FAKULTA APLIKOVANÝCH VĚD
ZÁPADOČESKÉ UNIVERZITY
V PLZNI

Semestrální práce z KKY/HDS

Fonetická transkripce textu

Dominik Zappe – A23N0011P
(zapped99@students.zcu.cz)

Březen 2024

Obsah

1 Implementace **1**
1.1 Použitá a implementovaná pravidla 1
1.2 Vynechaná pravidla 3

2 Problémy a jejich řešení **4**

3 Uživatelská příručka **5**
3.1 Spuštění 5

4 Závěr **6**

1 Implementace

Implementace využívá jednoduchého *parseru* pravidel, která jsou zapsána v souboru *rules.json*. Nedílnou součástí parseru je také soubor *epa.json*, který obsahuje definice symbolů jako např. V (*vowel*). Obyčejná pravidla využívájí strukturu slovníku – na klíči se nachází nahrazovaný a hledaný řetězec, hodnotou je pak řetězec, kterým má být původní řetězec nahrazen. Jelikož jsou často pravidla v dodané knize symbolická (viz uvedený příklad symbolu V (*vowel*)), byl parser také obohacen o možnost rozklíčovat symbolická pravidla. Symboly jsou psány mezi *špičaté závorky*, tedy ⟨V⟩ (pro *vowel* symbol). Většina pravidel funguje jen díky tomu, že symboly pro *znělé* a *neznělé* hlásky jsou zapsány seřazeně (pochopitelně stejně), takže znaky na daných indexech si odpovídají ve *znělostních dvojicích*. Označení pravidel v implementaci často odpovídá kapitolám v dodané knize, ze které se vychází. Text se v průběhu aplikací pravidel přepisuje a tedy výstup jednoho pravidla je vstupem pro další pravidlo – nevychází se z původních znaků a pravidly tedy často počítají s již přepsanými znaky z fonetické abecedy *EPA* (např. „*přes*“ a „*pQes*“).

1.1 Použitá a implementovaná pravidla

Před aplikací pravidel je původní text odekorován zleva a zprava řetězcem „|$|“. Text je rovněž pro jistotu převeden na malé znaky.

Jako první je aplikováno pravidlo *preprocess*, které jako jediné nepřebírá název na základě dodané knihy. Toto pravidlo primárně nahrazuje mezery za kůly (svislítka), dále maže interpunkci (respektive čárky nahrazuje znakem „#“) a maže další časté speciální znaky jako dvojtečky, uvozovky atp. Nakonec je uplatněno symbolické pravidlo přepisu několika častých cizích slov obsahujících měkké i, ve kterých však nemá dojít ke „změkčení“ souhlásky před „i“ (např. „cynik“), viz. symboly v *epa.json* ⟨FOREIGN⟩ a ⟨CZENGLISH⟩.

Dalším pravidlem je pravidlo 2.8.3.1 (pravidlo 2.8.3 bylo rozděleno na tři části z důvodu pořadí). Toto pravidlo zajišťuje přepiš českých speciálních symbolů – diakritika.

Následuje pravidlo 2.8.9.2, které zajišťuje přepis málo častých písmen – „w“, „q“, „x“. Zajímavé je pouze písmeno x, které se někdy přepisuje na „ks“, jindy na „gz“.

Jelikož už jsou přepsaná původní písmena „w“ a „x“, je nyní možné uplatnit pravidlo 2.8.3.2 a přepsat „ch“ na x a „dz“ (resp. „dž“) na w (resp. W). V tomto pravidle je poprvé vidět často užívaná technika u symbolických pravidel – dekorování za účelem vyhnutí se uplatnění nějakých pravidel. Pokud slovo začíná na předponu končící na písmeno „d“ a hned za ním je písmeno „z“ nebo „ž“ není potom uplatněn přepis na „w“ (resp. „W“) pomocí dekorace hvězdičkou. Tedy mezi „d“ a „z“ je vložen znak hvězdičky, potom se nejedná o sousedící hlásky „d“ a „z“ a tedy nemohou být spojeny ve „w“.

