XV

Jak stroj uhodne další slovo?

Po této kapitole vysvětlíš tokeny, embeddingy, pozornost i kontextové okno babičce i kolegovi — a v Shannonově hře si změříš vlastní perplexitu.

Zprávy mají často význam… Tyto sémantické aspekty komunikace jsou však pro inženýrský problém irelevantní.

— Claude E. Shannon, *„These semantic aspects of communication are irrelevant to the engineering problem.“ · A Mathematical Theory of Communication, 1948*

Matematik, který počítal Puškina

Třiadvacátého ledna 1913 předstoupil před Císařskou akademii věd v Petrohradě šestapadesátiletý matematik Andrej Andrejevič Markov a přednesl referát o Puškinovi. Literární historik by v něm nenašel jediné slovo o Taťáně, o souboji ani o ruské duši. Markov vzal prvních 20 000 písmen *Evžena Oněgina*, vyškrtal mezery a interpunkci a každé písmeno zařadil do jedné ze dvou přihrádek: samohláska, nebo souhláska. Z nejctěnějších veršů ruské poezie zbyla šňůra korálků o dvou barvách.

Pak počítal. Ručně, celé týdny — text si rozepsal do čtverců deset krát deset písmen a sčítal, kolikrát po samohlásce následuje zase samohláska. Samohlásek našel 8 638, tedy 43 procent. Kdyby písmena byla nezávislá jako hody mincí, vycházelo by dvojic samohláska–samohláska zhruba 3 731. Markov jich napočítal 1 104. Ne o něco méně. Třikrát méně.

Verše tedy nejsou loterie. Písmeno si pamatuje svého předchůdce: po samohlásce přijde další samohláska jen ve 13 procentech případů, po souhlásce v 66. Ta čísla nediktoval Puškin — diktuje je sama stavba jazyka, a dají se změřit s přesností účetní knihy.

Proč se nejlepší ruský matematik své doby týdny hrabal v cizích verších? Kvůli sporu. Moskevský matematik Pavel Někrasov tvrdil, že zákon velkých čísel — záruka, že se průměry z mnoha pozorování ustálí — vyžaduje nezávislost jednotlivých veličin.

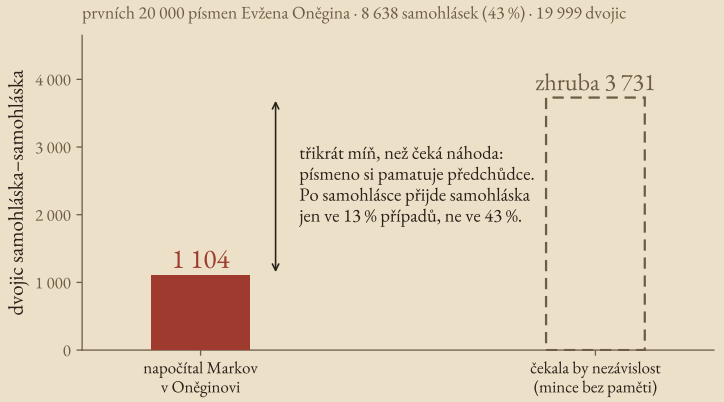
Sňatky, zločiny i sebevraždy přitom ve statistických ročenkách vykazují průměry stabilní jako hodiny; lidská rozhodnutí jsou tudíž nezávislá, a nezávislost, uzavřel Někrasov, je matematický otisk svobodné vůle. Markova — bojovného ateistu, kterému se pro jeho polemiky říkalo „vzteklý Andrej“ a který si po klatbě na Tolstého vyžádal vlastní exkomunikaci z církve — tenhle teologický vpád do matematiky rozzuřil. Roku 1906 dokázal větu: zákon velkých čísel platí i pro veličiny navzájem závislé, spřažené do řetězu. A roku 1913 na Oněginovi předvedl, že takové řetězy nejsou akademická chiméra: dvacet tisíc dokonale závislých písmen, a průměry přesto stojí. Z Někrasovova důkazu svobodné vůle nezůstalo nic.

Zbraň z akademické pře přežila oba soky. Řetězům závislých veličin dnes říkáme Markovovy a stojí na nich předpověď počasí, hodnocení webových stránek — a hlavně každý stroj, který ti doplňuje věty. Když ti telefon nabídne další slovo, běží prapravnuk té tabulky z Oněgina. Tahle kapitola je cesta od 1 104 dvojic korálků k němu.

