ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA APLIKOVANÝCH VĚD

Diskriminativní model pro porozumění mluvené řeči

Ing. Jan Švec

Disertační práce

k získání akademického titulu doktor v oboru Kybernetika

Školitel:
Prof. Ing. Josef Psutka, CSc.

Katedra:
Katedra kybernetiky

PLZEŇ, 2013

UNIVERSITY OF WEST BOHEMIA
FACULTY OF APPLIED SCIENCES

Discriminative model for spoken language understanding

Ing. Jan Švec

Dissertation thesis
submitted in partial fullfilment of the requirements
for the degree Doctor of Philosophy in the field of
Cybernetics

Advisor:
Prof. Ing. Josef Psutka, CSc.

Department:
Department of Cybernetics

PLZEŇ, 2013

Anotace

Předkládaná disertační práce je věnována problematice porozumění mluvené řeči. Práce prezentuje nový diskriminativní model určený pro tuto úlohu. Nejprve je popsána úloha porozumění řeči v kontextu hlasových dialogových systémů a jeho souvislost s rozpoznáváním řeči. Následuje přehled současného stavu řešené problematiky. Odstavce věnované tomuto tématu popisují jednak metody používané pro porozumění mluvené řeči, ale i metody z dalších oblastí zpracování řeči, které s prezentovaným modelem úzce souvisí. Dále jsou vytyčeny a odůvodněny cíle této disertační práce – především se jedná o vývoj nového diskriminativního modelu schopného zpracovat neurčitý vstup v podobě slovní nebo fonémové mřížky a následně vygenerovat více výstupních významových hypotéz. Jeden z podcílů je pak věnován výzkumu metody pro efektivní kombinaci znalostního a statistického přístupu k návrhu modulu porozumění. Porozumění mluvené řeči je dekomponováno do třech dílčích modelů – konceptového modelu, modelu detekce sémantických entit a modelu zarovnání. Zatímco konceptový model přiřazuje celé promluvě globální význam v podobě abstraktního sémantického stromu, model detekce sémantických entit označuje lokální dílčí významy pomocí jednotlivých sémantických entit. Následně model zarovnání provádí provázání těchto dvou dílčích významových reprezentací. Konceptový model je v této práci reprezentován hierarchickým diskriminativním modelem, který vznikl jako rozšíření existujícího statistického modelu založeného na klasifikátorech sémantických n-tic. Model detekce sémantických entit pak provádí hledání výskytů sémantických entit popsaných pomocí expertem definovaných bezkontextových gramatik. Po popisu těchto modelů následuje definice úlohy sestávající se z popisu dat použitých v experimentech a z popisu metodiky vyhodnocení. Součástí definice úlohy je i popis modelů a dekodéru pro automatické rozpoznávání řeči. Následuje experimentální ověření navržených modelů, přičemž jsou zdůvodněny konkrétní volby parametrů. Závěrečná kapitola shrnuje přínos navržené metody pro porozumění mluvené řeči. Rovněž popisuje splnění jednotlivých cílů disertační práce a předkládá další možné směry výzkumu navazující na tuto práci.

Klíčová slova: hlasové dialogové systémy; porozumění mluvené řeči; detekce sémantických entit; strojové učení; vážené konečné automaty

Annotation

The presented thesis is devoted to the spoken language understanding task. The thesis presents a new discriminative model for this task. First, the spoken language understanding is described in the context of spoken dialog systems and in relation to an automatic speech recognition. Then the state of the art is presented. The current methods for spoken language understanding are presented as well as methods related to the presented discriminative model. In the following chapter, the goals of the thesis are stated. The main goal is to develop a new discriminative model which is able to process uncertain input in the form of word-based or phoneme-based lattices and generate multiple output semantic hypotheses. One of the subgoals of this thesis is devoted to a research of method for effective combination of statistical and knowledge-based approaches to spoken language understanding. The spoken language understanding is decomposed into three partial models. A concept model assigns the global meaning of the utterance in the form of abstract semantic tree. A semantic entity detection tags the local parts of the meaning with the semantic entities. An alignment model links these two semantic representations and forms a discriminative spoken language understanding model. The concept model is represented by the hierarchical discriminative model which was developed as an extension of a statistical model based on semantic tuple classifiers. The semantic entity detection model performs the search for all occurrences of the semantic entities which are defined by knowledge-based context-free grammars. Then, the description of used data, recognition models, speech decoder, and evaluation methodology is presented. In the part devoted to experimental evaluation the values of specific parameters are selected and justified. The last chapter concludes the thesis and presents the overall performance of the presented method for spoken language understanding. It also describes the fulfilment of all goals of this thesis and presents the possible improvements and applications of the developed model.

Keywords: spoken dialog systems; spoken language understanding; semantic entity detection; machine learning; weighted finite state automata

Die Annotation

Die vorliegende Dissertation ist der Problematik des Verständnisses der gesprochenen Sprache gewidmet. Die Arbeit präsentiert ein neues diskriminatives für diese Aufgabe bestimmtes Model. Zuerst wird die Aufgabe des Verständnisses der Sprache im Kontext von Sprachdialogsysteme und sein Zusammenhang mit Erkennung der Sprache beschrieben. Dann folgt eine Übersicht des jetzigen Zustandes der zu lösenden Problematik. Die diesem Thema gewidmeten Absätze beschreiben einerseits die zum Verständnis der gesprochenen Sprache verwendeten Methoden, andererseits die Methoden aus weiteren Bereichen der Sprachverarbeitung, die mit dem präsentierten Model eng zusammenhängen. Weiter werden Ziele dieser Dissertation gestellt und begründet – vor allem handelt es sich um die Entwicklung eines neuen diskriminativen Models, das fähig ist, einen unbestimmten Eingang in Form eines Wort- oder Phonemgitters zu verarbeiten und anschließend mehr Ausgangs-Bedeutungshypothesen zu generieren. Eines der Subziele ist dann der Forschung der Methode für effektive Kombination des astatistischen Kenntnis-Zuganges zum Vorschlag des Verständnismoduls gewidmet. Das Verständnis der gesprochenen Sprache ist in drei Teilmodelle dekomponiert – in ein Konzeptmodel, in ein Model der Detektion von semantischen Entitäten und in ein Model der Ebnung. Während das Konzeptmodel der ganzen Ansprache eine globale Bedeutung in Form eines abstrakten semantischen Baumes zuordnet, bezeichnet das Model der Detektion von semantischen Entitäten lokale Teilbedeutungen mit Hilfe einzelner semantischen Entitäten. Anschließend führt das Model der Ebnung eine Verbindung dieser zwei Teil-Bedeutungsrepräsentationen durch. Das Konzeptmodel ist in dieser Arbeit durch ein hierarchisches diskriminatives Model repräsentiert, das als eine Erweiterung vom bestehenden statistischen Model entstand, gegründet auf Klassifikatoren von semantischen n-tupeln. Das Model der Detektion von semantischen Entitäten führt dann ein Aufsuchen von semantischen Entitäten durch, beschrieben mit Hilfe von definierten Grammatik ohne Kontext beschrieben durch einen Experten. Nach der Beschreibung dieser Modelle folgt eine Definition der Aufgabe, die aus einer Beschreibung der in den Versuchen verwendeten Daten und aus der Beschreibung der Methodik der Auswertung besteht. Ein Bestandteil der Definition der Aufgabe ist auch die Beschreibung der Modele und des Decoders für eine automatische Spracherkennung. Es folgt eine experimentale Beglaubigung von vorgeschlagenen Models, wobei konkrete Parameterwahlen begründet sind. Das Schlusskapitel fasst den Beitrag der vorgeschlagenen Methode zum Verständnis der gesprochenen Sprache zusammen. Ebenfalls beschreibt es die Erfüllung von einzelnen Zielen der Dissertation und liegt weitere mögliche Richtungen der Forschung vor, die an diese Arbeit anknüpfen.

Schlüsselwörter: Sprachdialogsysteme; Verständnis der gesprochenen Sprache; Detektion von semantischen Entitäten; Maschinenlernen

Prohlášení

Prohlašuji, že jsem předloženou disertační práci vypracoval samostatně, s použitím odborné literatury a pramenů, jejichž úplný seznam je její součástí.

V Plzni dne ....

podpis

Poděkování

Chtěl bych poděkovat mému školiteli, prof. Josefovi Psutkovi, za poskytnuté rady i odborné vedení v průběhu mého doktorského studia. Dále bych chtěl poděkovat Lubošovi Šmídlovi a Pavlovi Ircingovi za množství rad, podnětné diskuze a připomínky, které zcela jistě přispěly ke zkvalitnění této práce.

Největší díky si však zaslouží moje rodina – Terezka, Eliška a Kačenka – za veškerou trpělivost, lásku a podporu.

Disertační práce vznikla v rámci těchto výzkumných a vývojových projektů: projekt č. TE01020197 – *Centrum aplikované kybernetiky 3*, poskytovatel Technologická agentura České republiky, projekt č. FR-TI1/518 – *Inteligentní telefonní asistentka*, poskytovatel Ministerstvo průmyslu a obchodu.

Při řešení disertační práce byly využity výpočetní a úložné prostředky Národní gridové infrastruktury MetaCentrum financované z projektu č. LM2010005 – *Velká infrastruktura CESNET*, poskytovatel Ministerstvo školství, mládeže a tělovýchovy.