Následně je uplatněno pravidlo 2.8.5, které se primárně stará o „měkčení“ hlásek před měkkým i (resp e s háčkem („ě“)). Mimo to se ale také pravidlo stará o vkládání rázů – zejména na předělu slov, začíná-li druhé slovo samohláskou.

Pravidlo 2.8.6 je rozsahově malé a stará se o vkládání rázu po pauze – začína-li slovo samohláskou.

Dalším pravidlem je pravidlo 2.8.7.1.1 (pravidlo asimilace 2.8.7.1 bylo rozděleno do dvou částí z důvodu vyřešení precendence levého kontextu u výjimky znělosti hlásky „ř“). Toto pravidlo pouze asimiluje znělé „ř“ na neznělé „ř“, je-li před ním neznělá souhláska.

Druhá část asimilačního pravidla (2.8.7.1.2) je největším pravidlem v implementaci. Výjimky asimilace jsou ošetřeny pořadím, tedy nejprve se řeší výjimky, pak klasická asimilace znělosti. Nejprve jsou ošetřeny speciálně slova *shora*, *shůry* a *shluk*. Poté jsou veškerá „sh“ přepsána na „sch“ (EPA: „sx“). Slovo *přes* se navíc vyslovuje, jako kdyby končilo na „z“, což výrazně zjednoduší práci s jednoslabičnými předložkami, které jinak všechny končí na znělou párovou souhlásku. V pravidlu je hojně užit trik s dekorováním písmen hvězdičkami, aby se zabránilo případné asimilaci, když by k ní z nějakého výjimečného důvodu docházet nemělo (např. neslabičné předložky následované písmenem „v“). Na závěr asimilace je opět vynuceno přepsání znělého „ř“ na neznělé „ř“, je-li to nutné (např. slovní spojení „uvnitř hospody“ – „ř“ má zleva neznělé „t“ a zprava znělé „h“ – díky pravidlu 2.8.7.1.1 je sice „ř“ neznělé, ale po asimilaci zezní kvůli následujícímu „h“. Je proto nutné jeho neznělost opět vynutit, aby levý kontext měl přednost před pravým a „ř“ tak dále nenechávalo pokračovat asimilaci i přes něj). Hlavně kvůli tomuto pravidlu byl implementován systém opětovné aplikace pravidla, dokud se to jakkoliv projevuje na výstupu po aplikaci pravidla, neboť asimilace se zpětně může šířit slovem přes více kroků.

Následuje rozsahově malé pravidlo 2.8.7.2, které se pouze stará o *nosové* hlásky „n“ a „m“

Další pravidlo 2.8.7.3 se stará o *slabikotvorné* verze písmen „r“, „l“ a „m“.

Nyní již konečně může být ypsilon přepsáno na měkké „i“ díky pravidlu 2.8.3.3, neboť všechna pravidla využívající samohlásky „i“ a „y“ už byla použita.

Následně je aplikováno pravidlo 2.8.7.4, které už nemá moc velký vliv – spíše se jedná o autorovu preferenci výslovnosti slov jako „prázdniny“ („prázniny“ bez „d“) atp.

Poslední aplikované pravidlo je zjednodušené pravidlo „diftongů“. Pravidlo 2.8.4 je možné aplikovat hlavně díky tomu, že ypsilon je již přepsané na „i“. Zjednodušení spočívá v tom, že „au“, „eu“ a „ou“ jsou vždy přepsané jako dvojhlásky, neboť detekce kdy by se tomu tak stát nemělo je těžká. V tomto pravidle jsou rovněž symboly „ˆ“ přepsány na ráz – jelikož se jedná o poslední pravidlo.

1.2 Vynechaná pravidla

Některá pravidla zůstala neuplatněná hlavně z důvodu, že by mohla měnit původní význam slov. Dalším důvodem někdy byla osobní výslovnost slov autora.