Pramen: Brian Hayes, „First Links in the Markov Chain“, American Scientist *101(2), 2013; anglický překlad Markovovy přednášky vyšel v* Science in Context *19(4), 2006.*

Pavel Někrasov — matematik, bývalý rektor Moskevské univerzity; nezaměňovat s básníkem Nikolajem Někrasovem. Shoda jmen je náhodná; počítal i tenhle Někrasov, jen hůř.

Zastav se u toho, co přesně Markov vyrobil, protože je to menší a slavnější, než vypadá:



zdroj: A. A. Markov, 1913; Hayes, American Scientist 101(2), 2013

Z celé přednášky zbyla po vyčištění dvě čísla: pravděpodobnost samohlásky po samohlásce, 13 procent, a po souhlásce, 66 procent. A ta dvě čísla jsou — teď se podrž — **jazykový model**: stroj, který se podívá na to, co bylo napsáno, a řekne, jak pravděpodobné je každé možné pokračování. Markovův model má paměť jedno písmeno a slovník dvě přihrádky, ale už umí to hlavní: predikovat. Vsaď po samohlásce na souhlásku a po souhlásce na samohlásku a trefíš zhruba tři čtvrtiny písmen Oněgina; hloupý soupeř z kapitoly 12, který sází pořád souhlá-

Spočítej si to z grafu sám: 43 % písmen jsou samohlásky, po nich vsadíš souhlásku (úspěch 87 %); po souhláskách samohlásku (66 %). Dohromady 0,43 × 0,87 + 0,57 × 0,66 ≈ 0,75.

sku, trefí 57 procent. Dvě čísla vydřená z veršů — a už porážejí toho, kdo strukturu jazyka ignoruje.

Markovův vynález dostal jméno, které uslyšíš dodnes: **Markovův řetěz**. Řetěz proto, že text vzniká článek po článku a každý nový článek se zavěšuje jen na ten poslední — budoucnost nezajímá celá historie, stačí jí přítomný stav. U Markova je stav jediné písmeno: paměť dlouhá jeden článek. A přesto ta miniaturní paměť stačí, aby se z „nezávislých hodů mincí" stal rozpoznatelný stín jazyka.

A nikdo přitom modelu neřekl jediné pravidlo gramatiky. Jen počítal. To je myšlenka, kterou si z podobenství odnes: **statistika textu nese znalost jazyka** — a kdo počítá, ten ji vytěží. Zbytek kapitoly je jedno velké zvětšování téhle věty. Co kdyby paměť nebyla jedno písmeno, ale dvě? Věta? Stránka? A přihrádky ne dvě, ale všechna slova jazyka? Ten směr někdo domyslel do konce — o osmatřicet let později a devět tisíc kilometrů západněji.

Hra, kterou hraješ celý život

Rok 1951, New Jersey. Claude Shannon — inženýr Bellových laboratoří, který tři roky předtím jednou statí založil teorii informace — večer sundá z poličky knihu; podle vzpomínek detektivku Raymonda Chandlera. Otevře ji na náhodné stránce, text zakryje a jeho žena Mary Elizabeth, které nikdo neřekl jinak než Betty, hádá písmeno po písmenu, co následuje. Špatně? Hádej znovu. Shannon sedí vedle a čárkuje počty pokusů.

Pramen: C. E. Shannon, „Prediction and Entropy of Printed English", Bell System Technical Journal *30(1), 1951; scéna s detektivkou: Soni & Goodman,* A Mind at Play*, 2017.*

Vypadá to jako společenská hra pro velmi trpělivé manželství, a je to jeden z nejelegantnějších experimentů dvacátého století. Shannon totiž neměřil Betty — měřil **angličtinu v její hlavě**. Když Betty uhodne písmeno na první pokus, znamená to, že to písmeno bylo z její znalosti jazyka skoro jisté: neneslo žádnou novou informaci. Když se trápí a hádá napodesáté, drží v ruce písmeno, které skutečně něco sděluje. Sečti pokusy přes dost dlouhý text a máš číslo: kolik nejistoty v jazyce doopravdy zbývá, když ho čte někdo, kdo ho umí.

