TEXT MINING 4

Objavovanie znalostí v textoch

Peter Bednár

Jazykové modely a vyhľadávanie informácii

Aké otázky vie LLM zodpovedať?

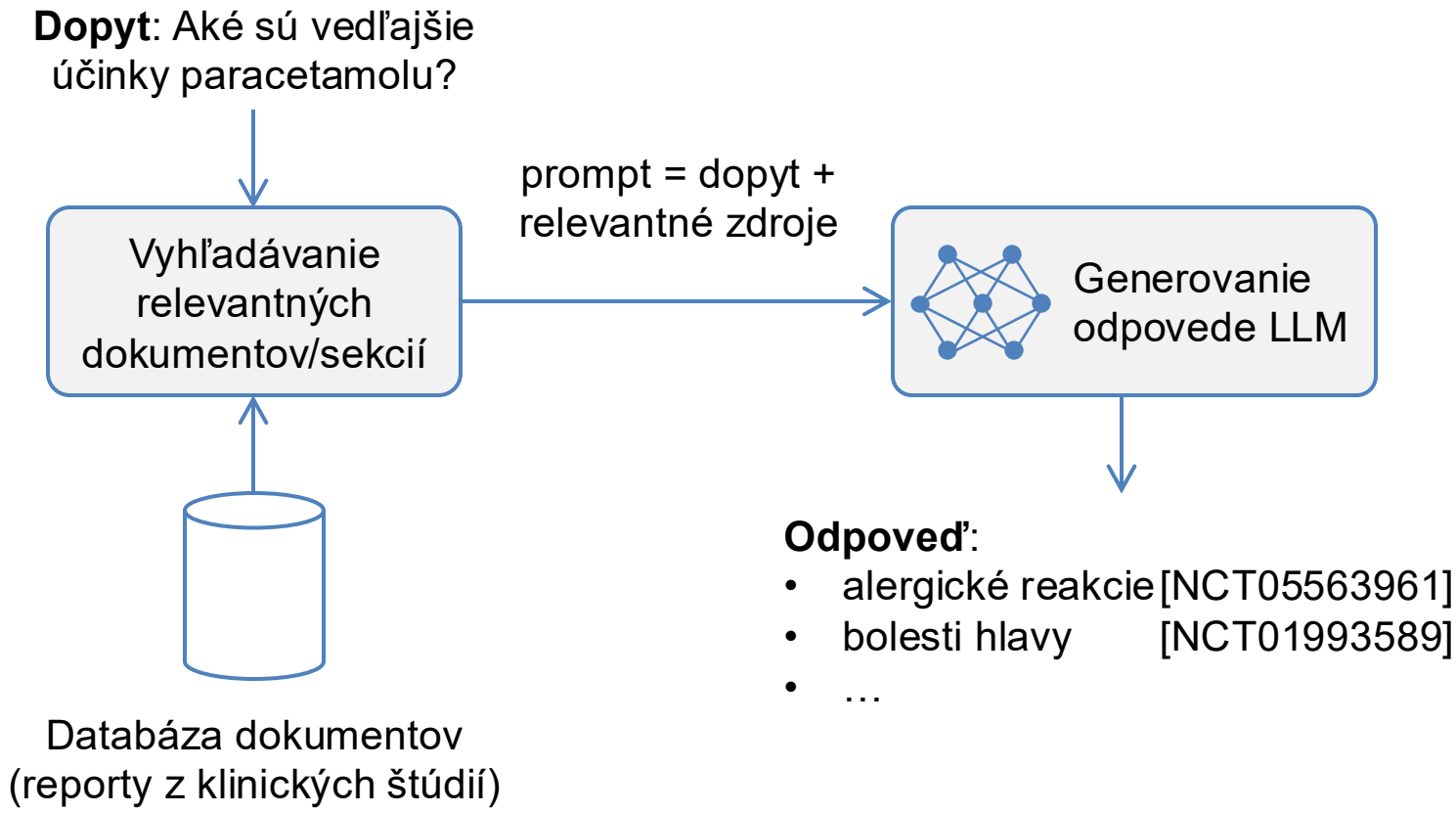
- Nedostatky LLM pri zodpovedaní otázok
  - LLM je naučený na statickej dátovej množiny a nemá prístup k aktuálnym informáciám
  - Halucinovanie (zvlášť pre informácie chýbajúce v trénovacích dátach)
  - Nemá prístup k proprietárnym dátam
  - Nevie spoľahlivo poskytnúť zdroj dát z ktorých odvodil odpoveď

RAG - Retrieval-Augmented Generation (1)