V kapitole 2.8.7.2 byla vynechána pravidla 2.35 a 2.36 z důvodu osobní preference autora při výslovnost slov jako „špatně“ a „anděl“. Pravidla 2.37, 2.38 a 2.39 byla vynechána hlavně z důvodu, že v jistých případech je vyžadována plná výslovnost. Konkrétně pravidla pro slova končící na „t“ (resp. „d“), kterým následují slova začínající na „s“, „š“ (resp. „z“, „ž“), byla vynechána na doporučení přednášejícího. Podle Zdeny Palkové (Fonetika a fonologie češtiny, 1994) by se tato pravidla určitě uplatňovat měla, ale po poradě s přednášejícím byla tato pravidla na doporučení nevyužita, údajně jsou v dodané knize zmíněny pouze kvůli spolupráci s fonetiky.

V kapitole 2.8.7.4 bylo vynecháno pravidlo 2.42 opět hlavně z důvodu osobní preference výslovnosti autora. Pravidlo 2.44 bylo vynecháno z důvodu možné aplikace pouze pokud nedojde ke změně původního významu – to se detekuje těžce. Dalším vynechaným pravidlem, které je tentokrát nečíslované, je pravidlo okolo zjednodušené výslovnosti slovesa „býti“ – pravidla je možné opět uplatnit pouze tehdy, když není sloveso „býti“ plnovýznamové – opět těžká detekce plnovýznamovosti slovesa. Další vynechaná nečíslovaná pravidla jsou pravidla pro jinou výslovnost číslic „sedm“ a „osm“, vynechávání „j“ v imperativech „přijď“ a „přijďte“, vynechávání „d“ ve slovech „džbán“ a „džber“ – vynecháno hlavně z důvodu osobní preference výslovnosti autora.

Celá kapitola 2.8.7.5 byla více méně vynechána, neboť pravidla jsou opět aplikovatelná jen tehdy, když nemění původní význam slov (resp. někdy je zjednodušená výslovnost považována za nedbalou až nespisovnou).

Posledním vynechaným pravidlem je pravidlo 2.81 v kapitole 2.8.9.1. Není přesně specifikováno, kdy přepisovat „i“ na „j“, a kdy na „ji“. Vyjmenované příklady jsou v implementaci ošetřeny pomocí symbolů ⟨FOREIGN⟩ a ⟨CZENGLISH⟩.

2 Problémy a jejich řešení

První problém při implementaci byla kombinace *pořadí* pravidel s fonetickou abecedou EPA. Kdyby se nejprve uplatnilo pravidlu přepisu ypsilon na „i“, a až poté se řešilo „měkčení“ hlásek před měkkým i, docházelo by k problémům. Obdobně je tomu tak např. s českým speciálním „ch“, které se přepisuje podle abecedy EPA na „x“ – je tedy nutné nejprve přepsat všechna originální „x“, a až poté řešit „ch“.

Dalším problémem byly výjimky (nejvíce u pravidla asimilace). Většina výjimek byla však vyřešena pořadím pravidel (nejprve zpracovat výjimky, poté obecnější pravidla), zbytek byl vyřešen odekorováním levého (resp. pravého) kontextu písmene asteriskem – např. výjimky v asimilaci pro písmeno „v“.