Výsledek stojí za zapamatování. Angličtina má 26 písmen a mezeru, celkem 27 znaků. Kdo hádá naslepo, potřebuje na písmeno v průměru 4,75 **bitu** informace — bit je jedna odpověď na otázku ano/ne, takže necelých pět otázek. Betty — a s ní každý zkušený čtenář angličtiny — potřebovala něco kolem jednoho bitu; Shannon z řady takových měření odhadl rozmezí 0,6 až 1,3. Zbytek — zhruba tři čtvrtiny textu — je redundance: písmena, která tam pravopis a gramatika vyžadují, ale která si čtenář doplní sám. Nevěříš? Přečti si tuhle větu: *pís_ena, kte_á si dopl_íš s_m*. Ušetřil jsem čtyři písmena a nepřišel jsi o nic.

bit — jednotka informace: jedna odpověď na otázku ano/ne. Písmeno hádané naslepo z 27 znaků „stojí" $\log_2 27 \approx 4{,}75$ *otázky; písmeno v ruce zkušeného čtenáře zhruba jednu.*

> *Kolo štěstí je Shannonův experiment převlečený za estrádu — a jeho pravidla jsou bezděčná teorie informace: souhlásky hádáš zadarmo, samohlásky se kupují. Markov by tu cenovou politiku schválil: samohláska nese nejméně informace z celé tabule, takže si kupuješ odpověď, kterou už napůl máš.*

Shannon mimochodem Markova četl a nijak to netajil. Už ve slavné stati z roku 1948 si jeho řetězy vypůjčil a pustil je opačným směrem: místo měření textu ho nechal **vyrábět**. Řetěz bez paměti chrlí písmennou polévku; řetěz, který zná četnosti dvojic, plodí slabiky k nevyslovení; s trojicemi písmen se začnou líhnout útvary, které vypadají skoro anglicky; a řetěz krmený četnostmi dvojic *slov* už skládá věty, které zní správně — a nedávají žádný smysl. Čím delší paměť, tím věrnější jazyk, a nikde ani stopa porozumění. Zapamatuj si ten směr; na konci kapitoly se k němu vrátíme s mnohem větším strojem.

A teď to podstatné: tuhle hru hraješ celý život. Když v hluku hospody slyšíš z kamarádovy věty půlku a přesto víš, co říká, když čteš rukopis lékaře, když doplňuješ křížovku — tvůj mozek předpovídá pokračování textu z jeho statistiky. Neseš v hlavě model svého jazyka, vycvičený desítkami let čtení a poslouchání. Shannonova hra ho měří. A přesně tutéž hru — hádej další kousek textu, za chybu zaplať — hraje při tréninku **velký jazykový model (LLM)**, stroj, kterému dnes říkáš chatbot. Než se podíváme, jak přesně ji hraje, potřebujeme metr: jak srovnat hráče mezi sebou? Číslo, které z téhle hry padá, si za chvíli naměříš sám na sobě, takže stojí za pořádnou definici.

DEMO

`lesk-a-bida-vibecodingu.gaussalgo.com/d/shannon`

Staň se jazykovým modelem: hádej další písmeno českého textu a změř si vlastní perplexitu.

perplexita — průměrné překvapení prediktora přepočtené na „mezi kolika rovnocennými možnostmi jako by vybíral“. Menší = lepší predikce. Opička bušící naslepo do 27 kláves: 27; plynulý čtenář: kolem 2.

EINSTEIN · MATEMATIKA — přeskočitelné

Obodujme hráče Shannonovy hry — člověka i stroj — stejným metrem. Poctivější než chtít jediný tip je nechat hráče rozdat důvěru: $q(x)$ je pravděpodobnost, kterou dal pokračování x. Když pak přijde skutečné písmeno, jeho překvapení měříme

$$\text{překvapení}(x) = -\log_2 q(x)$$

Slovy: čím menší pravděpodobnost jsi pravdě dal, tím víc bitů tě její příchod stojí. Příklady: $q = 1 \to 0$ bitů (věděl jsi to); $q = 1/2 \to 1$ bit; $q = 1/4 \to 2$ bity; $q = 1/27 \to 4{,}75$ bitu, slepé hádání. A kdo dal pravdě nulu, je překvapen nekonečně — proto se prediktorům vyplácí nikdy neříkat nikdy. Průměr přes text délky N:

$$H = -\frac{1}{N}\sum_{i=1}^{N} \log_2 q(x_i)$$

Téhle veličině se říká **křížová entropie** (cross-entropy): průměrná cena pravdy pro tvoje rozdělení q, v bitech na písmeno.