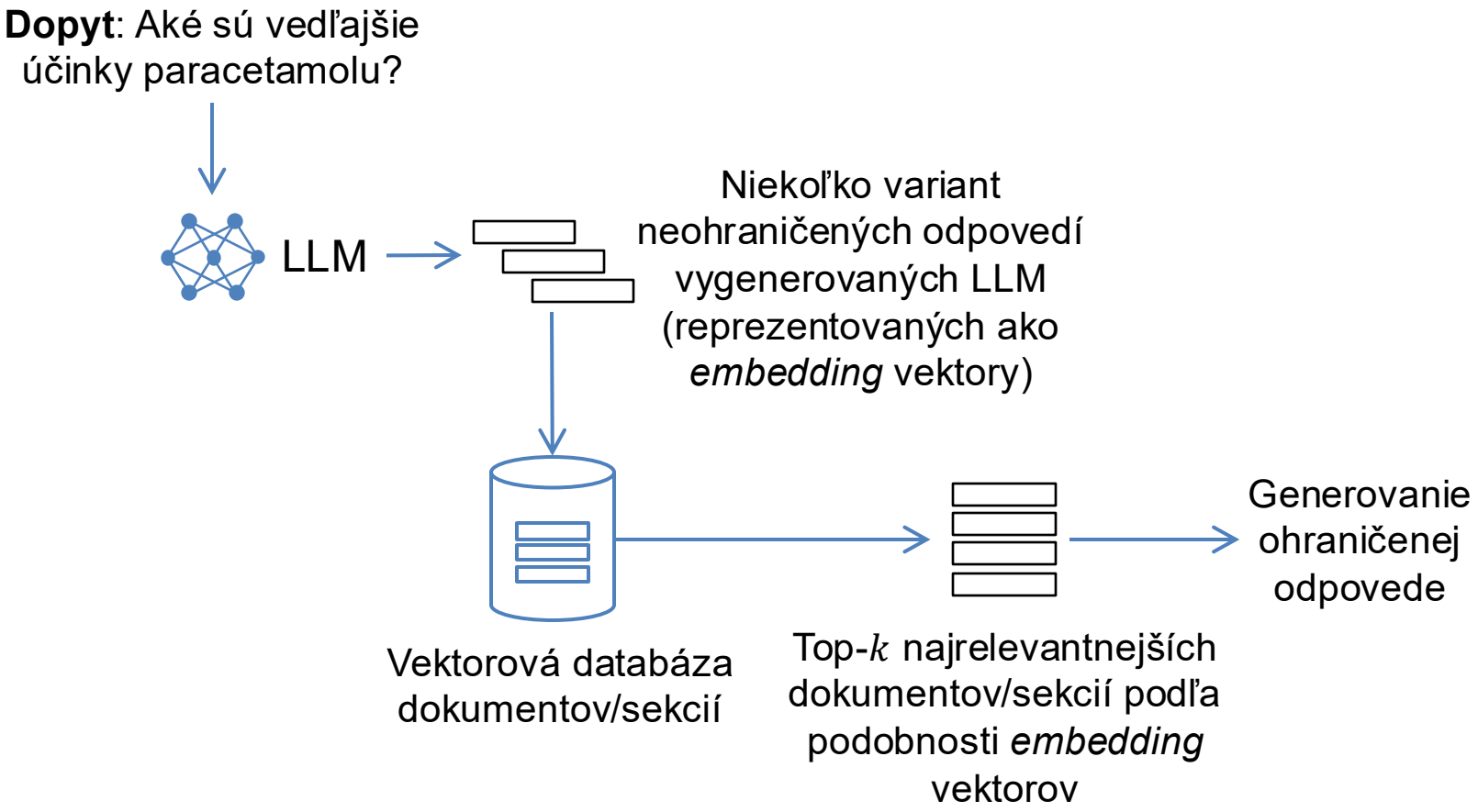
- Prepojíme databázu dokumentov/znalostnú bázu s LLM
- Vo vstupnom prompte ohraničíme generovanie odpovede v LLM iba na informácie z relevantných dokumentov vyhľadaných v báze znalostí

- Stačí aktualizovať databázu dokumentov pre pridanie nových informácií (nie je potrebné doučiť LLM)
- Databáza môže obsahovať aj privátne dokumenty
- V odpovedi môžeme zahrnúť aj odkazy na zdrojové dokumenty

RAG - Retrieval-Augmented Generation (2)



RAG - Retrieval-Augmented Generation (3)



RAG - Retrieval-Augmented Generation (4)

- Vektorové databázy je problematické škálovať
- Pre veľmi rozsiahle dáta sa vyžíva TF-IDF vyhľadávanie podľa termov na pred-filtrovanie dokumentov/sekcií, ktoré sa následne preusporiadanú podľa podobnosti *embedding* vektorov
- TF-IDF dopyt (kľúčové slová z otázky + súvisiace pojmy/synonymá) vygenerujeme pomocou LLM

Úlohy spracovania prirodzeného jazyka

Prehľad úloh spracovania prirodzeného jazyka (1)

- Spracovanie jazyka
  - Rozpoznávanie reči
  - Morfosyntaktická analýza
  - Automatický preklad
  - Sumarizácia textov
  - Generovanie kreatívneho textu
  - *Chatboty* / dialógové systémy
  - Odpovedanie na otázky

Prehľad úloh spracovania prirodzeného jazyka (2)

- Objavovanie znalostí z textov
  - Klasifikácia textov
  - Analýza sentimentu
  - Detekcia toxického obsahu
  - Zhlukovanie textov
  - Modelovanie tém
  - Extrahovanie informácií

Spôsoby vyhodnotenia metód

Výstup s preddefinovanými označeniami

- Označenie môže byť na úrovni celého textu, viet, alebo tokenov
- Základné metriky odvodené z kontingenčnej tabuľky pre klasifikačné úlohy
  - Presnosť, Návratnosť, F_β-skóre
  - Krivka presnosť-návratnosť, AUC (*Area Under the Curve*)
  - Mikro- a Makro spriemerovanie pre viacnásobnú klasifikáciu do viacerých tried

Metriky založené na prekrytí (1)

- Porovnáva sa vygenerovaná postupnosť tokenov s referenčnými reťazcami anotovanými expertom (môžeme mať viac správnych odpovedí, napr. pri preklade)

- Exaktné porovnávanie
  - Presne sa zhodujú tokeny a ich poradie
- Metriky založené na editačných vzdialenostiach
  - Meria sa počet tokenov/znakov, ktoré treba pridať, odstrániť alebo zameniť aby sa vygenerovaná postupnosť $\hat{y}$ prepísala na referenčnú postupnosť y
  - Napr. *Levenshteinova* vzdialenosť

Metriky založené na prekrytí (2)

- BLUE (*bilingual evaluation understudy*)
  - Vypočíta sa presnosť P na úrovni n-gramov, tzn. počet n-gramov v $\hat{y}$, ktoré sa vyskytli aj v y / celkový počet n-gramov v $\hat{y}$
  - Výsledné skóre je vážený geometrický priemer pre 1,2,3,4-gramy × penalizácia ak je $\hat{y}$ kratší než y

$$BP = \begin{cases} 1, & ak\ |\hat{y}| \geq |y| \\ e^{\left(1-\frac{|\hat{y}|}{|y|}\right)}, & ak\ |\hat{y}| < |y| \end{cases}$$

Metriky založené na prekrytí (3)

- ROUGE (*Recall-Oriented Understudy for Gisting Evaluation*)
  - Vypočíta sa presnosť P, návratnosť R a F_1 skóre pre:
  - ROUGE-1 – unigramy
  - ROUGE-2 – bi-gramy
  - ROUGE-L – najdlhšiu spoločnú sekvenciu (nemusí byť súvislá)

