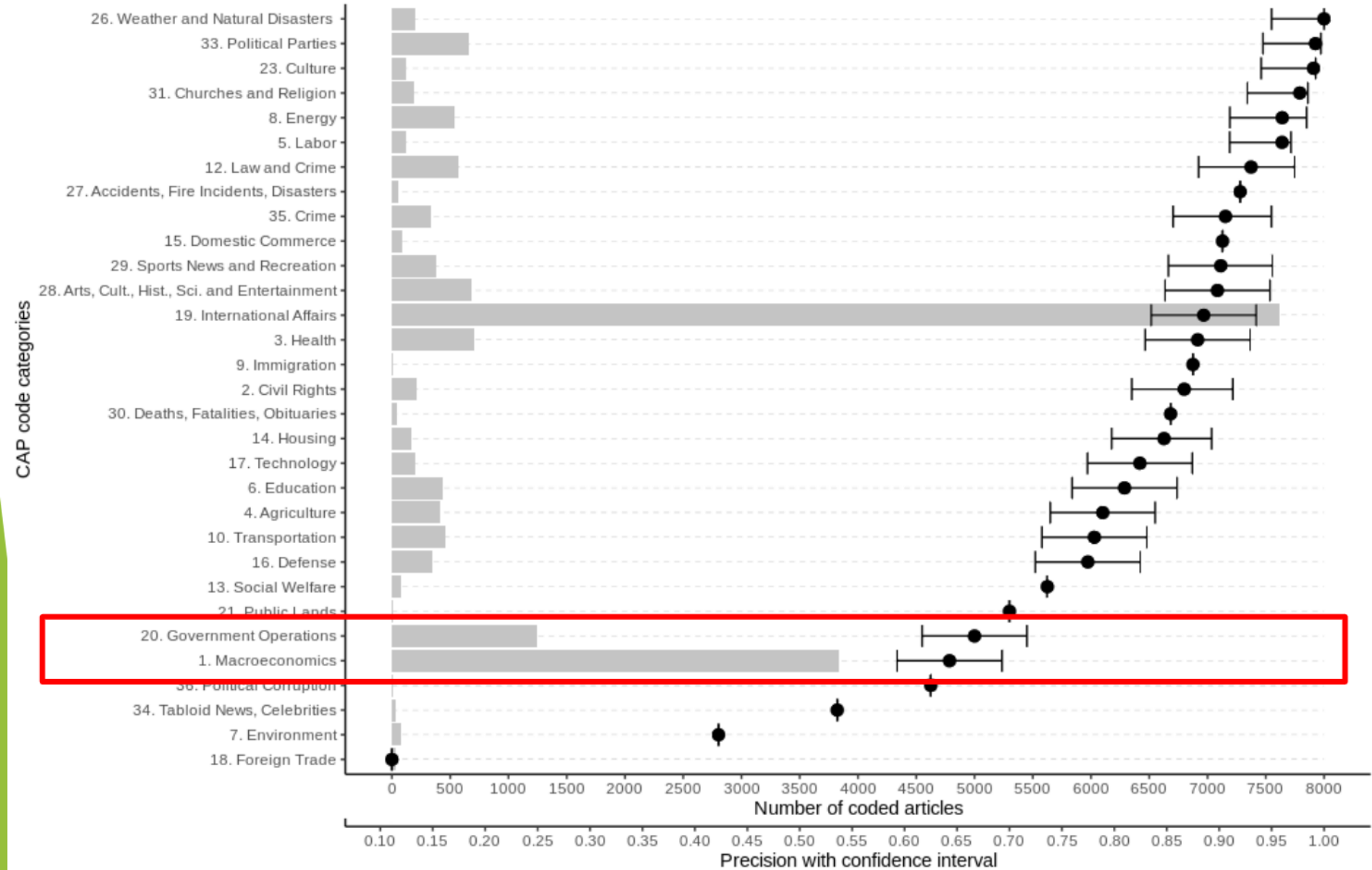
Precision and total number of coded articles of MN corpus by CAP code category



# Precision and total number of coded articles of MN corpus by CAP code category



# Precision and total number of coded articles of MN corpus by CAP code category



Major Topic	Coded Articles	Sample Size	Precision
1. Macroeconomics	3833	350	0.64
2. Civil Rights	207	135	0.87
3. Health	700	249	0.88
4. Agriculture	408	199	0.79
5. Labor	127	100	0.96
6. Education	436	205	0.81
7. Environment	71	71	0.42
8. Energy	542	226	0.96
9. Immigration	8	8	0.88
10. Transportation	459	210	0.78
12. Law and Crime	570	230	0.93
13. Social Welfare	72	72	0.74
14. Housing	168	118	0.85
15. Domestic Commerce	93	93	0.90
16. Defense	342	182	0.77
17. Technology	196	131	0.82
18. Foreign Trade	27	27	0.11
19. International Affairs	7617	366	0.89
20. Government Operations	1247	294	0.67
21. Public Lands	10	10	0.70
23. Culture	124	100	0.99
26. Weather and Natural Disasters	201	133	1.00
27. Accidents, Fire Incidents, Disasters	50	50	0.92
28. Arts, Cult., Hist., Sci. and Entertainment	677	246	0.90
29. Sports News and Recreation	385	193	0.90
30. Deaths, Fatalities, Obituaries	41	41	0.85
31. Churches and Religion	194	130	0.98
33. Political Parties	661	244	0.99
34. Tabloid News, Celebrities	28	28	0.54
35. Crime	339	181	0.91
36. Political Corruption	8	8	0.63
Total Result	19841	4630	0.84



# Main contributions of HBS and the present study



- Enhance ML precision and recall by both **human input** (validation) and **workflow design** (one-vs-all classification, ensemble voting)
- Start working from a **limited training set**
- Able to **maximize ROI on human coding**
- **Move between (intra-domain) corpora**
- Take advantage of **cloud infrastructure** and **parallel processing** with Apache Spark

# Further work

- Implement a finishing step using **regular expressions to correct systematic errors**
  - “design” in *Environment*
  - “icerink” in *Public Lands*
- Testing the HBS approach on **further languages**
- Generalizing the method to **other domains** beyond media



ELKH Cloud

**Thank you for your attention!**

sebok.miklos@tk.mta.hu  
kacsuk.zoltan@tk.mta.hu

poltext.tk.mta.hu  
cap.tk.mta.hu

 tkpti

The logo for tkpti, consisting of a stylized icon of three horizontal bars in blue, green, and orange, followed by the text 'tkpti' in a sans-serif font.