EINSTEIN · MATEMATIKA — přeskočitelné

A protože „1,9 bitu“ si člověk špatně představí, přepočítává se průměrné překvapení H na **perplexitu**:

$$\text{perplexita} = 2^{H}$$

— mezi kolika stejně pravděpodobnými možnostmi jako bys u každého písmene vybíral. Konkrétně z tvojí hry: na stovce písmen jsi padesátkrát dal správnému písmenu $q = 1/2$, třicetkrát $1/4$ a dvacetkrát $1/16$; průměrné překvapení je $(50 \cdot 1 + 30 \cdot 2 + 20 \cdot 4)/100 = 1{,}9$ bitu a perplexita $2^{H} \approx 3{,}7$ — hádáš, jako by ti u každého písmene zbývaly necelé čtyři rovnocenné možnosti.

A pointa, kvůli které tahle věž stojí: křížová entropie je **ztrátová funkce** velkých jazykových modelů. Vzpomeň si na poslední větu věže z kapitoly 10: vyber třídu funkcí, zvol ztrátovou funkci, hledej minimum. Dosaď: třída funkcí = transformery (druhá věž téhle kapitoly), ztrátová funkce = křížová entropie na dalším tokenu, data = biliony tokenů. Trénink LLM je Shannonova hra hraná miliardkrát za vteřinu — se skóre, které se dá derivovat.

→ kap. 2: scaling laws jsou křivky přesně téhle veličiny — křížová entropie na odložených textech klesá s parametry a daty po mocninném zákonu. Pokrok jazykových modelů se měří poklesem průměrného překvapení.

Tabulka by potřebovala vesmír

Markovovi stačila tabulka dva krát dva. Jak hru rozšířit na celý jazyk? Naivní plán zní: větší tabulka. Pro každý možný začátek textu ulož, co následovalo a jak často. Ten stroj nosíš v kapse a můžeš si ho vyzkoušet

hned: otevři na telefonu klávesnici, napiš „Dneska“ a pak už jen pořád mačkej prostřední návrh. Vyjde věta gramaticky hladká a obsahově dutá — slovní Markovův řetěz s pamětí na pár slov, Shannonova výroba textu z roku 1948 v sériové produkci. Přesně tam ale narazíš na strop tabulky: proč má našeptávač paměť jen na pár slov? Spočítej si sklad. Slovník dnešního modelu má zhruba 200 000 kousků a vstup ať má třeba jen sto tokenů — tak se těm kouskům říká a za chvíli je pokřtíme pořádně; možných začátků je $200\ 000^{100} \approx 10^{530}$. Atomů v pozorovatelném vesmíru je asi 10^{80}. I kdyby každý atom vesmíru byl paměťová buňka, chybí ti čtyři sta padesát řádů. A co hůř: i kdyby ses do vesmíru vešel, vyrobil bys jen dokonalého papouška z kapitoly 11 — tabulka umí zopakovat pouze to, co už někdo napsal, a drtivá většina vět, včetně téhle, nebyla nikdy napsána.

→ kap. 10–11: LLM je pořád model s parametry a chybou. Učí se minimalizací, predikuje dosazením — a může se přeučit úplně stejně jako čára přes dvaačtyřicet bytů.

Řešení už znáš z kapitoly 10: místo tabulky **model s parametry**. `model.nauč_se(text, další_token)` — a místo 10^{530} řádků pár set miliard čísel, která se gradientním sestupem nastaví tak, aby průměrné překvapení na tréninkových textech bylo co nejmenší. GPT-3 z roku 2020 měl takových čísel 175 miliard a přečetl při tréninku zhruba 300 miliard tokenů; dnešní modely čtou biliony. Ta komprese je celé kouzlo: nekonečno možných vět se musí vejít do konečného počtu parametrů, model nemá dost místa na šprtání — takže je **donucen hledat pravidla**. Koncovky, vazby, slovosled, a kousek dál i „Paříž je ve Francii“, protože znalost je nakonec taky jen úsporný způsob, jak předpovídat text.

→ kap. 5: a kde se vzaly ty biliony tokenů? Z nás. Web, knihy, Stack Overflow — dvacet let jsme se učili formulovat otázky a odpovědi a stroj se vyučil na nich. Had požírající vlastní ocas.