Metriky založené na prekrytí (4)

- METEOR (*Metric for Evaluation of Translation with Explicit ORdering*)
  - Vypočíta sa presnosť P, návratnosť R a F_3 skóre (návratnosť je 3x dôležitejšia než presnosť) $F_3 = \frac{10PR}{9R+P}$
  - Zohľadňuje sa nielen presná zhoda medzi tokenmi, ale aj stemming a synonymá z WordNetu

BLUE – príklad (1)

- Preklad vety „*ho cinquantatré anni*“ do angličtiny
- Vygenerovaná odpoveď: „*I have fifty three years*“
- Referenčné odpovede: „*I am fifty three years old*“, „*I am fifty three*“

P unigramy: I have fifty three years = $4/5$

P bigramy: I have have fifty fifty three three years = $2/4$

Geometrický priemer: $\sqrt[2]{4/5 \times 2/4}$

Výsledné skóre bez penalizácie, pretože vygenerovaná odpoveď je dlhšia než referenčná

BLUE – príklad (2)

- Pre opakujúce sa slová sa započíta iba počet výskytov v y

  Pre unigramy namiesto I have fifty fifty fifty fifty three years = $7/8$
  sa započíta fifty iba raz = $1/8$

- Poradie sa zachováva iba čiastočne

  Bigramy: fifty three three years years I I have = $2/4$?

ROUGE – príklad (1)

- Správa na soc. médiách: „*Just finished reading Pride and Prejudice and I'm once again amazed...* "
- Vygenerované zhrnutie: „*I really loved to read Jane Austen*"
- Referenčné zhrnutie: „*I loved reading Jane Austen*"

- ROUGE-1

  I really loved to read Jane Austen

  I loved reading Jane Austen

$$P = \frac{4}{7},\ R = \frac{4}{5},\ F_1 = \frac{2PR}{P+R} = 0{,}666$$

ROUGE – príklad (2)

- ROUGE-2

  I really really loved loved to to read read Jane Jane Austen

  I loved loved reading reading Jane Jane Austen

$$P = \frac{1}{6},\ R = \frac{1}{4},\ F_1 = \frac{2PR}{P+R} = 0{,}2$$

- ROUGE-L

  I really loved to read Jane Austen

  I loved reading Jane Austen

$$P = \frac{4}{7},\ R = \frac{4}{5},\ F_1 = \frac{2PR}{P+R} = 0{,}666$$

Metriky založené na jazykových modeloch (1)

- BERT Score
  - Kosínusová podobnosť medzi *embeddingami* vypočítanými pred-učeným jazykovým modelom
  - Odstraňuje niektoré nevýhody metrík založených na prekrytí (rôzne parafrázy s tým istým významom sú namapované na podobné vektory)

Metriky založené na jazykových modeloch (2)

- Model-as-a-Judge
  - Na výpočet ohodnotenia kvality výstupu je použitý hodnotiaci LLM
  - Hodnotiaci LLM má na vstupe prompt, ktorý je zložený z:
    1. Kontextu
    2. Hodnoteného výstupu/výstupov
    3. Inštrukcií aké kritérium sa má hodnotiť
    4. Inštrukcií ako má vyzerať výsledný formát hodnotenia

Príklad promptu Model-as-a-Judge

```
You are an impartial judge evaluating two answers for user's question.

Your goal is to determine which response better answers the user's
question according to the following criteria:
    - Accuracy: Is the information correct?
    - Completeness: Does it fully answer the question?
    - Clarity: Is the explanation clear and easy to understand?
    - Relevance: Does it stay focused on the question?

Question: [text]
Answer A: [text]
Answer B: [text]

Evaluate both responses and decide overall score X and Y in range 1-10
for each response. Explain your reasoning step by step, and then give
your final verdict in this format:

Verdict: Response [A/B] is better with score of X vs Y.
```

Metriky založené na jazykových modeloch (3)

- Hodnotiaci model by mal byť výkonnejší, použitie viacerých rozmanitých LLM
- Slepé testovanie – hodnotiaci model nemá mať priame informácie, ako bol vygenerovaný hodnotený výstup (expertom/ktorým modelom)
- Hybridné testovanie expertom
  - Aspoň pre časť hodnotení

Výhody Model-as-a-Judge

- Škálovateľnosť
- Testovanie rôznych kritérií (faktická presnosť, úplnosť odpovede, zrozumiteľnosť, nezaujatosť)
- Kvantitatívne aj kvalitatívne hodnotenie + vysvetlenie hodnotenia
- Je možné ho použiť aj na automatické zamedzenie generovania nevhodného obsahu (tzv. *blacklisting*)

Nevýhody Model-as-a-Judge

- Bias hodnotiaceho modelu
- Obmedzená transparentnosť
  - Vysvetlenie je vygenerované a nemusí zodpovedať kauzálnym vzťahom pri hodnotení
- Problémy s konzistentnosťou a stabilitou
  - Odpovede LLM môžu byť nedeterministické, tzn. pre ten istý vstup môžeme dostať rôzne hodnotenia
- Chýba overená pravdivá hodnota (*ground true*)
- Etické problémy

