Korpus studentských zdrojových kódů

Jiří Fišer, Jiří Škvor
katedra informatiky PřF UJEP Ústí nad Labem

Korpus – nástroj moderní lingvistiky

jazykový korpus

- **teoretický**: všechny mluvené a psané promluvy v daném jazyce
- **univerzální**: rozsáhlá strukturovaná o metadata rozšířená databáze autentických **textů** (ČNK)
- **specializovaný**: jen určitý typ promluv

Univerzální jazykové korpusy

- **synchronní** (aktuální stav jazyka)

  ČNK SYN (1990-2022) 4,9 mld. slov, ORAL 5,4 mil.

- **diachronní** (vývoj)

  ČNK DIAKORP 3,4 mil slov

Klíčové aspekty jazykových korpusů

- *representativnost* (nikoliv jen „dobří“ autoři)
- *vyváženost*
- *segmentace* (věty, slova, morfémy)
- *lemmatizace s disambiguací* (homonyma)
- *značkování* (slovní tvary, důležitý je model!)

representativnost: 1
reprezentativnost: 563

Několik výsledků založených na ČNK

- podstatná jména ženského rodu — vzory: *žena*, *růže*, *píseň*, *kost*
  ve skutečnosti mnoho (pod)vzorů
  krácení: dáma, šťáva, vrána, rána, váha, žíla, lípa, díra, víra, houba
  vkladné e: hra, růže; 2pl. ulice, expedice, chvíle
  píseň/kost: čelist, moc, past, lež, myš, pravomoc/noc, čtvrť a huť
  žena/růže: studna, sója, idea
- tvar byl + infinitiv, tzv. *absentiv* (byl navštívit)

Korpusy programovacích jazyků

- programovací jazyky jsou **lidské jazyky** (slouží ke komunikaci mezi lidmi) – obdoba matematické notace
- programovací jazyky mají dostatečně **komplexní syntaxi** (v užití vnořených konstrukcí překonávají přirozené jazyky)
- vyvíjejí se v čase a nabízejí **variantní konstrukce** (v některých případech lze mluvit o vzniku spontánních dialektů)