Zbývá vysvětlit, jak takový stroj uvnitř vypadá. Stačí čtyři pojmy — token, embedding, pozornost, teplota — a všechny čtyři se dají říct u piva.

Nejdřív: co je pro stroj „písmeno“? Písmena samotná jsou moc jemná — než by z nich složil myšlenku, řetěz by se mu rozpadl. Celá slova jsou moc hrubá — čeština skloňuje a časuje, slovník by neměl konec. Kompromis se jmenuje **token**: kousek textu mezi písmenem a slovem. Model si vstup nejdřív rozseká na tokeny a teprve pak ho začne číst.

token — jeden z kousků, na které si model text doopravdy rozseká; ne slovo, ne písmeno. Časté slovo = jeden token; vzácné se drolí na kusy.

TEENAGER · MECHANISMUS

Jak vypadá české slovo očima stroje:

```
tokeny =
rozsekej("Nejneobhospodařovávatelnějšími")
→ Ne·jne·obh·osp·oda·ř·ová·vat·eln·ější·mi
(11 tokenů)
```

Slovník tokenů si stroj vyrobil sám: prošel korpus a slepoval nejčastější sousedící znaky tak dlouho, až měl zhruba 200 000 kousků. Co je časté, dostane vlastní token; co je vzácné, rozbije se na drť. Anglické `prediction` je jeden token, česká `předpověď` se rozpadne na pět: `p·řed·p·ově·ď`. Všimni si taky, že model pak nikdy nevidí písmena, jen čísla tokenů — proto stroje, které píší eseje, umějí zaváhat nad otázkou, kolik je ve slově „jahoda“ písmen a. Příklady jsou z tokenizéru o200k (rodina GPT-4o a nástupců), stav k červenci 2026 — tokenizéry se mění s generacemi modelů a tyhle řádky zestárnou; princip drolení ne.

Proč čeština stojí víc tokenů? Tokenizér se učil na korpusu, kde vládne angličtina: časté anglické sekvence si vysloužily vlastní tokeny, česká slova se drolí. Stejné sdělení o zítřejším počasí: anglicky 17 tokenů, česky 37 (o200k, červenec 2026). A token je účetní jednotka — platí se jím místo v paměti modelu i cena odpovědi.

Číslo tokenu samo o sobě nic neznamená: token 71 403 není o nic „psovitější“ než token 12. První, co model s tokenem udělá, je proto překlad do souřadnic.

TEENAGER · MECHANISMUS

Embedding je poloha tokenu na mapě významů: vektor o stovkách až tisících souřadnic, který se model naučil sám z toho, *v jakých kontextech se token objevuje*. Tokeny s podobnými sousedy skončí blízko sebe: „pes“ u „psa“ i u „štěká“, „úrok“ u „hypotéky“. A protože je to mapa, dá se po ní počítat:

```
souřadnice("král") - souřadnice("muž") +
souřadnice("žena")
→ nejbližší token: "královna"
```

Směr „mužský → ženský“ je na mapě skutečná šipka a táž aritmetika najde i Paříž → Francie nebo Praha → Česko. Je to Beckova mapa slovníku (kapitola 10): souřadnice lžou o všem kromě jediného, na čem záleží — kdo se vyskytuje s kým.

V praxi: slavný příklad král – muž + žena pochází z modelu word2vec (Mikolov et al., 2013). V transformeru se souřadnice učí spolu se vším ostatním, gradientním sestupem z kapitoly 10.

Jenže slovo nemá význam samo o sobě. „Zámek“ u Hluboké a zámek u dveří sdílejí jeden token, jednu výchozí souřadnici — a dva různé významy. Embedding je poloha slova *bez kontextu*; něco musí nechat okolí, aby s ní pohnulo. To něco je nejdůležitější vynález celé architektury.

TEENAGER · MECHANISMUS

Mechanismus **pozornosti** (attention): každé slovo se zeptá všech ostatních, co je pro ně důležité. Každý token k tomu vysílá **dotaz** („co hledám“), nabízí **vizitku** („co jsem“) a nese **náklad** („co předám dál“).