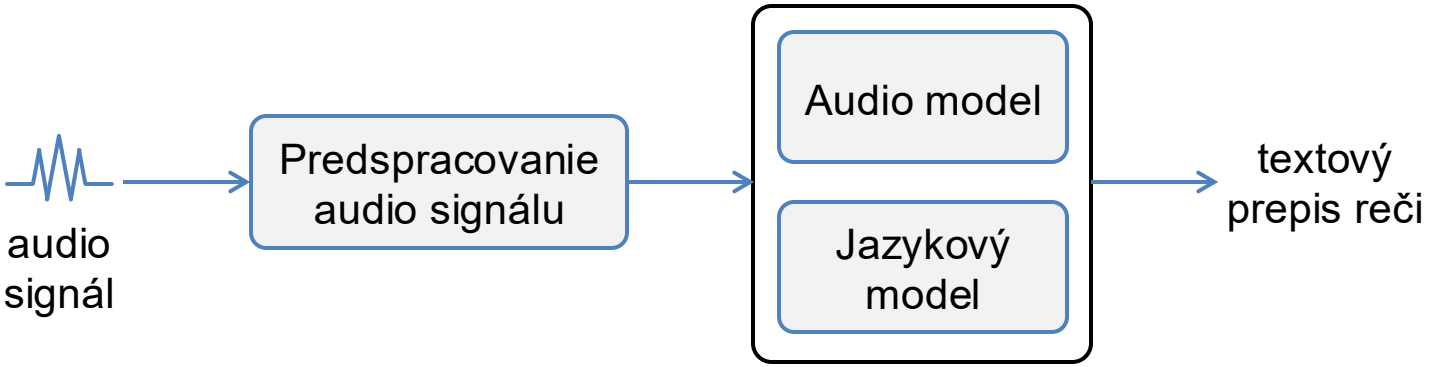
Rozpoznávanie reči a reprezentácia zvuku

Rozpoznávanie hovorenej reči

- Zložitosť úlohy na úrovni audio signálu:
  - Rôzne hlasy, rýchlosť hovorenia, artikulácia, prízvuk, rušné prostredie
- Ľudský mozog využíva pri rozpoznávaní hlások a slov zo zvuku ich jazykové vlastnosti (syntax a význam)

Hlboké učenie pre rozpoznávanie reči

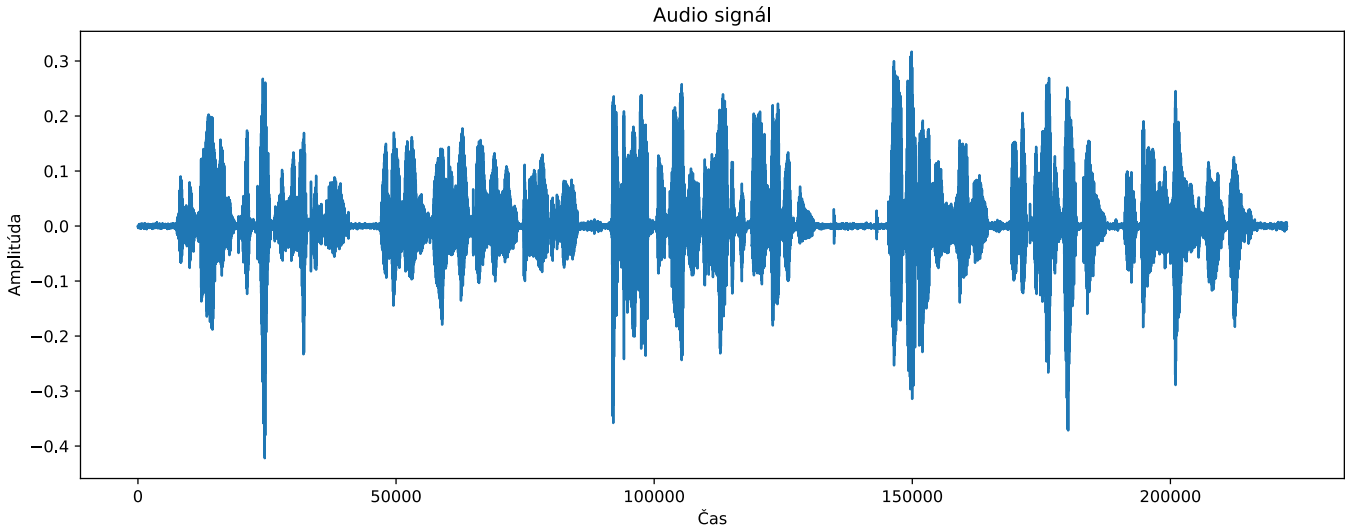
- Audio signál je reprezentovaný ako časové sekvencia – mapovanie sekvencií s rozdielnou dĺžkou – kodér/dekodér architektúra



Predspracovanie audio signálu (1)

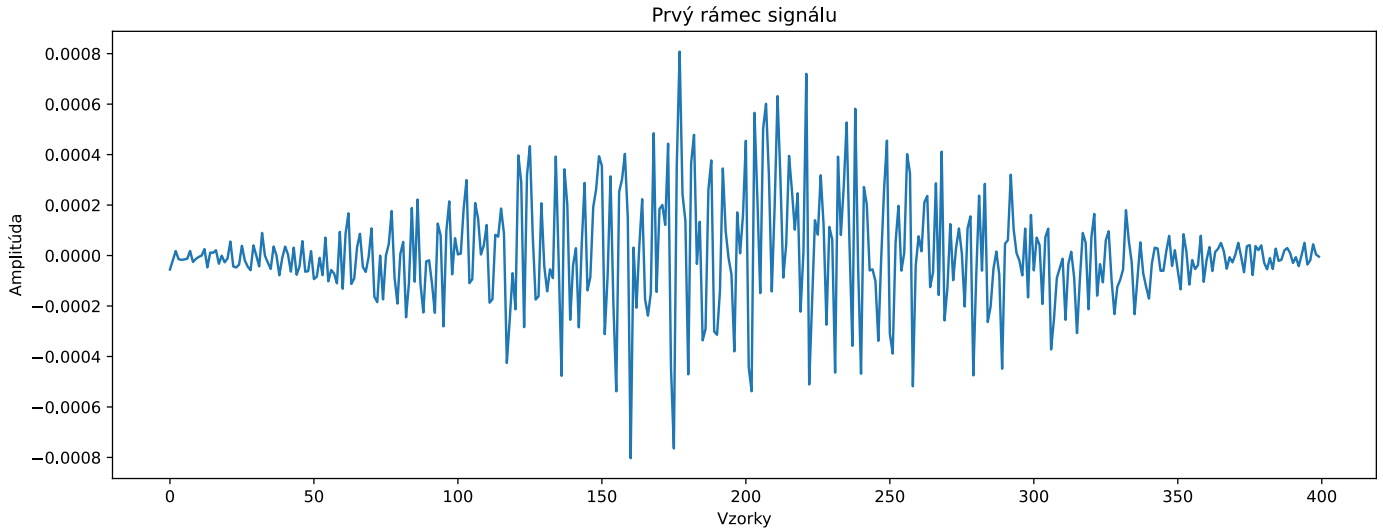
- Zvuk sa prenáša akustickými vlnami a je zaznamenaný ako časová sekvencia, štandardná vzorkovacia frekvencia pre rozpoznávanie reči je 16 kHz, pre kvalitné audio nahrávanie 44,1 kHz

