1 Descriere algoritm inteligent

1.1 Algoritm ales

Am ales **GPT-Neo Romanian 780M**, un model de tip autoregressive Transformer pre-antrenat pe text în limba română. Acesta poate genera text respectând specificul limbii române.

1.2 Motivația alegerii

- **Limbă nativă:** Modelul este special instruit pe limba română, deci output-ul va fi mult mai natural și corect din punct de vedere gramatical.
- **Dimensiune potrivită pentru CPU/Colab Free:** 780M parametri îl fac suficient de mic pentru a rula pe CPU fără a necesita resurse GPU extinse.
- **Transfer Learning / Few-Shot:** Setul de date este mic (∼100–200 exemple). În loc să antrenăm de la zero, folosim *prompt engineering* (few-shot learning) sau fine-tuning cu LoRA/QLoRA.
- **Flexibilitate:** Poate genera cereri de tip apel, recurs sau chemare în judecată, adaptând structura textului conform instrucțiunilor.

1.3 Mod de funcționare

- Modelul primește un prompt care conține toate detaliile cererii: tipul cererii, instanța, părțile implicate, obiectul cererii, temei legal și probele.
- Folosind mecanismul *self-attention* specific Transformer-elor, modelul generează un text coerent și formal, conform standardelor juridice românești.
- Pentru setul mic de date, nu se antrenează complet modelul, ci se folosește *few-shot prompting* pentru adaptarea la stilul juridic dorit.

1.4 Librării folosite

- `transformers` (Hugging Face) – pentru încărcarea modelului GPT-Neo Romanian 780M
- `torch` (PyTorch) – backend pentru calcul tensorial și generare text
- `random` – generarea prompturilor variate pentru experiment

1.5 Avantaje și limitări

- **Avantaje:** generare text în română, nu necesită GPU pentru seturi mici, permite testarea rapidă a diferitelor tipuri de cereri judiciare

- **Limitări:** precizia depinde foarte mult de claritatea prompt-ului și de lungimea cererii, modelele mai mici pot produce outline-uri sau propoziții incomplete pentru texte lungi (¿200 cuvinte), textele generate nu sunt întotdeauna fluente

2 Metodologie experimentală și rezultate obținute

2.1 Hiperparametri de generare

- `max_length` = 250
- `top_k` = 50
- `top_p` = 0.95
- `temperature` = 0.7

2.2 Setul de date

Am folosit un set mic (∼100 exemple) de cereri judiciare sintetice, generate pe baza unor șabloane standard în format JSON. Datele au fost anonimizate, respectând bune practici etice și GDPR.

2.3 Procedura generală

1. Se selectează aleator:
   - **parte_reclamant** și **parte_parat** din lista `parti_implicate`
   - **scop** din `context_scop`
   - **actiune** din `tip_actiune`
2. Se construiește prompt-ul care descrie elementele cererii.
3. Modelul generează text complet folosind metoda `.generate()` cu sampling (`top_k`, `top_p`) pentru diversitate.
4. Textul generat include secțiuni pentru data, semnătura și ștampilă (în practică, acestea nu au fost întotdeauna respectate).

2.4 Rezultate

- **BLEU score:** 0.17

3 Concluzii

- Răspunsurile generate LLM nu sunt coerente și nu urmează întotdeauna instrucțiunile primite în prompt.
- LLM-ul încearcă să completeze prompt-ul, nu să răspundă direct la cerere.
- In urma utilizarii unui alt LLM mai mic: LLM-ul mai mare oferă răspunsuri mai structurate.
- Fine-tuning-ul nu a făcut o diferență majoră în ceea ce privește calitatea răspunsului.