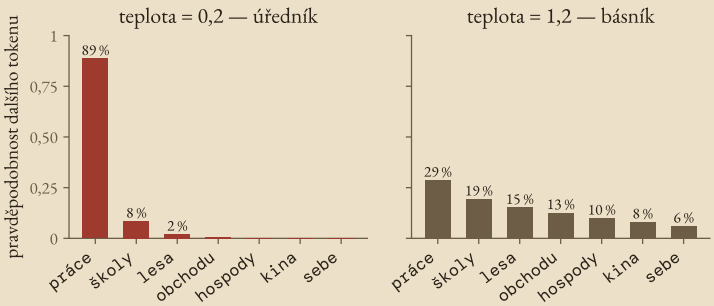
```
pro každý token:
    shody  = jeho dotaz · vizitky všech tokenů
    váhy   = na_pravděpodobnosti(shody)
    význam = vážený_průměr(nákladů, váhy)
```

Ve větě „Klíč nešel do zámku, protože byl ohnutý“ vyšle token „byl“ dotaz po podmětu; vizitka „klíče“ ladí (ohnutý bývá klíč, ne zámek), takže si „byl“ stáhne hlavně jeho náklad. Napiš „protože byl zamčený“ a tytéž váhy se přelijí k „zámku“. Tohle se děje pro všechny tokeny najednou a v desítkách vrstev nad sebou: v první si „zámek“ přitáhne „klíč“, o pár vrstev výš už celá věta „ví“, že je o dveřích, a ne o Hluboké.

V praxi: architektura transformer — stať „Attention Is All You Need“ (Vaswani et al., 2017). Česky nepřekládat: transformátor bzučí v trafostanici.

Model tedy přečte vstup, slova si vyjasní významy — a výstup? Tady čeká poslední překvapení: model **neodpovídá slovem**. Odpovídá rozdělením: seznamem všech 200 000 tokenů slovníku, každý se svou pravděpodobností. Někdo z toho seznamu musí vybrat. A jak se vybírá, rozhoduje knoflík s termodynamickým jménem:

„Karel šel do ___“ · týž model, jiná teplota (pravděpodobnosti ilustrativní)



TEENAGER · MECHANISMUS

```
další_token = losuj(rozdělení, teplota = 0,7)
```

Teplota přeostřuje rozdělení před losem. Nízká (vlevo): skoro vždy padne nejpravděpodobnější token — úředník: spolehlivý, opakovatelný, nudný; při dlouhém textu sklouzává do smyček. Vysoká (vpravo): rozdělení se rozprostře — básník: překvapivý, občas úplně mimo. Proto ti tatáž otázka dá dnes jinou odpověď než včera: není to porucha, je to losování, a dá se vypnout. Vzoreček najdeš ve věži o stránku dál.

Na grafu si všimni: teplota nemění pořadí kandidátů, jen kontrast mezi nimi. Úředník i básník preferují „práce“ — básník si jen častěji dovolí odbočku.

Poskládej to dohromady a máš celý stroj: rozsekej vstup na tokeny, přelož na souřadnice, nech vrstvy pozornosti významy vyjasnit, přečti rozdělení dalšího tokenu, vylosuj podle teploty, přilep k textu — a **opakuj**, token po tokenu, dokud nevznikne odpověď. Nic víc v tom není.

U toho „opakuj“ se na vteřinu zastav, protože v něm bydlí vlastnost, která tě v praxi překvapí. Přilepený token se stává součástí vstupu — model v každém kroku čte i **svoje vlastní předchozí slova** a předpovídá pokračování celku. Proto odpověď drží nit: každý další token se podmiňuje vším, co už padlo. Ale ze stejného důvodu se drží i chyba — když los na začátku odpovědi vytáhne nešťastný token, všechno další se předpovídá jako pokračování textu, ve kterém ta chyba stojí. Stroj necouvá; umí jen dál. Vzpomeň si na to, až uvidíš odpověď, která se zkraje drobně splete a pak tu chybu deset odstavců sebejistě rozvíjí. Právě proto stojí za to vylézt na věž a podívat se té jediné rovnici, která to celé pohání, do očí.