- Príklad: 222 562 vzoriek, vzorkovacia frekvencia 16 kHz, 13,9s



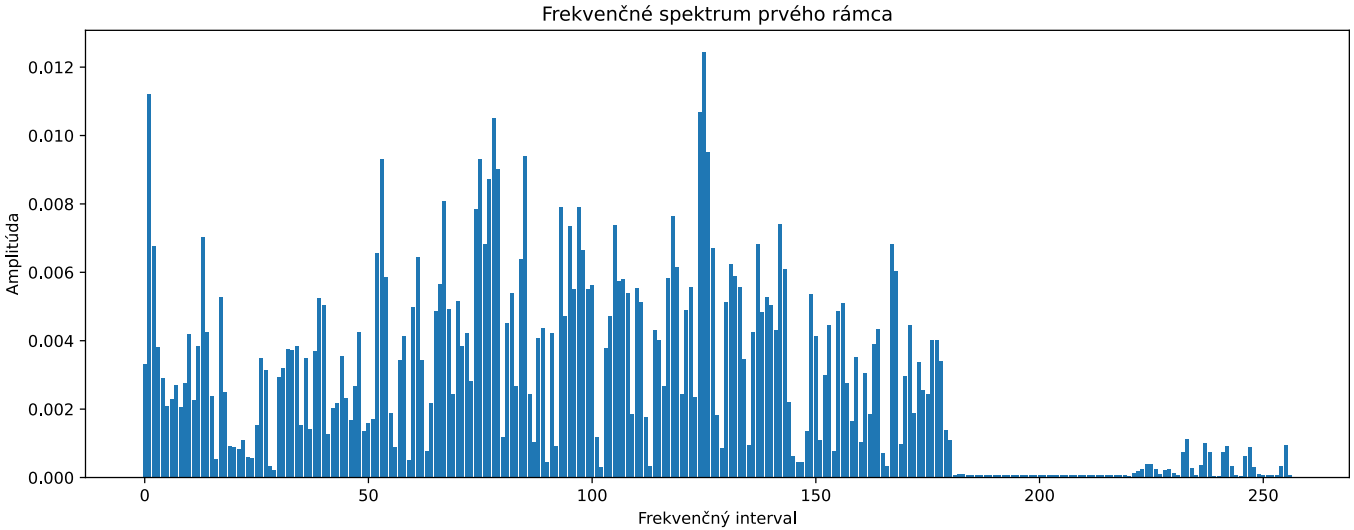
Predspracovanie audio signálu (2)

- Postupnosť sa rozdelí na krátke prekrývajúce sa rámce najčastejšie 20-40ms
- Na hodnoty rámca sa aplikuje Hammingov filter, ktorý vyhladí okrajové hodnoty
- Príklad: prvý rámec 25ms / 400 vzoriek, posun 10ms



Predspracovanie audio signálu (3)

- Signál sa prevedie do frekvenčnej domény rýchlou Fourierovou transformáciou (FFT - *Fast Fourier Transformation*)

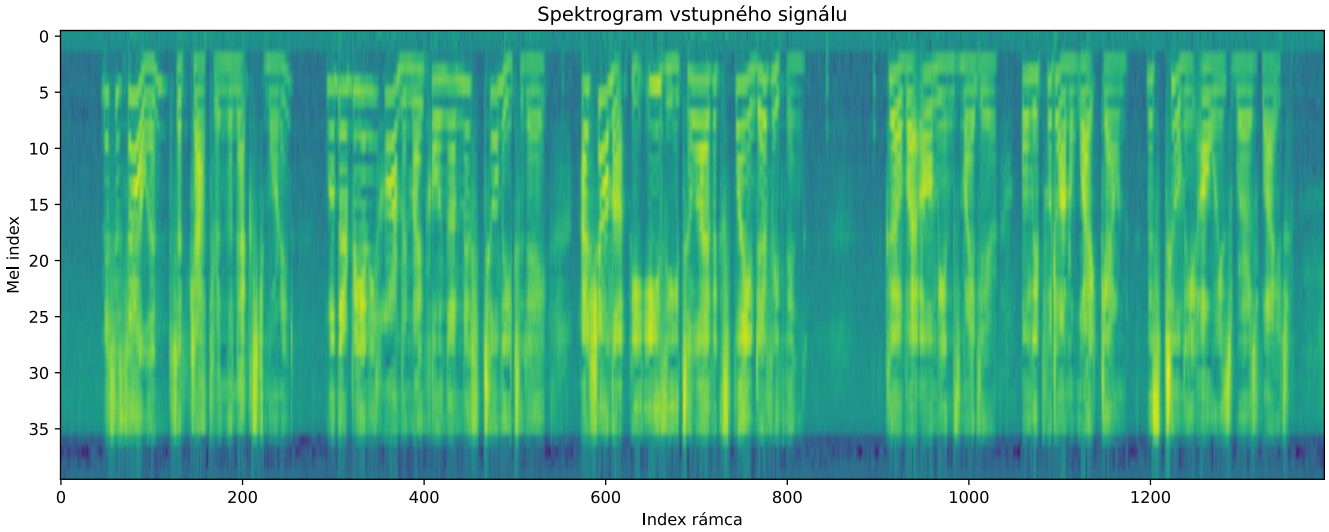


Predspracovanie audio signálu (4)

- Frekvencie spektra sa premapujú na mel stupnicu a hodnoty amplitúdy sa logaritmicky transformujú
  - Mel stupnica logaritmicky rozdeľuje intervaly frekvencií a priradzuje im index
  - Človek nerozlišuje frekvencie rovnako (rozdiel medzi 100 – 200 Hz vnímame výraznejšie než medzi 1100-1200 Hz)
  - Hlasitosť (ktorá je daná veľkosťou amplitúdy) tiež vnímame logaritmicky