Obsah

Seznam obrázků

Seznam tabulek

Seznam symbolů a zkratek

Symbol	Popis (Poznámka)
$\mathcal{V}$, $\mathcal{T}$, $\mathcal{A}$, ...	Množina (kaligrafické písmo, velká písmena)
$\mathbf{o}$, $\mathbf{x}$, $\mathbf{w}$, ...	Vektor (tučný řez písma, malá písmena)
$\mathbf{Q}$, $\mathbf{M}$, $\mathbf{A}$, ...	Matice (tučný řez písma, velká písmena)
$\mathbf{A}^\intercal$	Transpozice matice $\mathbf{A}$
$\mathbf{x} \cdot \mathbf{w}$	Skalární součin vektorů $\mathbf{x}$ a $\mathbf{w}$
T, A, B, ...	Mimo jiné transducery a akceptory (základní řez písma, velká písmena)
O, U, T, ...	Náhodná proměnné (bezpatkové písmo, velká písmena)
$\mathrm{P}(A)$	Pravděpodobnost jevu A (symbol P je v základním řezu písma)
$\mathrm{P}(A\|B)$	Pravděpodobnost jevu A podmíněná jevem B
$\mathrm{E}(\mathsf{X})$	Střední hodnota náhodné proměnné X (symbol E je v základním řezu písma)
$\sigma^2(\mathsf{X})$	Rozptyl náhodné proměnné X
$\mathrm{cov}(\mathsf{X}, \mathsf{Y})$	Kovariance náhodných proměnných X a Y
$\mathbb{R}$	Reálná čísla
$\mathbb{N}$	Přirozená čísla
$\mathbb{K}$	Polookruh $\mathbb{K} = (\mathcal{K}, \oplus, \otimes, \bar{0}, \bar{1})$ (str. 32)
Θ	Množina sémantických konceptů (str. 73)
$\{\nu\}$	Speciální množina odp. chybějící lexikální realizaci (str. 75)
$\mathcal{V}$	Rozpoznávací slovník (množina slov)
$\mathcal{T}$	Trénovací množina (prvky jsou dvojice (*trénovací příklad*, *cílová třída*), nad prvky trénovací množiny existuje libovolné uspořádání pomocí indexů $i = 1, 2, \ldots l$)
$K(\cdot, \cdot)$	Jádrová funkce (str. 28)
$\mathrm{sgn}(x)$	Znaménková funkce (str. 26)
$\|a\|$	Absolutní hodnota a, pro řetězec a pak jeho délka
ϵ	Prázdný řetězec, u transducerů pak symbol se speciálním významem (str. 37)
$\mathcal{A}^*$	Kleeneho uzávěr množiny $\mathcal{A}$, množina $\mathcal{A}^*$ obsahuje všechny řetězce nad symboly z $\mathcal{A}$ včetně prázdného řetězce ϵ.
$\mathrm{cnt}(A, x)$	(Střední) počet výskytů jevu x ve struktuře A.
$W = (w_1, \ldots, w_N)$	Posloupnost W složená z prvků w_1, w_2 až w_N. Zkrácené zápisy: $W = w_1 w_2 \ldots w_N = (w_i)_{i=1}^N$

Symbol	Popis (Poznámka)
$\mathcal{A}$, $\mathcal{B}$	Vstupní, resp. výstupní abeceda transduceru (str. 32)
$\mathcal{Q}$	Množina stavů transduceru (str. 32)
$\mathcal{I}, \mathcal{F}$	Množina počátečních, resp. koncových stavů transduceru (str. 32)
$\mathcal{E}$	Množina přechodů transduceru (str. 32)
$\mathcal{P}$	Množina cest transducerem (str. 32)
ϵ, ρ, ϕ, σ	Symboly se speciálním významem (str. 37)
$\oplus$	Součet v daném polookruhu, sjednocení transducerů (str. 32, 34)
$\otimes$	Součin v daném polookruhu, konkatenace transducerů
T^*, T^+	Kleeneho uzávěr, Kleeneho plus transduceru T (str. 34)
T^{-1}	Inverze transduceru T (str. 35)
$T_1 \circ T_2$	Kompozice transducerů T_1 a T_2 (str. 35)
$\Pi_1(T)$, $\Pi_2(T)$	Projekce T na vstupní, resp. výstupní symboly (str. 35)
$\alpha[q]$, $\beta[q]$	Nejkratší vzdálenost z počátečních stavů do stavu q, resp. ze stavu q do koncových stavů transduceru (str. 35)
$F(T)$	Faktorový transducer nad transducerem T (str. 35)
rmeps, det, min	Optimalizační algoritmy nad transducery: odstranění ϵ-přechodů, determinizace, minimalizace (str. 36)
Acc, $Corr$, WER	Slovní přesnost, správnost, četnost chyb (str. 10)
$cAcc$, $cCorr$	Konceptová přesnost, správnost (str. 111)
BH	Bezplatné Hovory (řečový korpus)
HDM	Hierarchical Discriminative Model, hierarchický diskr. model
HHTT	Human-Human Train Timetable (řečový korpus)
HMM	Hidden Markov Models, skryté Markovské modely
HVS	Hidden Vector State (parser), parser se skryt. vektorovým stavem
NLP	Natural Language Processing, zpracování přirozeného jazyka
OOT	Out-of-topic, (věty) mimo téma
OOV	Out-of-vocabulary, (slova) mimo slovník
STC	Semantic Tuple Classifiers, klasifikátory sémantických n-tic
SVM	Support Vector Machines
TBL	Transformation Based Learning, učení založené na transformacích
TIA	Telefonní Inteligentní Asistentka (řečový korpus)

Kapitola 1

Úvod

Již v roce 1950 se Alan Turing ve své práci „Computing Machinery and Intelligence" [1, str. 442] zamýšlel nad otázkou, zda stroje mohou přemýšlet. Z této práce pochází i následující pasáž:

> *. . . at the end of the century the use of words and general educated opinion will have altered so much that one will be able to speak of machines thinking without expecting to be contradicted.*

> *. . . na konci století budou mluva a poučené obecné mínění pozměněné natolik, že bude možno mluvit o myslících strojích, aniž by se mluvčí musel obávat nesouhlasu.*

Jak je podotknuto v knize [2], i přes dlouhá desetiletí výzkumu v oblasti porozumění mluvené řeči není stále možné považovat současný stav za konečný. V reakci na výše zmíněný citát Alana Turinga její autor Gokhan Tür uvádí:

> *Yet, now we are well past the year 2000, and we wonder whether he meant the end of 21st century when machines will be able to „understand" us.*

> *Nyní, kdy je rok 2000 již delší dobu za námi, se musíme ptát, zda nebyl myšlen konec 21. století jako doba, kdy nám stroje budou schopny „porozumět".*

Hlavním tématem této práce je strojové porozumění mluvené řeči. Práce popisuje nový model pro porozumění řeči v omezené doméně za účelem použití v hlasových dialogových systémech. Výsledný model umožňuje efektivně kombinovat jak expertní znalost o dané úloze, tak znalosti (a modely) získané statistickými metodami z trénovacích dat. V průběhu řešení bylo nutné zkombinovat celou řadu různých přístupů.

Již v 50. letech 20. století se zformovala dvě paradigma pro zpracování přirozeného jazyka: přístup založený na automatech a přístup pravděpodobnostní [3]. První paradigma bylo reprezentováno osobnostmi jako Alan Turing (položil základy moderní informatiky), Warren McCulloch a Walter Pitts (popsali zjednodušený model neuronu – perceptron), Stephen Cole Kleene (objevitel regulárních výrazů) nebo Avram Noam Chomsky (např. popsal

Chomského hierarchii formálních jazyků). Druhé paradigma pak těžilo především z teorie zašuměného kanálu, se kterou přišel Claude Elwood Shannon. Touto teorií je proces porozumění řeči modelován jako přenos jejího významu pomocí akustického komunikačního kanálu.

Na konci 50. let a v průběhu 60. let 20. století se zpracování řeči a přirozeného jazyka velmi čistě rozdělilo do dvou směrů: symbolického a stochastického. K symbolickému přístupu přispěla jednak práce Chomského a dalších v oblasti teorie formálních jazyků a dále rozvoj symbolické umělé inteligence využívající modely usuzování a formální logiku. Symbolický přístup vedl k prvním modelům pro porozumění mluvené řeči [3]. Stochastický přístup používal Bayesovské metody pro modelování charakteristik pozorovaných jevů. Byl použit nejprve pro rozpoznávání textů (optical character recognition), v 70. letech však byl stochastický přístup použit pro rozpoznávání řeči. Modely pro automatické rozpoznávání řeči využívající skryté Markovské modely spolu s modelem zašuměného kanálu byly využívány především na pracovišti Thomas J. Watson Research Center firmy IBM. Tento přístup k rozpoznávání řeči je nerozlučně spjat se jménem českého emigranta Bedřicha Jelínka (v USA začal používat jméno Frederick Jelinek) a s jeho prací *Continuous speech recognition by statistical methods* [4]. Jeho přístup byl v tomto směru naprosto inovativní, neboť již v roce 1957 A. N. Chomsky napsal, že [5]:

> *We are forced to conclude that grammar is autonomous and independent of meaning, and that probabilistic models give no insight into the basic problems of syntactic structure.*
>
> *Jsme nuceni konstatovat, že gramatika je autonomní a nezávislá na významu a že pravděpodobnostní modely nedávají žádný náhled do základní problematiky syntaktických struktur.*

Přestože v 70. letech byl tento názor Chomského brán jako axiom komputační lingvistiky, F. Jelinek dokázal přijít se statistickým modelem, který je, i přes velkou snahu celé komunity vědců věnující se rozpoznávání řeči, stále nejrozšířenějším typem modelu [5].

V 80. letech 20. století se objevují aplikace konečně stavových modelů v oblasti zpracování přirozeného jazyka. Na základě úspěchů v rozpoznávání řeči jsou statistické metody používány i v dalších oblastech komputační lingvistiky.

V 90. letech 20. století a na počátku 21. století pak dochází k aplikaci teorie vážených konečných automatů v oblasti rozpoznávání a porozumění řeči. Jmenujme například práce, jejímiž autory jsou Mehryar Mohri, Fernando C. N. Pereirra a Michael Riley, popř. Cyril Allauzen [6, 7, 8]. Vážené konečné transducery působí jako jednotící prvek spojující symbolické a statistické paradigma. Například – znalosti v podobě regulárních nebo bezkontextových gramatik speciálního typu je možné převést do podoby váženého konečného transduceru. Vážené konečné transducery také umožňují reprezentovat generativní modely (typicky skryté Markovské modely) [9] a s využitím teorie racionální jádrových funkcí je lze použít i v modelech diskriminativních [10].

Modely pro porozumění řeči je možné rozdělit do tří kategorií [2]. Systémy náležející do první z těchto kategorií se ve skutečnosti o porozumění ani nesnaží, pouze jej předstírají. Typickým zástupcem této třídy je systém ELIZA [11], který pomocí jednoduchých transformačních pravidel aplikovaných na uživatelův vstup generoval svůj výstup. Druhá

kategorie vychází z teorie (symbolické) umělé inteligence. Tyto systémy jsou založeny na formální reprezentaci znalostí a na formální sémantické interpretaci. Provádějí mapování věty na její reprezentaci ve zvolené formální logice. Ukázalo se však, že tyto systémy jsou vhodné pouze pro velmi omezené domény.