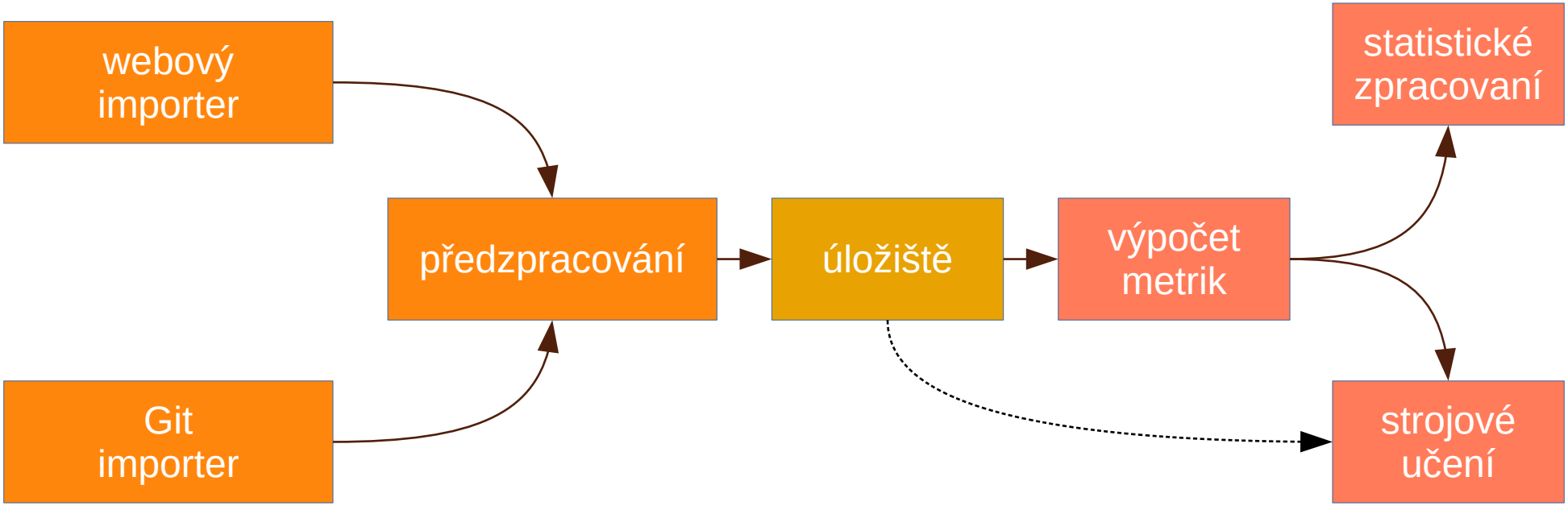
Korpus studentských zdrojových kódů

- specializovaný korpus (méně teoretických i praktických problémů)
- lze snadněji definovat cíle výzkumu a získávat **metadata**
  - předběžné znalosti
  - sociologická data (zázemí)
  - vzájemně se ovlivňující skupiny (třídy)
- přispěvatelé
  - studenti SŠ s výukou programování,
    bakalářské obory: UJEP Ústí nad Labem, ČVUT Děčín

Možné otázky

- Jak výuku ovlivňuje znalost ostatních programovacích jazyků, jak na straně vyučujících, tak vyučovaných?
- Jaké konstrukce studenti využívají v různých fázích výuky? (např. z hlediska paradigmatu či doby od zavedení v jazyce)
- Jak roste komplexnost programů, a to jak u jednotlivých studentů, tak i jejich skupin?
- Jaké konstrukce, resp. moduly jsou využívány obecně a jaké jsou specifické pro různé typy studia, resp. kurzů?

Architektura korpusu



Webový importer

- jednoduchá webová aplikace podporující import jednotlivých zdrojových souborů
- metadata
  - identifikátor studenta (dig. otisk e-mailu), persistence SŠ ⟶ VŠ
  - skupina
  - předběžné znalosti
  - rodinné zázemí (z hlediska uplatnění IT)

Podpora programovacích jazyků

v první fázi: **Python** a **C#**

- oba jsou využívány pro výuku v Ústeckém kraji
- procedurálně objektové paradigma
  (statické × dynamické typování, typové anotace)
- transformace do AST ve standardních knihovnách

v dalších fázích: Java, Javascript, R, …

Preprocesing

- extrakce zdrojového kódu (Jupyter notebooky)
- transformace do abstraktního syntaktického stromu
  - kontrola syntaktické správnosti
  - kontrola sémantické správnosti? (řešení: *doctest*, *jednotný kontext - dockerizace*)
- serializace AST do strukturovaného XML

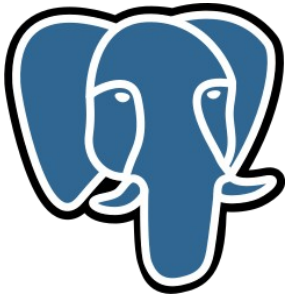
Úložiště

- metadata, zdrojové kódy a serializované AST

  (AST neobsahuje formátování a komentáře)

- databázový systém s integrací XML (AST) a JSON daty (metadata)

  PostgreSQL



Metriky

- pro další zpracování je vhodné extrahovat (číselné) metriky
- klasické metriky využívané při vývoji softwaru (*Quality Assurance*)
- **ad hoc metriky** (použití různých konstrukcí a jejich kontextu)

  příklad: podíl while(for)/foreach cyklů, použití explicitního polymorfismu, kontext použití n-tic

Zpracování dat

- použití standardních statistických metod
- **použití strojového učení** (klasifikace, shluková analýza)
- budoucnost:

  strojové učení nad AST (obdoba Natural Language Processing), viz například současné „inteligentní“ generátory kódu (*Github Copilot*)

Spolupráce se SŠ učiteli

- je nutné učitele přesvědčit, že cílem není hodnocení žáků a pedagogů, a dokonce ani určení, jaké je jediné správné programovací paradigma, jazyková konstrukce apod.
- je nezbytné aktivní zapojení učitelů (řešení obdobných úloh, využití výsledků, podíl na publikování výsledků)