Predspracovanie audio signálu (5)

- Po prepočítaní každého rámca dostaneme spektogram vstupného signálu
- Príklad: 40 mel indexov × počet rámcov

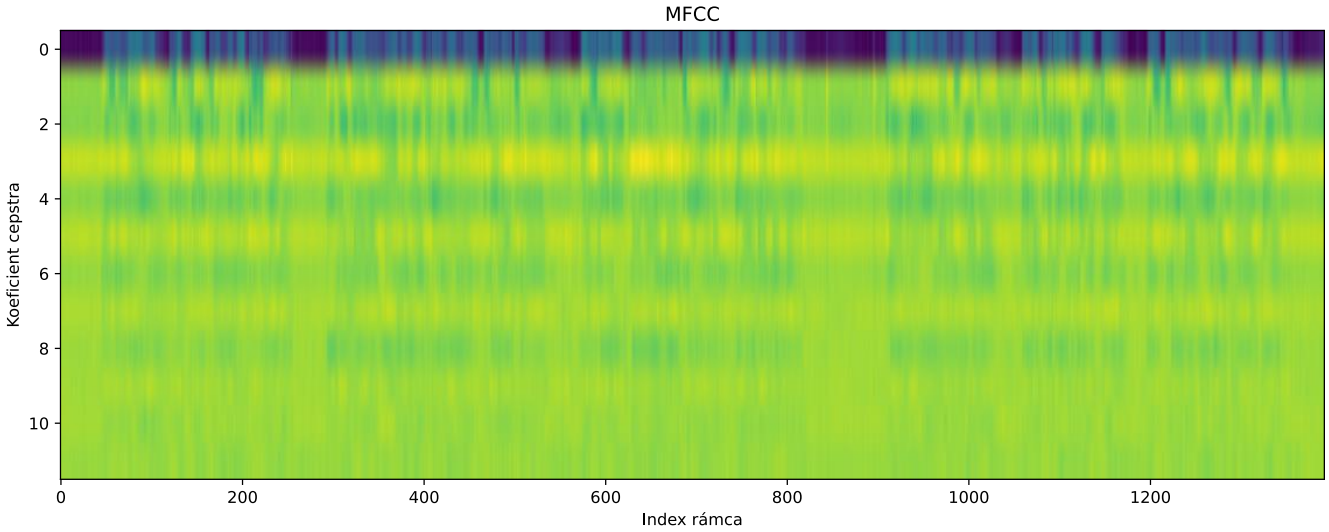


Predspracovanie audio signálu (6)

- Pri rozpoznávaní reči nás zaujímajú najmä prechody medzi frekvenciami
  - Pomocou diskrétnej kosínusovej transformácie (DCT – *Discrete Cosine Transformation*) získame kepstrum – spektrum spektra

Predspracovanie audio signálu (7)

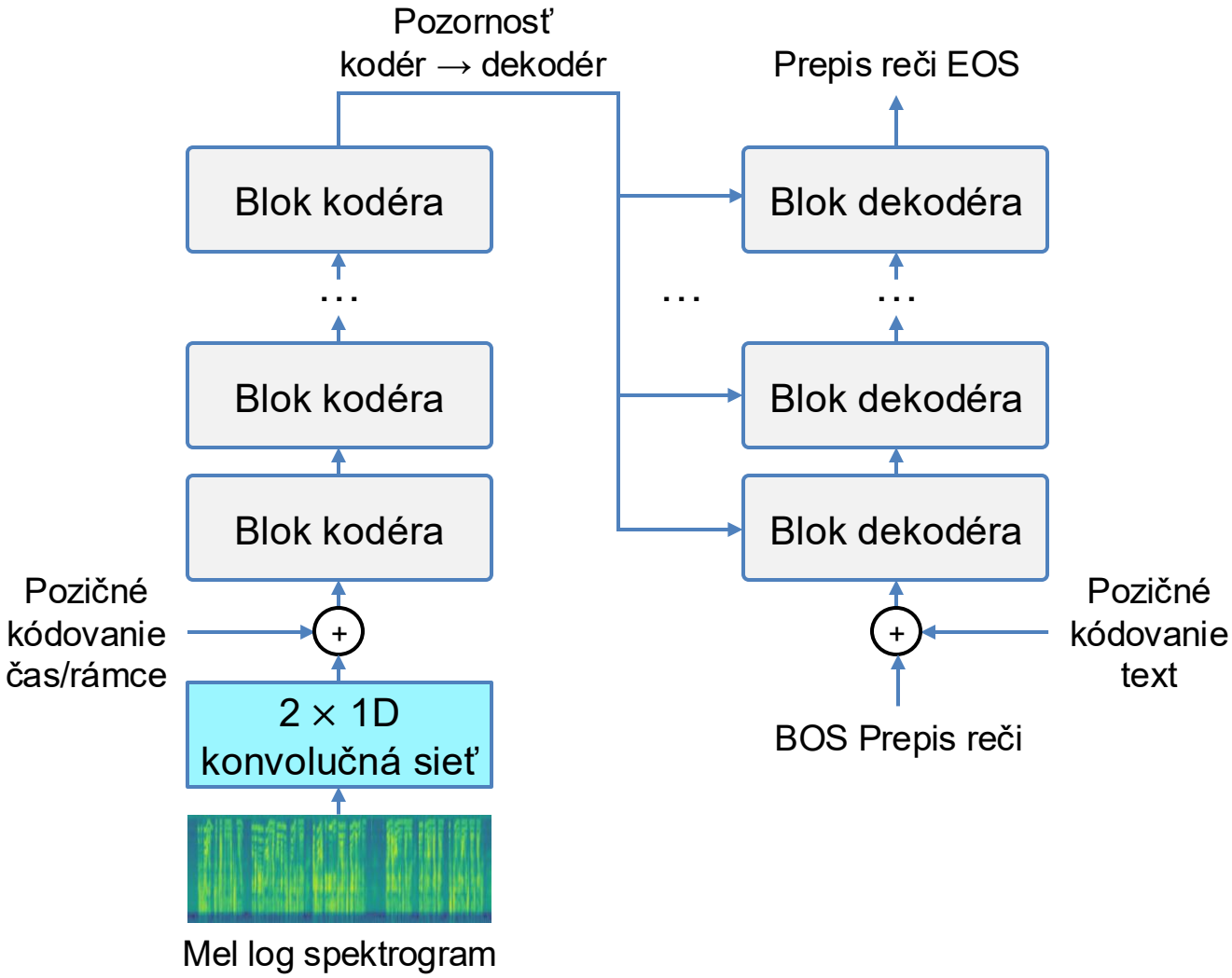
- Výsledná reprezentácia – MFCC – *Mel-Frequency Cepstral Coefficients*
- Príklad: 12 MFC koeficientov × počet rámcov