Systémy spadající do třetí kategorie redukují porozumění řeči na problém zpracování přirozeného jazyka založený zpravidla na jeho statistickém zpracování. V současné době je snaha o úspěšné globální porozumění mluvené řeči otevřený problém, nicméně porozumění řeči v dané problémové oblasti (doméně) je řešitelné. Je však nutné poznamenat, že porozumění mluvené řeči není pouze jediná technologie, ale existuje celá řada přístupů vhodných pro konkrétní nasazení. Z důvodů chyb při automatickém rozpoznávání řeči není zpravidla možné použít obecný rozpoznávač řeči a jeho textový výstup použít v obecných algoritmech zpracování přirozeného jazyka. Je nutné brát v úvahu neurčitost vznikající při rozpoznávání řeči a tuto neurčitost převést i v neurčitost významových hypotéz.

Tato práce je členěna do jedenácti kapitol. Druhá kapitola *Hlasové dialogové systémy* (strana 4) zasazuje modul porozumění řeči do kontextu celého hlasového dialogového systému. Její část 2.2 – *Rozpoznávání řeči* (strana 7) poskytuje úvod do problematiky rozpoznávání řeči. V části 2.3 – *Porozumění mluvené řeči* (strana 10) je pak popsána úloha porozumění mluvené řeči v hlasových dialogových systémech. Kapitola 3 – *Přehled současného stavu řešené problematiky* (strana 13) shrnuje metody a práce v oboru zpracování a porozumění řeči. Tento souhrn popisuje i modely, které v byly použity jako referenční. Jsou zde shrnuty i další metody, které byly inspirací nebo které jsou analogické některým přístupům prezentovaným v této práci. Kapitola 4 – *Cíle disertační práce* (strana 21) vytyčuje cíle disertační práce. Další kapitola 5 – *Teoretický základ použitých metod* (strana 23) popisuje použité metody, jmenovitě teorii support vector machines (kapitola 5.1), teorii vážených konečných transducerů (kapitola 5.2), teorii racionálních jádrových funkcí (kapitola 5.3), základní popis stochastických bezkontextových gramatik (kapitola 5.4), úvod do n-gramových jazykových modelů (kapitola 5.5) a dále popis dvou referenčních modelů – parseru se skrytým vektorovým stavem (kapitola 5.6) a modelu využívajícího klasifikátorů sémantických n-tic (kapitola 5.7).

Následují kapitoly obsahují již vlastní přínos disertační práce pro oblast porozumění mluvené řeči. Kapitola 6 – *Diskriminativní model pro porozumění mluvené řeči* (strana 57) ustavuje základní pravděpodobnostní rámec diskriminativního modelu pro porozumění řeči. Kapitola 7 – *Hierarchický diskriminativní model* (strana 61) popisuje plně statistický diskriminativní model, který vstupní promluvě přiřazuje abstraktní sémantický strom.[1] Následující kapitola 8 – *Detekce sémantických entit* (strana 85) definuje přístup, který na základě dodané expertní znalosti vede k získání konkrétních hodnot sémantických entit přiřazených abstraktnímu sémantickému stromu. Fúzí informace z hierarchického diskriminativního modelu a detekce sémantických entit lze vstupní promluvě přiřadit částečně zarovnaný sémantický strom. Tato fúze je přitom realizována v rámci pravděpodobnostního modelu, přičemž je využito jak expertního, tak statistického přístupu k návrhu modelů. Kapitola 9 – *Definice úlohy* (strana 99) definuje dvě úlohy, nad nimiž je provedeno experimentální vyhodnocení popisovaného modelu. Součástí popisu úlohy je i popis způsobu vyhodnocení. Kapitola 10 – *Experimentální ověření* (strana 124) pak uvádí řadu experimentů v nichž byl ověřen navržený model a jeho části. Konečně kapitola 11 – *Závěr* (strana 151) uzavírá tuto disertační práci, vyhodnocuje její přínos a splnění jejích cílů.

[1] Pojem abstraktní sémantický strom je blíže definován v kapitole 2.3, strana 11.

Kapitola 2

Hlasové dialogové systémy

Z kybernetického úhlu pohledu můžeme na hlasový dialogový systém pohlížet jako na systém o dvou subsystémech, přičemž řeč tvoří komunikační prostředek mezi těmito dvěma subsystémy. Jeden z těchto subsystémů nazývejme *uživatel* a druhý *hlasový agent*. Uživatel, zpravidla člověk, užívá služeb hlasového agenta pro vykonání svých cílů a proto jej instruuje (řídí) tak, aby těchto cílů dosáhl. Na druhou stranu hlasový agent pro splnění cílů často musí řídit uživatele takovým způsobem, aby od něj získal požadované informace a mohl splnit jeho cíl.

Je nutné poznamenat, že jako hlasový dialogový systém nazýváme celou dvojici subsystémů. Odtud vyplývá zásadní požadavek na návrh hlasového dialogového systému – *nelze* navrhovat hlasového agenta bez znalosti (modelu) uživatele hlasového dialogu!

Obecně je při návrhu hlasového dialogového systému požadováno splnění následujících požadavků [12]:

- *Vysoká robustnost*, u naprosté většiny hlasových dialogových systémů se předpokládá použití více než jedním uživatelem v různorodých prostředích. Systém musí být schopen spolehlivě rozpoznat uživatelův vstup, zároveň musí být schopen zamítnout (nepřijmout) neodpovídající vstup tvořený šumem, řečí na pozadí nebo promluvami mimo doménu.
- *Dobrá srozumitelnost*, výstup systému musí být srozumitelný a jednoznačný. Zároveň uživatel komunikující s agentem musí mít možnost získání nápovědy, uživatel se nesmí v dialogu cítit „ztracen."
- *Rychlost odezvy*, při využití řeči jako běžného komunikačního prostředku nelze tolerovat zpoždění v komunikaci. Toto zpoždění způsobuje hlasový agent při rozpoznávání, porozumění, řízení dialogu a generování odpovědi. Toto zpoždění by nemělo být vyšší než řádově desetiny sekundy, v opačném případě dochází ke zbytečným kolizím mezi promluvou uživatele a odpovědí agenta a uživatel tuto interakci vnímá jako vysoce nepřirozenou.

Splněním těchto požadavků je nutnou podmínkou pro uživatelsky přívětivý hlasový dialog. Nejde však o podmínku postačující – lidé se zpravidla ostýchají komunikovat se strojem pomocí přirozené řeči, mají přehnaná očekávání, případně systém podceňují nebo jim komunikace připadá nepřirozená.

V současné době rovněž dochází k prudkému rozvoji mobilních technologií a bezdrátových datových sítí a často je jednodušší požadované informace získat prostřednictvím vizuálního rozhraní kombinovaného s dotykovým rozhraním než prostřednictvím hlasového dialogového systému.

Hlasová rozhraní však stále mají své místo v případech, kdy uživatel má zaměstnány oči jinou činností, například při řízení automobilu nebo v různých průmyslových a lékařských aplikacích. Zajímavou a slibně se rozvíjející oblastí jsou multimodální dialogové systémy, které mohou použít jak vizuální, tak řečovou interakci. Typ interakce se pak volí v závislosti na nejvhodnější a nejpřirozenější podobě vhodné pro danou informaci.

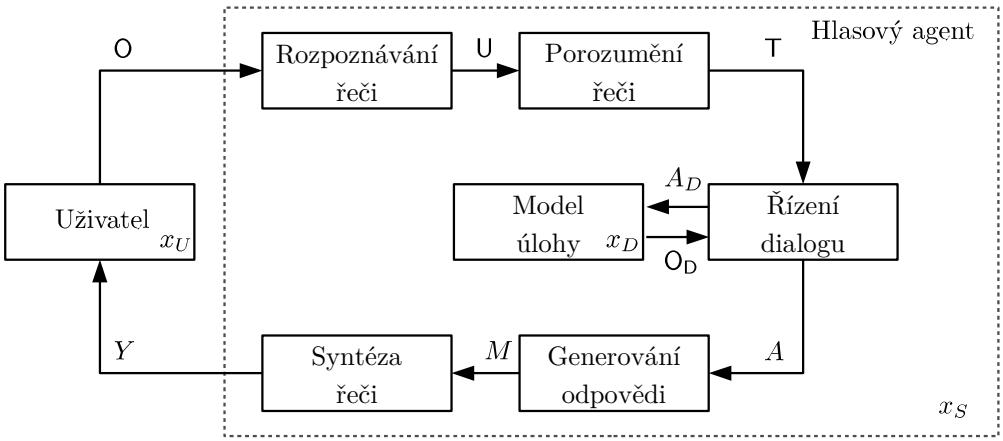
Hlasové dialogové systémy dále můžeme dělit na systémy s *iniciativou agenta* a se *smíšenou iniciativou*. Systémy s iniciativou agenta jsou takové systémy, kdy agent volí pořadí dotazů kladených uživateli a uživatel pouze pasivně odpovídá bez možnosti ovlivnit průběh hlasového dialogu. Oproti tomu systémy se smíšenou iniciativou nabízejí bohatší paletu interakcí – uživatel může nejprve agentovi předat všechny informace, které si myslí, že jsou užitečné pro splnění jeho cíle. Na agentovi pak je převzetí iniciativy, vyřešení případných nejednoznačností, chyb rozpoznávání a chybějících informací a následně prezentace hledané informace. Tímto však interakce nekončí a uživatel má možnost s výsledkem dále manipulovat, zužovat, případně rozšiřovat omezující podmínky, provádět další akce v kontextu výsledku a podobně [13, 14, 15, 16, 17].

Uveďme možný budoucí směr výzkumu a vývoje na poli hlasových dialogových systémů – jedná se o *inkrementální*, popřípadě o *spojité* hlasové dialogové systémy. Tyto dialogové systémy nepracují na úrovni jednotlivých promluv (turnů), ale průběžně rozpoznávají akustický signál od uživatele a v případě potřeby okamžitě reagují, přičemž není nutné čekat na konec promluvy uživatele, ale je možné provést „vboření" (barge-in) do jeho promluvy, přerušit jí a převzít iniciativu [18, 19, 20]. Přestože spojité hlasové dialogové systémy jsou v této práci zmíněny pouze tímto odstavcem, byly uvažovány při výzkumu zde uvedených metod a modelů.