EINSTEIN · MATEMATIKA — přeskočitelné

Pozornost krok za krokem. Z embeddingu každého tokenu i si model třemi naučenými maticemi vyrobí tři vektory: dotaz q_i („co hledám“), klíč k_i (vizitka: „co jsem“) a hodnotu v_i (náklad: „co předám dál“). Všechny mají d souřadnic. Shoda dotazu tokenu i s vizitkou tokenu j je skalární součin, zkrocený odmocninou z rozměru:

$$s_{ij} = \frac{q_i \cdot k_j}{\sqrt{d}}$$

Slovy: $q_i \cdot k_j$ měří souznění dvou směrů na mapě (velké = „to hledám!“), a dělí se $\sqrt{d}$, protože s rostoucím počtem souřadnic skalární součiny bobtnají a bez zkrocení by další krok zdegeneroval na jednobodovou volbu. Skóre se pak převedou na váhy funkcí softmax:

$$a_{ij} = \frac{e^{s_{ij}}}{\sum_{l=1}^{n} e^{s_{il}}}$$

Slovy: umocnění (e^s) zesílí rozdíly a zaručí kladnost, dělení součtem přes všech n tokenů vstupu zaručí, že váhy a_{ij} dají dohromady jedničku — token i rozdává sto procent své pozornosti.

EINSTEIN · MATEMATIKA — přeskočitelné

Sklizeň a celá rovnice. Nový význam tokenu i je vážený průměr nákladů všech tokenů:

$$z_i = \sum_{j=1}^{n} a_{ij} v_j$$

Token si stáhne především to, co ladí s jeho dotazem — hospodská intuice z dílny, jen se sečtenými vektory. Naskládej teď dotazy, klíče a hodnoty všech n tokenů po řádcích do matic Q, K, V a všechny tři kroky se slijí do jednoho řádku — nejcitovanějšího vzorce současné informatiky:

$$\text{Pozornost}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d}}\right)V$$

Čti zevnitř ven: $QK^\top$ je tabulka shod všech dotazů se všemi vizitkami, $\sqrt{d}$ ji krotí, softmax z každého řádku udělá váhy a násobení V posbírá náklady. Ta tabulka má $n \times n$ polí — a to je ta **kvadratická cena kontextu**, $O(n^2)$: dvakrát delší text, čtyřikrát víc porovnávání i paměti.

EINSTEIN · MATEMATIKA — přeskočitelné

Dvě poznámky a slíbený vzoreček. Zaprvé, skalární součiny neznají pořadí — pozornost sama o sobě vidí pytel slov, takže se do souřadnic tokenů přimíchává i jejich pozice (**poziční kódování**); jinak by „pes kousl pošťáka“ a „pošťák kousl psa“ byly táž věta. Zadruhé, tohle celé běží v desítkách „hlav“ vedle sebe a desítkách vrstev nad sebou; každá hlava se naučí ptát jinak — jedna na podměty, jiná na závorky.

A slíbená teplota T: surová skóre tokenů slovníku (logity u_i) se před losováním dělí teplotou,

$$p_i = \frac{e^{u_i/T}}{\sum_j e^{u_j/T}}$$

takže $T \to 0$ rozdělení zostří k jistotě (úředník), velké T je rozprostře (básník) a $T = 1$ vrací model tak, jak byl natrénován.

Pracovní stůl, ne knihovna

Poslední pojem je nejdůležitější pro praxi — a křtíme u něj metaforu, která tě doprovodí až do konce knihy. Všechno, co model ví o tobě a o téhle chvíli, mu musí ležet před očima: tvůj **prompt** — text, který mu položíš — předchozí průběh rozhovoru, vložené dokumenty. Prostor na to všechno se jmenuje **kontextové okno** a představuj si ho jako **pracovní stůl, ne knihovnu**. Model nemá polici, kam by si odkládal včerejší rozhovory: co neleží na stole, neexistuje. GPT-3 měl stůl na 2 048 tokenů, pár stránek textu. Dnešní modely mívají stoly na stovky tisíc tokenů, celé knihy — jenže pořád je to stůl: konečný, placený od tokenu, a jak víš z věže, kvadraticky drahý na roztahování.

→ část II: až budeš krmit agenta svým repozitářem, bude tohle nejdůležitější věta kapitoly. Nerozhoduje, co model „ví“ — rozhoduje, co mu položíš na stůl. Celé řemeslo kontextu je úklid stolu.

Model má tedy vědění dvojího druhu, a vyplatí se je nezaměňovat. V **parametrech** nese destilát korpusu — vzpomínky bez data a bez pramene, zapečené při tréninku a od té chvíle neměnné. Na **stole** má jediné, co čte doslova a teď. Když si chatbot „pamatuje“ tvoje jméno z minulého týdne, nenašel ho v knihovně — aplikace mu ho potichu položila na stůl. A když dlouhý rozhovor začne ztrácet nit a model si plete, co jsi říkal na začátku, nezhoršila se mu paměť: nejstarší listy prostě spadly ze stolu. Nikdy je neměl „v hlavě“ — měl je na desce, a deska je plná.