Metódy rozpoznávania reči

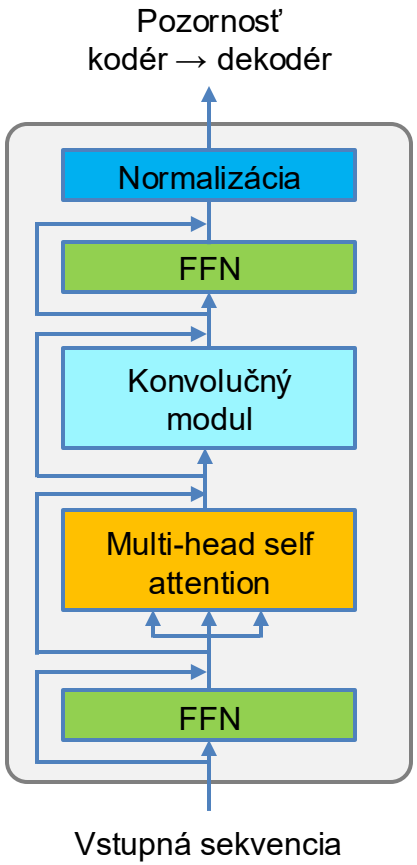
- Audio model
  - Rozpoznávanie hlások
  - Lokálne príznaky extrahované konvolučnou sieťou
- Jazykový model
  - Spájanie hlások do slov a slov do viet/vyjadrení
  - Mapovanie sekvencia-sekvencia s rôznou dĺžkou
  - Vzdialené závislosti
    - Transformer – kodér/dekodér architektúra
- Hybridný model
  - Conformer

Transformer pre rozpoznávanie reči



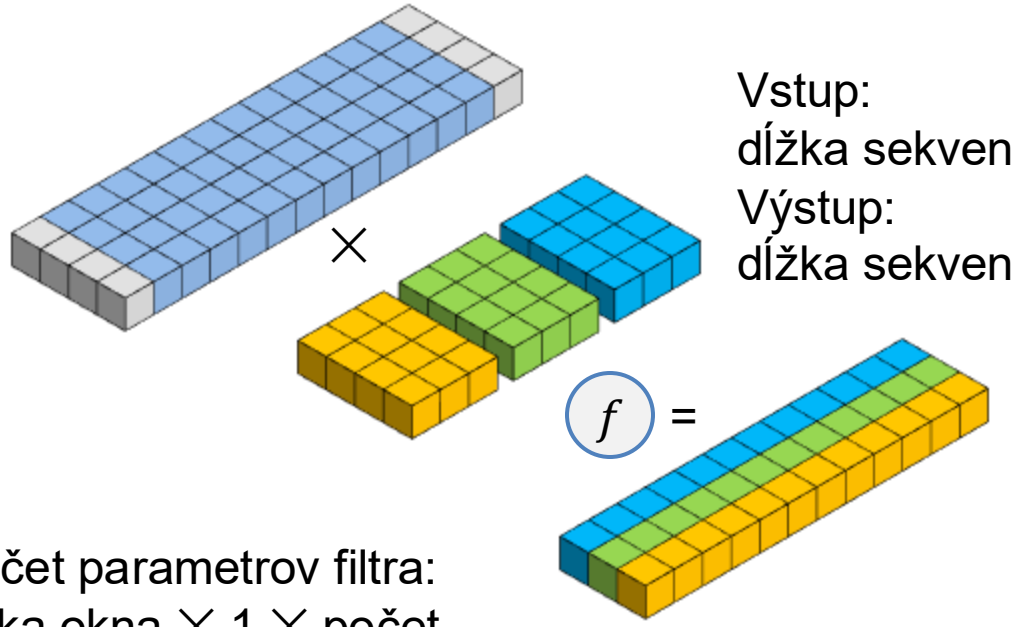
Conformer – konvolučná vrstva + pozornostná vrstva

Blok kodéra
$n \times$



- Lokálne závislosti
  - Rozpoznávanie príznakov zo zvuku
  - Konvolučný modul zložený z *depth-wise* a *point-wise* konvolúcie
- Vzdialené závislosti
  - Fonetické zloženie slova, syntax, sémantika
  - Samo-pozornostá vrstva

Depth-wise a *point-wise* 1D konvolúcia (1)