2.1 Struktura hlasového dialogového systému

Pro účely modelování hlasového dialogového systému definujme *dialogový akt* (angl. *dialogue act*) jako základní jednotku hlasového dialogu. Celý dialog se skládá z výměny jednotlivých dialogových aktů mezi komunikujícími stranami. Dialogový akt jedné strany je zpravidla následován dialogovým aktem druhé strany a naopak, nicméně mohou existovat i dva a více zřetězených dialogových aktů jedné komunikující strany, navíc bez explicitního odlišení nebo oddělení jednotlivých dílčích dialogových aktů.

Model hlasového dialogu mezi uživatelem a agentem v nejjednodušší podobě je zobrazen na obrázku 2.1. V tomto modelu uvažujeme existenci následujících podsystémů – modulu *rozpoznávání řeči* a *porozumění řeči*, modulu *generování odpovědi* a *syntézy řeči*, modulu *řízení dialogu* a *modelu úlohy*. Zatímco rozpoznávání a porozumění řeči převádí uživatelův akustický řečový signál do strojové reprezentace dialogového aktu, generování odpovědi zpracovává výstupní dialogové akty agenta na akustický řečový signál. Modul řízení dialogu generuje na základě stavu úlohy a stavu agenta nový dialogový akt agenta. Také interaguje s modelem úlohy, odkud získává informace potřebné ke splnění cíle dialogu [16, 17].



Obrázek 2.1: Model hlasového dialogového systému.

Předpokládejme, že hlasový dialog je zahájen agentem prostřednictvím dialogového aktu A, který je v modulu *generování odpovědi* převeden na slovní realizaci M a v modulu *syntézy řeči* je vysyntetizovaná řeč Y odpovídající aktu A.

Uživatel pak na základě svého stavu x_U a signálu Y provede aktualizaci svého stavu a vygeneruje svojí promluvu o. Tato promluva je zpracována subsystémem *automatického rozpoznávání řeči* na rozpoznané jednotky u. Tyto jednotky jsou zpravidla tvořeny slovy, nicméně je možné uvažovat i o subslovních jednotkách jako jsou fonémy nebo slabiky popřípadě i o jiném způsobu reprezentace promluvy. Těmto jednotkám je následně v *modulu porozumění mluvené řeči* přiřazen významový popis t – dialogový akt uživatele.

Poznamenejme, že o, u a t jsou zatíženy neurčitostí, proto jsou zpravidla reprezentovány pomocí pravděpodobnostního rozdělení $\mathrm{P}(\mathsf{O} = o)$, $\mathrm{P}(\mathsf{U} = u|\mathsf{O} = o)$ a $\mathrm{P}(\mathsf{T} = t|\mathsf{U} = u, \mathsf{O} = o)$. Výstupem modulu porozumění řeči je pak pravděpodobnostní rozdělení $\mathrm{P}(\mathsf{T}|\mathsf{U}, \mathsf{O})$, se kterým pracuje modul řízení dialogu, který na základě $\mathrm{P}(\mathsf{T}|\mathsf{U}, \mathsf{O})$, svého stavu x_S a *strategie řízení* π vygeneruje dialogový akt agenta $A = \pi(x_S, \mathrm{P}(\mathsf{T}|\mathsf{U}, \mathsf{O}))$. Dialogový akt agenta je následně opět převeden na akustický signál a cyklus se opakuje.

V průběhu interakce s uživatelem si agent na základě pozorování O aktualizuje svůj stav x_S. Modul řízení dialogu nemusí nutně provádět interakci pouze s uživatelem, ale může také prostřednictvím akcí A_D řídit model úlohy a pozorovat výstupy tohoto modelu O_D. Stav modelu úlohy x_D je pozorován prostřednictvím náhodné proměnné O_D, přičemž model úlohy může, ale nutně nemusí být plně pozorovatelný. V praxi je často modelem úlohy velmi rozsáhlá databáze např. vlakových spojení a z důvodu enormního nárůstu počtu stavů je nepraktické tuto databázi zahrnovat přímo do hlasového agenta, ale je vhodnější vyčlenit ji jako model úlohy.

Pro dosažení vysoké robustnosti je nutné při návrhu hlasového dialogového systému zohlednit i chyby zavlečené do dialogu prostřednictvím jednotlivých modulů. Jedná se především o chyby rozpoznávání a porozumění řeči. Tyto chyby mají svůj dopad na stav x_S, přičemž musí být v možnostech hlasového dialogového systému (tj. uživatele ve spolupráci s hlasovým agentem) tyto chyby detekovat a prostřednictvím svých dialogových aktů chybu opravit – zotavit se z chyby nebo se navrátit k některé z předchozích hodnot stavu x_S. I přes maximální snahu v oblasti rozpoznávání řeči je stále chybovost na úrovni slov neza-

nedbatelná a tudíž subsystém porozumění řeči musí být vůči těmto chybám rozpoznávání velice odolný.

Deterministické systémy používají pouze nejlepší hypotézu o rozpoznaných jednotkách u a významu t a na základě této informace aktualizují svůj stav x_S a následně realizují deterministickou strategii $A = \pi(x_S, t, u)$. Tyto deterministické systémy jsou však podmnožinou výše uvedeného stochastického popisu a proto se jejich popisu nebudeme dále věnovat.

Klíčovým modulem hlasového dialogového systému je subsystém rozpoznávání řeči a subsystém porozumění řeči. Na přesnosti rozpoznávání a porozumění řeči je závislá efektivita celého hlasového dialogového systému. Přestože je možné jako výsledek procesu rozpoznávání a porozumění získat více hypotéz o téže promluvě, chyby vzniklé při tomto procesu je nutné napravit na úrovni modulu řízení dialogu pomocí interakce s uživatelem (zotavení z chyby). To však prodlužuje a zpomaluje samotný průběh dialogu. Poznamenejme, že chybný návrh rozpoznávání a nebo porozumění řeči může znemožnit zadání některých vstupů bez ohledu na modul řízení dialogu a tím velmi negativně ovlivnit celkovou použitelnost systému.

Tato práce je zaměřena na metody pro porozumění řeči, proto budou v následujících odstavcích popsány pouze moduly rozpoznávání a porozumění řeči. Detailní popis dalších modulů lze najít v literatuře [14, 16, 17, 21, 22, 23, 24, 25, 26, 27].

2.2 Rozpoznávání řeči

Rozpoznávání řeči je velmi složitou úlohou umělé inteligence. Počátky výzkumu rozpoznávání řeči lze najít v 50. a 60. letech 20. století [28]. Výzkum v 70. a 80. letech 20. století vedl již k současným metodám rozpoznávání řeči. Tyto metody jsou založené na statistickém přístupu, nejčastěji na skrytých Markovských modelech (Hidden Markov Models, HMM) [3, 4, 29, 30]. HMM jsou nejjednodušší dynamickou Bayesovskou sítí s množinou pozorovaných proměnných a množinou skrytých náhodných proměnných [31]. V úloze rozpoznávání řeči skryté proměnné (velmi zjednodušeně) odpovídají jednotlivým rozpoznávaným jednotkám, pozorované proměnné pak příznakovým vektorům generovaným z akustického signálu.

Systémy rozpoznávání řeči lze dělit na systémy na řečníku *závislé* a *nezávislé*. Pro účely hlasových dialogových systémů jsou používané téměř výhradně systémy na řečníku nezávislé, nicméně je možné provádět adaptaci na konkrétního řečníka a tím zvýšit přesnost rozpoznávání řeči.

Dále hlasové dialogové systémy téměř výhradně používají systémy rozpoznávání *souvislé řeči*. V těchto systémech se často uplatňují jevy související s dynamikou hlasového ústrojí člověka – koartikulace – řečová realizace sousedních slov se vzájemně ovlivňuje.

Navíc u systémů s iniciativou agenta uživatelé velmi často používají spontánní řeč. Ta se diametrálně liší od řeči čtené či diktované [32], neboť v proudu slov se častěji vyskytují různé neřečové události (nedořeky, přeřeky, váhání).

Rozpoznávání řeči využívající statistických metod lze formulovat jako úlohu *dekódování podle maximální aposteriorní pravděpodobnosti* [3, 30]. Nechť náhodná proměnná

$\mathsf{U} = \mathsf{W}_1\mathsf{W}_2 \ldots \mathsf{W}_N$ je tvořena posloupností N náhodných proměnných W_i reprezentujících jednotlivá slova nebo jiné jednotky promluvy.[1] Posloupnost $o = \{\mathbf{o}_1, \mathbf{o}_2, \ldots \mathbf{o}_T\}$ je posloupnost vektorů příznaků akustického řečového signálu. K tomu jsou použity metody akustické analýzy založené v současnosti nejčastěji na využití Melovských kepstrálních koeficientů (MFCC, [33]) či perceptivní lineární prediktivní analýzy (PLP, [34]).

Cílem rozpoznávání řeči je nalézt takovou posloupnost $\hat{u}$, která maximalizuje aposteriorní pravděpodobnost $\mathrm{P}(\mathsf{U} = u|\mathsf{O} = o)$:

$$\hat{u} = \arg\max_u \mathrm{P}(\mathsf{U} = u|\mathsf{O} = o) \tag{2.1}$$

Využitím Bayesova vztahu získáme následující rovnost (konkrétní hodnoty náhodných veličin jsou pro přehlednost vynechány):

$$\begin{aligned} \hat{u} = \arg\max_u \mathrm{P}(\mathsf{U}|\mathsf{O}) &= \arg\max_u \frac{\mathrm{P}(\mathsf{U})\mathrm{P}(\mathsf{O}|\mathsf{U})}{\mathrm{P}(\mathsf{O})} \\ &= \arg\max_u \mathrm{P}(\mathsf{U})\mathrm{P}(\mathsf{O}|\mathsf{U}) \end{aligned} \tag{2.2}$$

Tímto jsme aposteriorní pravděpodobnost nahradili součinem dvou modelů – *jazykového modelu* $\mathrm{P}(\mathsf{U})$ a *akustického modelu* $\mathrm{P}(\mathsf{O}|\mathsf{U})$. Tyto modely je možné trénovat nezávisle na sobě a každý z nich nese jinou část znalostí o řešené úloze. Akustický model $\mathrm{P}(\mathsf{O} = o|\mathsf{U} = u)$ vyjadřuje pravděpodobnost pozorování posloupnosti o při uvažování, že byla pronesena slova u. Jazykový model $\mathrm{P}(\mathsf{U} = u)$ pak vyčísluje apriorní pravděpodobnost výskytu posloupnosti u tvořené slovy nebo jinými jednotkami.