Z té dvojice plyne praxe, kterou budeš používat denně. Nová konverzace znamená čistý stůl — všechno, na čem záleží, tam musíš položit znovu, a je to výhoda: čistý stůl je jediný spolehlivý restart. Když chceš odpověď o **svém** dokumentu, polož ho na stůl celý, místo abys doufal, že si model „vzpomene“ z parametrů — vzpomínka bez data a pramene je přesně ten materiál, ze kterého se plynule dopisuje. A když se dlouhá

seance začne kazit, nehádej se s ní: shrň, co platí, a začni s tím shrnutím na čistém stole. Neuklizený stůl není hřích proti etiketě, je to statistika — model předpovídá pokračování *všeho*, co na stole leží, včetně tvých slepých uliček.

Plynulost není pravda

Projdi ten řetěz ještě jednou, pomalu: tokeny, souřadnice, pozornost, rozdělení, los, další token. Našel jsi v něm krok „ověř, jestli je to pravda“? Není tam. Model při tréninku minimalizoval jediné číslo, křížovou entropii — průměrné překvapení nad texty korpusu. Umí dokonale to, co ta chyba měří: znít jako pokračování. Pravda do rovnic nevstoupila ani jednou.

Většinou to kupodivu nevadí — a právě to mate. Nejpravděpodobnější pokračování věty „dva plus dva jsou“ je „čtyři“, protože korpus je plný pravdivých vět; pravda bývá na predikci levná, a tak se pravděpodobné s pravdivým skoro pořád shoduje. Ale „skoro pořád“ není „vždy“. Chtěj po modelu pramen a dostaneš dokonale formátovanou citaci: příjmení, rok, ročník časopisu, čísla stran. Občas i existující. Model nelže — lež předpokládá záměr. Model **sebejistě doplňuje**: vyrábí nejpravděpodobnější tvar odpovědi, a nejpravděpodobnější tvar citace je prostě citace, ne mlčení. Plynulost, kterou tě ohromuje, je vlastnost pravděpodobností — kvalita mapy textů, ne mapy světa.

→ kap. 3: plynulá řeč je pro nás od pravěku důkaz mysli za slovy. Stroj, který plynule mluví, spouští týž reflex jako tvář v mracích — největší pareidolie v dějinách.

Rozdíl se dá říct i číslem. Když model dává tokenu „čtyři“ pravděpodobnost 0,97, neznamená to „na 97 % platí, že dva plus dva jsou čtyři“. Znamená to „v korpusu po takovémhle textu následovalo ‚čtyři‘ v 97 % případů“. Je to míra řeči, ne míra světa — a splétat ty dvě míry dohromady je nejdražší zkratka, jakou dnes v práci s těmihle stroji můžeš udělat. Co s tím rozdílem provede výcvik na lidský potlesk — proč ti model přitakává, proč si vymýšlí nejochotněji právě tehdy, když se ptáš nejnaléhavěji, a co se s tím dá dělat — to je celá příští kapitola.

→ kap. 16: RLHF (posilované učení z lidské zpětné vazby) — výcvik na potlesk. Vezme rozdělení z téhle kapitoly a přiohne ho k tomu, co se líbí lidem. Plynulost tím ještě stoupne; pravda ne nutně.

„Jak bych poznal, že se pletu?“

Zahraj si Shannonovu hru dvakrát (demo výše): jednou na úryvku z Máchova *Máje*, jednou na svém včerejším pracovním e-mailu. Změř si obě perplexity. Tvrdím, že na e-mailu budeš výrazně lepší prediktor než na národním klenotu — protože tvůj vnitřní model se trénoval na tvém korpusu: na poradách, ne na klasicích. Vyjde-li ti perplexita na obou textech stejná, kapitola se plete a žádný statistický model jazyka v hlavě nenosíš. A vyjde-li rozdíl, víš od této chvíle z první ruky, co znamená „záleží, na čem byl trénovaný“ — příště si na to vzpomeň, než se stroje zeptáš na něco, co v jeho korpusu skoro nebylo.