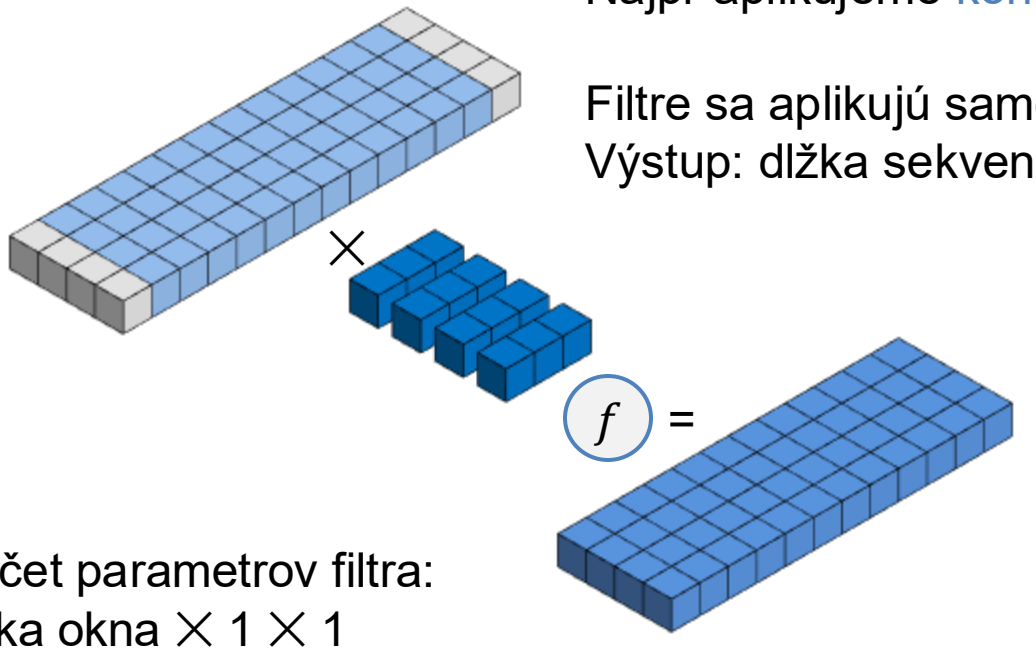


Klasická 1D konvolúcia

Vstup:
dĺžka sekvencie + okraj × 1 × počet kanálov
Výstup:
dĺžka sekvencie × 1 × počet filtrov

Počet parametrov filtra:
šírka okna × 1 × počet kanálov (voliteľne + 1 pre bias)

Depth-wise a *point-wise* 1D konvolúcia (2)



Najpr aplikujeme konvolúciu do hĺbky (*depth-wise*)

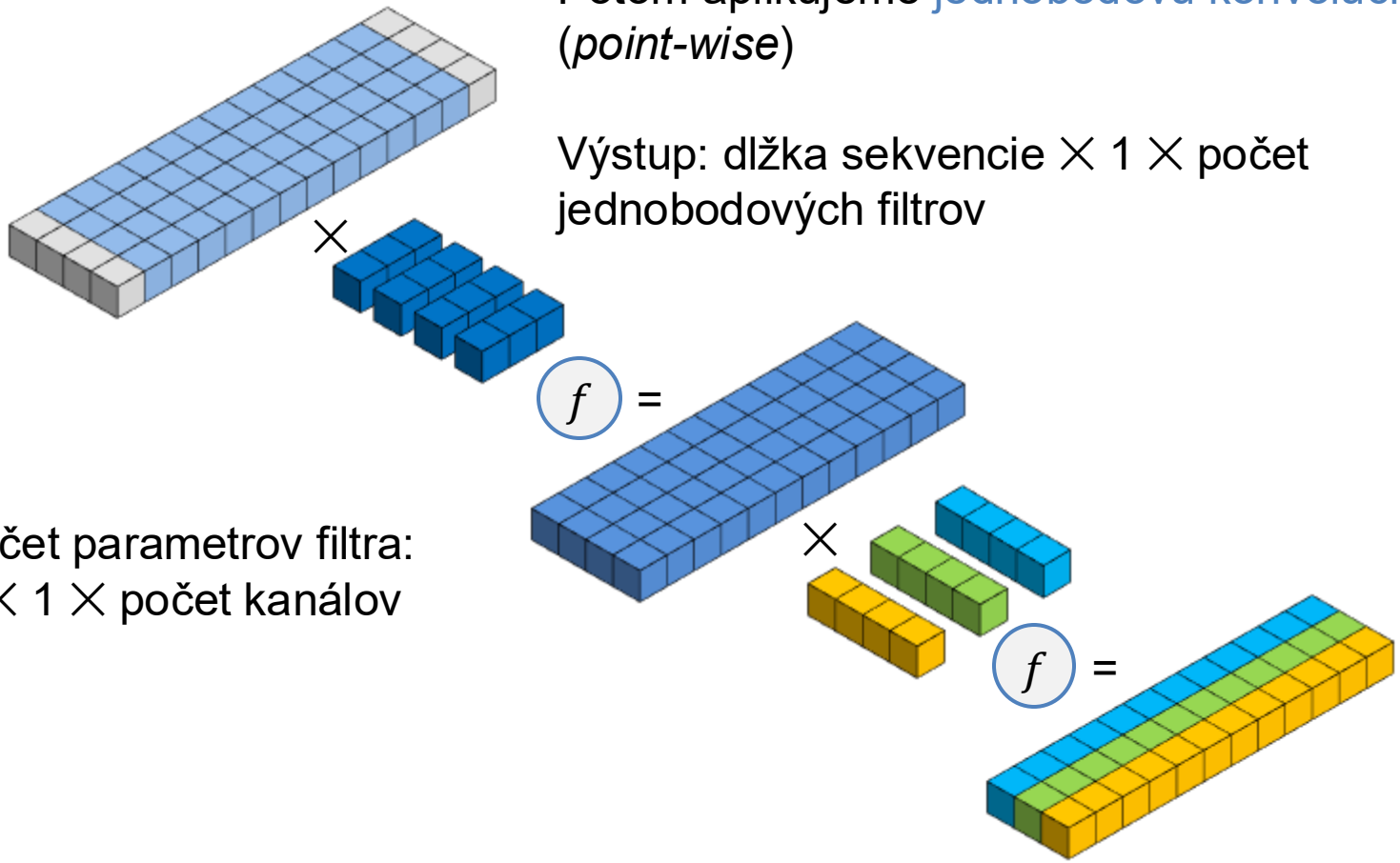
Filtre sa aplikujú samostatne na každom kanály
Výstup: dlžka sekvencie × 1 × počet kanálov

Počet parametrov filtra:
šírka okna × 1 × 1

Depth-wise a *point-wise* 1D konvolúcia (3)

Potom aplikujeme jednobodovú konvolúciu (*point-wise*)

Výstup: dlžka sekvencie × 1 × počet jednobodových filtrov



Počet parametrov filtra:
1 × 1 × počet kanálov

Conformer – konvolučný blok

- Konvolúcia do hĺbky najprv vyextrahuje vzory v jednotlivých frekvenčných intervaloch s nelineárnou aktiváciou
- Jednobodová konvolúcia ich lineárne agreguje a znova nelineárne transformuje