Nalezení nejpravděpodobnější posloupnosti slov $\hat{u}$ se provede aplikací vhodné prohledávací strategie. Pro snížení výpočetní náročnosti je často použito tzv. *Viterbiho aproximace* [30]. Algoritmus používající tuto aproximaci pak na základě známé posloupnosti o sestaví nejpravděpodobnější posloupnost stavů skrytého Markovského modelu a této posloupnosti stavů přiřadí posloupnost $\hat{u}$.

Jazykový model $\mathrm{P}(\mathsf{U} = u)$ určuje apriorní pravděpodobnost výskytu posloupnosti u *bez ohledu na vstupní posloupnost pozorování* o. Kvalitu jazykového modelu lze vyčíslit například hodnotou křížové entropie vypočítanou na datech nepoužitých pro trénování jazykového modelu [30]. Čím kvalitnější jazykový model, tím více přispívá k určení posloupnosti $\hat{u}$. V extrémním případě, kdy $\mathrm{P}(\mathsf{U} = \hat{u}) = 1$, tj. v případě, kdy jazykový model dopředu určuje promluvu $\hat{u}$, která bude výsledkem rozpoznávání, mluvíme o speciálním rozpoznávacím módu *forced-alignment*. V tomto módu používáme přímo slovní reprezentaci přiřazenou člověkem (anotátorem) dané promluvě. Stále však mluvíme o automatickém rozpoznávání řeči, neboť prohledávací strategie (2.2) přiřazuje nejpravděpodobnější posloupnost stavů skrytého Markovského modelu. Z této nejpravděpodobnější posloupnosti stavů pak lze odvodit časové indexy jednotlivých fonémů a slov stejně jako jejich akustickou pravděpodobnost $\mathrm{P}(\mathsf{O}|\mathsf{U})$.

Jak již bylo řečeno v kapitole 2.1, výstupem systému automatického rozpoznávání nemusí být pouze první nejlepší hypotéza, ale rozložení pravděpodobnosti $\mathrm{P}(\mathsf{U}|\mathsf{O})$. Efektivní

[1] Na rozdíl od literatury [3] nebo [30] je v této práci posloupnost rozpoznávaných jednotek značena jako U, nikoli W, neboť výstupem systému automatického rozpoznávání řeči nemusí nutně být slova – lze uvažovat i systémy rozpoznávající posloupnosti fonémů nebo slabik.

reprezentací tohoto rozložení je tzv. mřížka (angl. lattice) [35, 36]. Pro odlišení typu jednotek obsažených v mřížce budeme používat i spojení slovní mřížka nebo fonémová mřížka. Mřížka je acyklický graf, který reprezentuje různé hypotézy u odpovídající vstupní posloupnosti o a přiřazuje jim pravděpodobnost $\mathrm{P}(\mathsf{U} = u|\mathsf{O} = o)$. Velmi častou reprezentací mřížek jsou vážené konečné akceptory [6] (použita notace z kapitoly 5.2). Uvažujme, že pravděpodobnostnímu rozložení $\mathrm{P}(\mathsf{U}|\mathsf{O})$ odpovídá vážený konečný akceptor U nad pravděpodobnostním polookruhem. Potom $\oplus$-suma vah všech cest π z počátečního stavu U do některého z koncových stavů U a se vstupními symboly u (tj. $i[\pi] = u$) odpovídá pravděpodobnosti $\mathrm{P}(\mathsf{U} = u|\mathsf{O} = o)$:

$$\mathrm{P}(\mathsf{U} = u|\mathsf{O} = o) = \bigoplus_{\pi \in U : i[\pi] = u} w[\pi] \tag{2.3}$$

Posloupnosti u jsou nejčastěji tvořeny slovy, nicméně lze použít i další subslovní jednotky jako slabiky [37] nebo fonémy [38], popř. jejich kombinace [39]. Tyto jednotky jsou vybírány z množiny jednotek (slov, fonémů, slabik) $\mathcal{V}$ nazývané rozpoznávací slovník. Systém automatického rozpoznávání řeči je schopen rozpoznat pouze jednotky z rozpoznávacího slovníku. Pokud řečník použije v promluvě slovo, které nenáleží do používaného rozpoznávacího slovníku – tzv. *slovo mimo slovník* (angl. *out-of-vocabulary word*, *OOV word*) – pak toto slovo nemůže být nikdy správně rozpoznáno. Navíc díky vlivu jazykového modelu, který modeluje pravděpodobnost slova v závislosti na jeho kontextu, se tato chyba může rozšířit i na okolní slova. Proto je nutné při návrhu hlasových dialogových systémů věnovat pozornost právě rozpoznávacímu slovníku. Ten je možné odhadnout z trénovacích dat a je možné jej dále doplnit o slova z doménově závislých databází [40]. Pro detekci slov mimo slovník je možné rovněž použít *míry důvěry* (angl. *confidence measure*) [41]. Míra důvěry vyjadřuje aposteriorní pravděpodobnost $P(\mathsf{U}_{ij} = w|\mathsf{O} = o_i, \ldots, o_j)$, že v daném úseku vyjádřeného indexy $i, \ldots j$ se vyskytlo slovo w. Nicméně i tyto míry důvěry jsou pouze odhady a OOV slova nemusí spolehlivě detekovat.

Četnost slov mimo slovník se běžně pohybuje v řádech jednotek procent [42]. Systémy založené na slabikách již mají velikost slovníku nižší, čímž klesá i pravděpodobnost výskytu slabiky mimo slovník. A konečně fonémové rozpoznávače pracují s konečnou množinou fónů, která odpovídá akustickému modelu a pro konkrétní jazyk ji lze zpravidla předem sestavit.

Pro vyhodnocení přesnosti systémů automatického rozpoznávání řeči se používá postupu, kdy rozpoznaná promluva (hypotéza) je nejprve *zarovnána* s referenční transkripcí vytvořenou anotátorem. Pro zarovnání se používá zpravidla algoritmu pro výpočet Levenshteinovy vzdálenosti [43], přičemž průchodem posloupností editačních operací, která vede na minimální editační vzdálenost, jsou získána následující čísla:

- H – počet správně rozpoznaných slov
- S – počet slov, která jsou chybně rozpoznaná jako jiná slova
- D – počet slov chybějících v rozpoznané hypotéze
- I – počet slov přebývajících v rozpoznané hypotéze
- N – počet slov v referenční transkripci

Potom lze definovat následující míry: *přesnost* (accuracy, *Acc*), *správnost* (correctness, *Corr*) a *četnost slovních chyb* (word error rate, *WER*):

$$Acc = \frac{N - D - S - I}{N} \tag{2.4}$$

$$Corr = \frac{H}{N} \tag{2.5}$$

$$WER = 1 - Acc = \frac{D + S + I}{N} \tag{2.6}$$

Tyto míry budou použity v experimentální části disertační práce pro vyhodnocení přesnosti subsystému automatického rozpoznávání řeči. Na těchto mírách je rovněž založeno odvození měr pro vyhodnocení přesnosti systému porozumění mluvené řeči (kapitola 9.3, strana 107).

2.3 Porozumění mluvené řeči

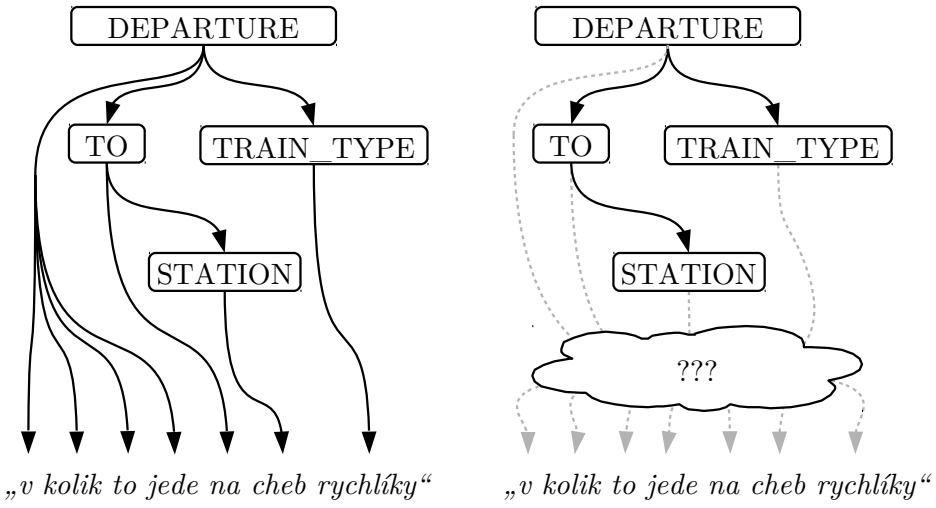
Cílem porozumění mluvené řeči je na základě pravděpodobnostního rozdělení $P(\mathsf{U}|\mathsf{O})$ sestavit pravděpodobnostní rozdělení $P(\mathsf{T}|\mathsf{U},\mathsf{O})$. V praxi je často uvažována aproximace, která předpokládá, že rozpoznané jednotky U obsahují veškerou informaci o významu T a tudíž $P(\mathsf{T}|\mathsf{U},\mathsf{O}) \approx P(\mathsf{T}|\mathsf{U})$.