V neposlední řadě byl někdy problém s dodanou knihou pravidel. Pravidlo 2.20 v kapitole 2.8.5 o přepisu „ě“ na „je“ uvádí chybně levý kontext, neboť jsou vyjmenovány pouze hlásky „b“, „p“ a „v“, ale v příkladu je i hláska „f“ („harfě“). V kapitole 2.8.7.1 pravidlo 2.29 uvádí, že při asimilaci se „ř“ chová výjimečně a řídí se mimo pravý kontext taktéž kontextem levým. Nikde však není zmíněna precedence kontextů – např. „uvnitř hospody“, „ř“ je zde obklopeno neznělým „t“ zleva a znělým „h“ zprava – tento problém byl vyřešen domluvou s přednášejícím a přednost má levý kontext před pravým. V kapitole 2.8.7.3 je chybně popsáno pravidlo o slabikotvorném „m“ (pravidlo 2.40). Je uvedeno, že slabikotvorné „m“ vzniká za levého kontextu jakékoliv souhlásky a za pravého kontextu jakékoliv souhlásky kromě hlásek „r“ a „l“ – protipříkladem může být slovo „reformní“, kde je „m“ obklopeno kontexty, které odpovídají pravidlu, avšak „m“ ve slově „reformní“ nemá být slabikotvorné. Tento problém byl vyřešen opět poradou s přednášejícím, kde údajně slabikotvorné „m“ vzniká pouze ve slovech „sedm“ a „osm“ (resp. další slova odvozená – „sedmnáct“, „osmnáct“…). Pravidlo 2.31 v kapitole 2.8.7.1 je dobře popsáno v textu, ale zápis je poměrně zavádějící, neboť se jedná o pravidlo o posledním písmenu v jednoslabičným předložkách, ale zápis naznačuje písmeno následující za celou jednoslabičnou předložkou, nikoliv poslední písmeno dané předložky.

Dalším problémem mohou být pravidla, která se mají uplatnit pouze v případě, že nedochází ke změně původního významu slov – bylo by zapotřebí implementovat klasifikátor / detektor významů. Tento problém tedy nebyl řešen nijak a tato pravidla nebyla implementována.

Závěrem kapitoly bych chtěl taktéž jako problém uvést velikost dodané testovací sady. Deset vět je poměrně málo pro kontrolu správnosti vlastního přepisu. Navíc dodaných deset přepisů obsahuje poměrně málo pravidel z dodané knihy – je tedy vcelku jednoduché se trefit vlastním přepisem do testovací sady a myslet si, že je práce hotová, ačkoliv tomu tak opravdu není. Částečným řešením tohoto problému bylo zavedení vlastní testovací sady – bohužel však neexistují oficiální přepisovače do abecedy EPA, proto je řešení pouze částečné.

3 Uživatelská příručka

Aplikace nemá žádné vnější závislosti a není tak nutné doinstalovávat žádné knihovny. Všechny užité knihovny jsou součástí základního interpreteru pro Python. Vše bylo testováno na verzi Pythonu 3.9, ale vše by mělo fungovat bezproblémů i na nižších verzích Pythonu 3.

3.1 Spuštění

Spustitelný je především hlavní skript – main.py. Program je možné spustit bez parametrů z příkazové řádky, pak se použijí výchozí cesty pro vstupní a výstupní soubor (viz výpis v konzoli). Druhou možností je spustit program s dvěma parametry z příkazové řádky, první parametr je cesta ke vstupnímu souboru a druhý parametr je cesta k výstupnímu souboru, který bude programem vytvořen (soubor s přepisem do abecedy EPA).

Spuštění na platformě Windows může vypadat následovně:

```
C:\SP\src>python main.py ../data/input.txt ../data/output.txt
```

Spuštění na platformě Linux může vypadat následovně:

```
user@pc:/SP/src$ python3 main.py
```

Po spuštění je v konzoli výpis o relativních cestách užitých v programu. Dále program vypisuje svůj průběh – jestli ještě dochází k fonetické transkripci, či jestli už program doběhl a zda úspěšně.

4 Závěr

Výstupem semestrální práce je aplikace sloužící k fonetickému přepisu textu do abecedy EPA. Práci je těžké zhodnotit, neboť dodaná testovací sada byla poměrně malá a neobsahovala příliš případů z pravidel z dodané knihy. Dále mi metrika hodnocení na základě správně přepsaných slov, či snad i znaků, přijde vhodnější, než hodnocení založené na základě správně přepsaných celých vět.

Do budoucna by se práce dala vylepšit implementací klasifikátorů a detektorů pro významy slov, aby se daly implementovat i logicky složitěji aplikovatelná pravidla, která mohou měnit původní významy slov. Dále by se práce dala vylepšit o normalizaci a lepší předzpracování dat – to bylo v rámci práce zanedbáno, neboť je zaručen normalizovaný vstup uživatele.