Konkrétní hodnoty t náhodné proměnné T mohou mít různou strukturu. Nejčastěji se jedná o seznam párů *atribut:hodnota* [44] nebo o *sémantické stromy*. Uzly sémantických stromů, případně atributy v seznamu párů jsou tvořeny *sémantickými koncepty*. Samotný pojem *koncept* přibližuje následující definice:

> *[Concept is] an idea or mental image which corresponds to some distinct entity or class of entities, or to its essential features, or determines the application of a term (especially a predicate), and thus plays a part in the use of reason or language.*

> *[Koncept je] představa nebo mentální obraz, který odpovídá nějaké jedinečné entitě nebo třídě entit nebo jejím základním vlastnostem, popřípadě určuje způsob použití jednotlivých termínů (především predikátů) a tudíž hraje roli v uvažování nebo v jazyce jako takovém.*
>
> [The New Oxford Dictionary of English]

V této práci budeme sémantickými koncepty nazývat značky (tagy), které odlišují různé významy promluv a různé třídy entit v rámci jednotlivých promluv. Sémantické koncepty budeme vždy považovat za doménově závislé, množina sémantických konceptů bude jiná pro úlohu inteligentní asistentky a jiná pro úlohu navigačního software. Některé sémantické koncepty mohou mít přiřazenu konkrétní hodnotu – *sémantickou entitu*. U dalších sémantických konceptů pak konkrétní lexikální realizace je nepodstatná, pro samotný význam promluvy je důležitá pouze jejich přítomnost nebo nepřítomnost.



Obrázek 2.2: Zarovnaný sémantický strom (vlevo) a abstraktní sémantický strom (vpravo). Často je používán i linearizovaný zápis abstraktního sémantického stromu ve tvaru: DEPARTURE(TO(STATION), TRAIN_TYPE). Obdobně linearizovaný zápis zarovnaného sémantického stromu je ve tvaru: DEPARTURE(*v kolik to jede* TO(*na* STATION(*cheb*)) TRAIN_TYPE(*rychlíky*)).

Sémantická entita odpovídá konkrétní lexikální (slovní) realizaci v dané promluvě. Sémantická entita je reprezentována svým typem a interpretací (kapitola 8, str. 85). Typem sémantické entity může být například datum, čas, jméno apod. Interpretace dále doplňuje sémantickou entitu o konkrétní hodnoty a slouží jako obraz reálného objektu v rámci modelu hlasového agenta.

Pro reprezentaci významu t budeme používat strukturu *sémantického stromu*. Sémantický strom vyjadřuje hierarchickou závislost mezi jednotlivými sémantickými koncepty a slovy vstupní promluvy. Budeme proto používat i ekvivalentní termín *zarovnaný sémantický strom*. Sémantické koncepty blíže kořenu sémantického stromu jsou zpravidla obecnější, sémantické koncepty dále od kořenu pak specifičtější. V listech zarovnaného sémantického stromu jsou uložena slova – lexikální realizace jednotlivých konceptů.

Dále budeme používat i termín *abstraktní sémantický strom* (popř. nezarovnaný sémantický strom). Abstraktní sémantické stromy popisují pouze množinu a strukturu sémantických konceptů přiřazených dané promluvě bez vazby na sémantické entity nebo na slova promluvy. Budeme též říkat, že abstraktní sémantické stromy jsou nezarovnané s původní promluvou, konceptům na základě abstraktního sémantického stromu nelze přiřadit konkrétní hodnoty [45, 46, 47]. Příklad sémantického stromu a abstraktního sémantického stromu je uveden na obrázku 2.2.

Na tomto obrázku jsou vyobrazeny sémantické stromy sestavené ze sémantických konceptů DEPARTURE, TO, TRAIN_TYPE a STATION. Z pohledu hlasových dialogových systémů můžeme koncepty DEPARTURE a TO uvažovat jako koncepty, u nichž nezáleží na jejich lexikální realizaci, pro řízení dialogu je významná pouze jejich přítomnost v sémantickém stromu. Oproti tomu koncepty TRAIN_TYPE a STATION se pojí se sémantickými entitami *train_type:R* a *station:cheb* reprezentující objekt „rychlík” a objekt „stanice Cheb”.

Odtud vyplývá potřeba definovat poslední z používaných termínů – *částečně zarovnaný sémantický strom*. V těchto stromech je pouze některým sémantickým konceptům přiřazena konkrétní lexikální realizace nebo konkrétní část vstupní promluvy. Použijeme-li příklad stromu na obrázku 2.2, můžeme částečně zarovnaný sémantický strom zapsat jako DEPARTURE(TO(STATION(*cheb*)), TRAIN_TYPE(*rychlíky*)). Zde je sémantický strom zarovnán pouze se slovy *cheb* a *rychlíky*. Poznamenejme, že částečně zarovnaný sémantický strom nemusí mít ve svých listech uloženy lexikální realizace, ale například sémantické entity získané ze vstupní promluvy.

V případě, kdy nebude nutné detailně rozlišovat mezi zarovnanými, nezarovnanými (abstraktními) a částečně zarovnanými sémantickými stromy, budeme používat jediný společný termín „sémantický strom" a až v případě nutnosti budeme tyto případy rozlišovat vhodným přídavným jménem.

Na rozdíl od metod rozpoznávání řeči, kde jsou de-facto standardem skryté Markovské modely, je na poli porozumění řeči používáno velké množství různých metod. V kapitole 3 jsou popsány různé přístupy se zaměřením především na statistické metody.

Obecně můžeme rozlišovat dva typy statistických modelů – *generativní* a *diskriminativní*. Mějme pozorovanou náhodnou proměnnou U a na základě pozorované hodnoty u chceme odhadnout pravděpodobnostní rozdělení $\mathrm{P}(\mathsf{T} = t|\mathsf{U} = u)$ skryté (nepozorované) náhodné proměnné T.

Generativní modely modelují sdruženou pravděpodobnost $\mathrm{P}(\mathsf{U}, \mathsf{T})$, odhad $\mathrm{P}(\mathsf{T}|\mathsf{U})$ pak vyplývá z Bayesova vztahu:

$$\mathrm{P}(\mathsf{T}|\mathsf{U}) = \frac{\mathrm{P}(\mathsf{U}|\mathsf{T})\mathrm{P}(\mathsf{T})}{\mathrm{P}(\mathsf{U})} \propto \mathrm{P}(\mathsf{U}|\mathsf{T})\mathrm{P}(\mathsf{T}) = \mathrm{P}(\mathsf{U}, \mathsf{T}) \tag{2.7}$$

Zpravidla se nemodeluje přímo sdružená pravděpodobnost $\mathrm{P}(\mathsf{U}, \mathsf{T})$, ale dílčí pravděpodobnosti $\mathrm{P}(\mathsf{U}|\mathsf{T})$ a $\mathrm{P}(\mathsf{T})$. Zde $\mathrm{P}(\mathsf{T})$ vyjadřuje apriorní pravděpodobnostní rozdělení nad hodnotami skryté proměnné T a $\mathrm{P}(\mathsf{U}|\mathsf{T})$ pravděpodobnost generování (pozorování) hodnoty $\mathsf{U} = u$, pokud je $\mathsf{T} = t$. Odtud vyplývá i původ názvu generativní modely.

Oproti tomu diskriminativní modely modelují přímo pravděpodobnostní distribuci $\mathrm{P}(\mathsf{T}|\mathsf{U})$, nikoli sdruženou pravděpodobnost. Diskriminativní modely jsou vhodné téměř výhradně pro učení s učitelem, modifikace těchto modelů pro učení bez učitele není zpravidla tak jednoduchá, jako v případě generativních modelů. Mezi diskriminativní modely patří například klasifikátory založené na support vector machines nebo na umělých neuronových sítích.

Kapitola 3

Přehled současného stavu řešené problematiky

Tato kapitola je věnována přehledu metod a obecně prací v oboru zpracování řeči a přirozeného jazyka za účelem porozumění řeči. Nejde o vyčerpávající přehled, záměrem je přinést stručný popis metod a technologií příbuzných s těmi využívanými a vyvinutými v rámci této práce:

- *Stochastické bezkontextové gramatiky* představují velice často používaný formalismus pro zachycení expertní znalosti v oblasti automatického rozpoznávání a porozumění řeči. Formalismus stochastických bezkontextových gramatik je použit jako inspirace pro *sémantické gramatiky* popisované v kapitole 7.3, jež slouží pro reprezentaci pravděpodobností různých realizací daného významového konceptu ve zpracovávané promluvě.
- *Parser se skrytým vektorovým stavem* reprezentuje jeden z referenčních modelů, vůči kterému je porovnáván přínos hierarchického diskriminativního modelu dále popisovaného v následujících kapitolách.
- *Klasifikátory sémantických n-tic* jsou druhým z referenčních modelů, zástupce třídy diskriminativních modelů využívající SVM klasifikátoru pro predikci sémantického stromu.
- *Transformation-based learning* je další z možných metod vedoucích na diskriminativní model. Tato metoda je používaná především v úlohách zpracování přirozeného jazyka, nicméně v odstavci věnovaném tomuto způsobu trénování je zmíněna i aplikace na problém porozumění mluvené řeči.
- *Detekce klíčových slov pomocí hierarchických klasifikátorů* spočívá v dekompozici úlohy detekce klíčových slov do dvou vrstev diskriminativních klasifikátorů. Práce je zmíněna především kvůli analogii s vyvinutým hierarchickým diskriminativním modelem pro porozumění řeči.
- *Detekce klíčových slov pomocí vážených konečných transducerů* je inovativní přístup k indexaci slovních mřížek využívající tzv. faktorový automat. Přístup faktorového automatu byl použit k efektivnímu výpočtu racionální jádrové funkce a tím k dosažení vysoké rychlosti predikce významu.

3.1 Stochastické bezkontextové gramatiky

Používání stochastických bezkontextových gramatik v oblasti automatického rozpoznávání řeči je možné dohledat do úplných prvopočátků tohoto vědního oboru. Reprezentace jazykové a posléze i sémantické znalosti pomocí expertně navržených gramatik je relativně přímá a vhodná pro rychlé prototypování hlasových dialogových systémů [3, 48, 49]. Schopnost přehledně reprezentovat jazyk řešené úlohy a zároveň možnost návrhu jednoduchého modelu porozumění vedla i ke standardizaci těchto gramatik pro použití v řečových technologií konsorciem W3C v rámci standardů VoiceXML [50] a Speech Recognition Grammar Specification (SRGS) [51].

Nevýhody stochastických bezkontextových gramatik spočívají především v příliš striktních požadavcích na vstup. Předpokládá se, že jazyk generovaný gramatikami plně odpovídá možným promluvám. Přestože uživatelé hlasového dialogového systému mohou být vedeni dobře navrženými výzvami k používání vhodné struktury promluv, nikdy není možné vyloučit jevy běžné ve spontánních promluvách, jako např. váhání, přeřeky, opakování slov apod. Modelování těchto jevů pomocí gramatik je sice možné [52], nicméně jejich plné pokrytí přináší zvýšené nároky při vývoji a následně při strojovém zpracování vstupu. Proto se používají i přístupy, které modelují jazykovou pravděpodobnost $\mathrm{P}(\mathsf{U})$ pomocí n-gramového jazykového modelu, nicméně z výstupní posloupnosti slov získávají sémantickou informaci pomocí různě modifikovaných stochastických bezkontextových gramatik [53, 54]. Bezkontextové gramatiky jsou použity i pro modelování sémantických entit a pro jejich sémantickou interpretaci – více v kapitole 8.

3.2 Parser se skrytým vektorovým stavem

Parser se skrytým vektorovým stavem (Hidden Vector State parser, HVS parser) [55] je jedním z generativních modelů určených pro porozumění řeči. Tento model modeluje sdruženou pravděpodobnost pozorování posloupnosti slov U a posloupnosti sémantických konceptů C, přičemž pro získání nejpravděpodobnější posloupnosti sémantických konceptů $\hat{C}$ se aplikuje:

$$\hat{C} = \arg\max_{C} \mathrm{P}(\mathsf{U}, \mathsf{C}) = \arg\max_{C} \mathrm{P}(\mathsf{U}|\mathsf{C})\mathrm{P}(\mathsf{C}) \tag{3.1}$$

kde $\mathrm{P}(\mathsf{U}|\mathsf{C})$ je nazýváno *lexikálním modelem* a $\mathrm{P}(\mathsf{C})$ *sémantickým modelem*. V případě, že posloupnost skrytých stavových proměnných C je tvořena pouze jednoduchými, nestrukturovanými sémantickými koncepty, je struktura modelu ekvivalentní konečně stavovému taggeru. Na tento tagger lze rovněž nahlížet jako na skrytý Markovský model s pozorováními U a skrytou proměnnou C. Tento jednoduchý model byl použit například v systému Chronus [44] pro úlohu ATIS [56].

Parser se skrytým vektorovým stavem pak rozšiřuje výše uvedený konečně stavový tagger tak, že do posloupnosti C přidává interní strukturu – prvky posloupnosti C jsou vektory náhodných proměnných $\mathsf{C} = [\mathsf{C}_t]_{t=1}^{N}$ a $\mathsf{C}_t = [\mathsf{C}_t[1], \mathsf{C}_t[2], \ldots, \mathsf{C}_t[m_t]]$. Náhodná proměnná C_t pak odpovídá stavům zásobníkového automatu pro slovo U_t a skládá se z jednotlivých konceptů $\mathsf{C}_t[i]$.

Sémantický model $\mathrm{P}(\mathsf{C})$ pak modeluje přechody mezi jednotlivými stavy zásobníkového automatu pomocí zásobníkových operací *push* (uložení nového konceptu na zásobník) a *pop*

(odstranění konceptu ze zásobníku). Tento model navrhl Yulan He a Steve Young v roce 2003 v práci [55]. Nový model experimentálně ověřili na úloze ATIS [56], přičemž prokázali, že HVS model dosahuje lepších výsledků v porovnání s konečně stavovým taggerem.

V roce 2007 provedl Filip Jurčíček [46] rozšíření původního HVS parseru o možnost generování sémantických stromů s levo-pravým větvením. Bylo ukázáno, že rozšířením struktury modelu o další stavové proměnné explicitně modelující operace *push* a *pop* vede na statistický významné zvýšení přesnosti modelu na korpusu HHTT [57].

Autor této práce provedl v roce 2007 rozšíření HVS parseru o možnost zpracování vstupu reprezentovaného posloupností příznakových vektorů, přičemž příznakový vektor byl složen z původního slova a lingvistických příznaků – lemmatu a morfologické značky [58].

Mezi zásadní výhody HVS parseru (a rovněž konečně stavového taggeru) patří možnost trénovat model z abstraktních sémantických anotací, tj. anotací, které neobsahují zarovnání sémantických konceptů a slov vstupní promluvy. Při trénování je použita varianta Expectation-Maximization algoritmu. V E-kroku algoritmu je provedeno zarovnání sémantické anotace pomocí existujícího modelu a následně v M-kroku dojde k reestimaci parametrů na základě zarovnání získaného v E-kroku. Další vlastností vhodnou pro nasazení v reálných hlasových dialogových systémech je možnost převedení HVS parseru do podoby váženého konečného transduceru a následně využití optimalizačních metod definovaných nad těmito automaty. Reprezentace pomocí váženého konečného transduceru rovněž umožňuje použití výstupu rozpoznávače v podobě slovní mřížky.

Nevýhody HVS modelu pak plynou především z generativní podstaty modelu. Obtížný je především odhad parametrů lexikálního modelu $P(\mathsf{U}|\mathsf{C})$ pro málo četná slova a málo četné stavy zásobníkového automatu C_t.

Detailnější popis HVS parseru je uveden v kapitole 5.6, str. 51.

3.3 Klasifikátory sémantických n-tic

V práci [59] z roku 2009 François Mairesse a kol. popisují inovativní model porozumění využívající množinu diskriminativních klasifikátorů. Tento model nazývají jako *klasifikátory sémantických n-tic* (Semantic Tuple Classifiers, STC). Při trénování tohoto modelu je každý sémantický strom dekomponován do množiny sémantických n-tic (příklad dekompozice uveden v kapitole 5.7, str. 53). Pro každou z těchto n-tic je natrénován diskriminativní model – binární klasifikátor – predikující přítomnost dané n-tice v sémantickém stromu odpovídajícím vstupní, neznámé promluvě. Příznakový vektor klasifikátorů využívá lexikosyntaktické příznaky generované ze vstupní promluvy (např. četnosti všech n-gramů délky 1 až 3). Autoři použili klasifikátory založené na SVM, čímž se vyhnuli problémům s vysokou dimenzí vstupního příznakového vektoru.

Při přiřazení významu neznámé vstupní promluvě jsou vyčísleny predikce jednotlivých klasifikátorů a následně je sestavena množina odpovídajících sémantických n-tic z nichž je rekonstruován výsledný sémantický strom. Autoři v práci rovněž využívají znalosti získané z databáze dané úlohy – posloupnosti slov ve vstupní promluvě, které odpovídají některé z databázových položek, jsou nahrazeny jednotným obecným identifikátorem třídy. Tento přístup obecně vede k robustnějšímu modelu jednak díky lepší schopnosti zobecňovat a také díky menší dimenzi příznakového vektoru. Vyšší zobecňovací schopnost je odůvodněna

možností zpracovávat i neviděné posloupnosti slov, které však odpovídají některým položkám z existující databáze. Redukce dimenze příznakového vektoru pak vyplývá z náhrady mnoha různých slov jediným identifikátorem třídy.

Autoři prezentují vyšší přesnost predikce sémantických stromů v porovnání s HVS parserem. Na úloze ATIC STC model dosahuje přesnosti srovnatelné s mnohem komplexnějšími modely, např. Probabilistic Combinatory Categorial Grammars (PCCG) [60].

Mezi výhody tohoto přístupu opět patří možnost trénování z nezarovnaných dat. Dále pak využití diskriminativních klasifikátorů pro predikci sémantických stromů, čímž se lze vyvarovat problémům spojeným s generativními modely, především týkající se řídkosti vstupních dat (více na str. 57). STC model rovněž umožňuje využití expertních a databázových znalostí k předzpracování vstupní promluvy a zvýšení robustnosti.

Nevýhody tohoto přístupu spočívají především v nutnosti získávání vektoru příznaků o relativně vysoké dimenzi. To může být jistou komplikací v okamžiku, kdy pro trénování jsou použity celé slovní mřížky namísto první nejlepší slovní hypotézy. STC model rovněž předpokládá, že výskyty jednotlivých sémantických n-tic jsou nezávislé jevy, neboť tyto jevy modeluje nezávislými klasifikátory. Ve skutečnosti jsou však výskyty různých sémantických n-tic vzájemně korelované. Navíc rekonstrukce sémantického stromu z výsledných predikcí jednotlivých klasifikátorů je založena na heuristice. Díky tomu původní STC model neumožňuje získání více sémantických hypotéz spolu s aposteriorními pravděpodobnostmi. V práci [61] autoři zmiňují modifikaci tohoto modelu za účelem získání n-nejlepších hypotéz, nicméně i v této modifikaci je opět uvažován předpoklad nezávislosti klasifikátorů. Tyto nedostatky jsou jednou z motivací pro vývoj hierarchického diskriminativního modelu popsaného v kapitole 7.

3.4 Transformation-based learning

Průkopnickou prací na poli Transformation-based learning (TBL) je výzkum Erica Brilla. TBL je přístup k trénování klasifikátorů založený na odvozování uspořádané množiny pravidel pomocí iterativní korekce chyb nad trénovací množinou. Prvotní aplikace TBL byly z oblasti zpracování přirozeného jazyka (Natural Language Processing, NLP). Mezi tyto aplikace TBL patří morfologické značkování (POS tagging) [62] a syntaktická analýza větné struktury [63]. V následujících pracích pak byla metoda TBL použita nejen v oblasti NLP, ale i v oblasti zpracování řeči, například pro klasifikaci dialogových aktů [64]. V práci [65] je TBL přístup použit i pro porozumění mluvené řeči v hlasových dialogových systémech.

TBL využívá trénovací korpus manuálně označených dat. Pro trénování je nutné sestavit sadu šablon pro generování množiny možných transformačních pravidel. Rovněž je nutné určit pravidlo pro vygenerování počátečního výstupu (např. počáteční posloupnosti POS značek, počátečního sémantického stromu, atd.). V procesu trénování je dále použita kriteriální funkce pro hodnocení přínosu konkrétních pravidel. Šablony transformačních pravidel jsou obecně ve tvaru:

$$\text{změň výstup z } a \text{ na } b \;/\; \text{platí-li } \textit{podmínka}$$

přičemž *podmínka* může být buď *nelexikalizovaná* – potom dotazy vyskytující se v *podmínce*

se týkají pouze výstupní struktury (POS tagů, stromů atd.), nebo *lexikalizovaná* – pak se v podmínce vyskytují i slova vstupní promluvy. Příklady pravidel:

- Změn POS značku aktuálního slova z t_i na t_j, je-li předchozí slovo označeno značkou t_a (nelexikalizované).
- Změn POS značku aktuálního slova na t_i, je-li aktuální slovo w_a a předchozí slovo w_b (lexikalizované).
- Změn dialogový akt na $REJECT$, je-li aktuální slovo rovno „*ne*" (lexikalizované).
- Přidej do výstupu slot *to.station* s hodnotou „*Praha*", je-li v promluvě bigram „*do Prahy*" (lexikalizované).

Algoritmus trénování pak při dané trénovací množině a počáteční transformaci postupuje v následujících krocích:

1. Vygeneruj všechna pravidla, která opravují alespoň jednu chybu v predikci.
2. Pro všechna pravidla:
   - Aplikuj pravidlo na kopii trénovací množiny.
   - Ohodnoť výsledné predikce pomocí kriteriální funkce.
3. Vyber pravidlo s nejlepší hodnotou kriteriální funkce, přidej ho na konec uspořádané množiny pravidel a data, na která bylo aplikováno toto pravidlo použij jako novou trénovací množinu.
4. Zastav, je-li hodnota kriteriální funkce menší než zadaný práh. Jinak pokračuj krokem 1.

Při dekódování daného vstupu je pak použit algoritmus:

1. Aplikuj počáteční transformaci shodnou s trénovací fází.
2. Postupně aplikuj pravidla z uspořádané množiny natrénovaných pravidel.
3. Po aplikaci všech pravidel vrať výslednou predikovanou strukturu (posloupnost, strom atd., dle úlohy).

Mezi výhody TBL přístupu patří především možnost podchycení delšího kontextu než v případě použití skrytých Markovských modelů. Trénování rovněž probíhá nad celou trénovací množinou, přičemž je optimalizována přímo daná kriteriální funkce, která může být takřka libovolná. Navíc výstupem může být libovolná struktura, například strom, množina párů atribut-hodnota apod. Výpočetní náročnost samotného dekódovacího algoritmu je velmi nízká, neboť se jedná o aplikaci konečného počtu pravidel.

Mezi nevýhody TBL lze zařadit především neoptimální trénovací proces. Algoritmus je tzv. hladový (angl. greedy) – nejlepší nalezené pravidlo je přidáno do množiny pravidel, přestože kombinace například dvou jiných pravidel může v dané iteraci trénovacího algoritmu lépe optimalizovat kriteriální funkci. Další problematickou otázkou při použití TBL

je zpracování neznámých slov, tj. slov, které se nevyskytly v trénovací množině, ale mohou se vyskytnou ve vstupních datech. Toto se zpravidla řeší na úrovni šablon pravidel, kde existují šablony, jejichž podmínka závisí na subslovních jednotkách (předpony, přípony).

Zásadnější nevýhodou především pro nasazení ve statistických hlasových dialogových systémech je nemožnost zpracování neurčitého výstupu automatického rozpoznávání řeči ve formě mřížky. Dále je rovněž problematické získání více výstupních hypotéz spolu s aposteriorními pravděpodobnostmi. Tento problém byl řešen v práci [66]. Přístup autorů spočíval v převedení uspořádané množiny transformačních pravidel na rozhodovací strom s přiřazením aposteriorních pravděpodobností jednotlivým třídám ekvivalence definovaným tímto stromem.

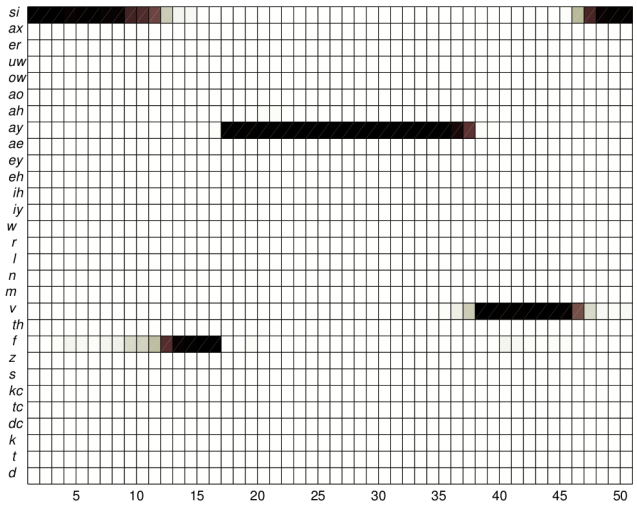
Z výše uvedeného vyplývá, že TBL přístup k trénování klasifikátoru, je vhodný pro systémy s „jistým" vstupem tvořeným posloupností slov. Navíc klasifikátor trénovaný pomocí TBL ve svém základu poskytuje opět pouze první nejlepší hypotézu. Tyto předpoklady nejsou na překážku v úlohách NLP, nicméně v hlasových dialogových systémech, které mohou přirozeně pracovat s neurčitostí, není TBL příliš vhodné.

3.5 Detekce klíčových slov pomocí hierarchických klasifikátorů

V práci [67] se autoři Fousek a Heřmanský věnují alternativnímu přístupu k detekci klíčových slov založenému na hierarchii diskriminativních klasifikátorů a postupné redukci množství informace zpracovávané těmito klasifikátory. Autoři navrhují dvouúrovňové schéma, kde první úroveň klasifikátorů zpracovává posloupnost akustických příznakových vektorů získaných metodou Multi-resolution RASTA a každých 10ms predikuje aposteriorní pravděpodobnosti výskytu jednotlivých fonémů (v citované práci 29 anglických fonémů). Druhá vrstva pak na základě aposteriorních pravděpodobností fonémů v daném časovém okénku generuje aposteriorní pravděpodobnosti výskytu jednotlivých klíčových slov.

Pro predikci aposteriorní pravděpodobnosti jednotlivých fonémů byla natrénována dopředná neuronová síť. Základní vlastností tohoto přístupu je schopnost predikovat aposteriorní pravděpodobnosti výskytu fonému v daném časovém okénku nezávisle na ostatních fonémech. Jinými slovy, diskriminativní klasifikátory mohou ve stejném čase přiřadit vysokou pravděpodobnost výskytu dvěma různým, na základě akustických příznaků nerozlišitelným fonémů. Časových průběh aposteriorních pravděpodobností pro jednotlivé fonémy (výstupy jednotlivých klasifikátorů) je možné zobrazit jako tzv. posteriogram 3.1. Klasifikátory predikující aposteriorní pravděpodobnosti jednotlivých fonémů jsou trénovány z fonémově anotovaných akustických dat.

Pro predikci aposteriorní pravděpodobnosti přítomnosti daného klíčového slova v dané části akustického signálu je trénována vícevrstvá dopředná perceptronová neuronová síť. Její vstup je reprezentován 2929 příznaky získanými z 1010 ms dlouhých úseků posteriogramu (29 aposteriorních pravděpodobností generovaných každých 10 ms). Posuvem tohoto relativně dlouhého časového okna o 10 ms je generován posteriogram pro jednotlivá klíčová slova. Tato druhá vrstva hierarchického klasifikátoru provádí redukci informace, kdy od ekvidistantně vzorkovaných aposteriorních pravděpodobností jednotlivých fonémů se přechází k neekvidistantně rozloženým aposteriorním pravděpodobnostem slov.



Obrázek 3.1: Posteriogram pro anglické slovo *five*. Slovu předchází a následuje ticho, reprezentované fonémem *si*. Obrázek převzat z [67].

Popsaný přístup však vykazuje některé vlastnosti, které jsou překážkou pro použití v rozpoznávání spojité řeči s rozsáhlým slovníkem. Jedná se především o nutnost trénovaní klasifikátorů druhé vrstvy pro každé detekované slovo, metoda nenabízí žádnou možnost, jak vytvořit model nového slova ze subslovních jednotek. Rovněž výpočetní nároky lineárně vzrůstají s počtem detekovaných slov.

Zmíněné problémy vedly pravděpodobně k opuštění tohoto přístupu, nicméně v kontextu této práce je zajímavá analogie mezi využitím hierarchických klasifikátorů pro detekci klíčových slov a hierarchických klasifikátorů pro porozumění řeči.

3.6 Detekce klíčových slov pomocí vážených konečných transducerů

Poněvadž výsledkem zpracování různých digitalizovaných dat, jako je řeč, psané písmo, hudba a podobně, jsou často struktury reprezentující alternativní hypotézy spolu s přiřazenými vahami, je přirozené tyto struktury reprezentovat pomocí mřížek – acyklických vážených konečných automatů. V případech, kdy je třeba nad množinou těchto automatů provádět vyhledávání, je vhodné sestavit jejich index. Takový index je výsledkem vhodných operací a jeho prohledávání je rychlejší, než sekvenční prohledávání dílčích automatů.

Lze najít celou řadu prací, kde se autoři věnují problematice indexace vážených konečných automatů [68, 69, 70]. Tyto práce vychází z teorie vážených konečných transducerů, která je blíže popsána v kapitole 5.2, str. 32. Jejich přístup k problému indexace je založen na tzv. faktorovém automatu. Faktorový automat (kapitola 5.2.2) je takový automat, který přijímá všechny podřetězce původního automatu. Díky vhodnému způsobu zpracování vstupních mřížek lze zachovat odkaz na původní mřížku, ze které daný faktor (cesta faktorovým automatem) vzešla [70], popř. i informaci o čase, v němž se daný faktor v rámci mřížky vyskytuje [69]. Z možných aplikací indexace vážených konečných automatů zmiňme

indexaci slovních a fonémových mřížek za účelem vyhledávání klíčových slov (přesnější angl. termín: *spoken term detection*) [69] nebo identifikaci hudebních skladeb [70].

Důležitou vlastností přístupu založeného na použití faktorového automatu je možnost optimalizovat index pomocí standardních metod pro optimalizaci vážených konečných transducerů. Mezi tyto metody patří odstranění ϵ-přechodů, determinizace a minimalizace. Po aplikaci optimalizací je získán *optimální index* a výpočetní složitost hledání všech výskytů daného řetězce symbolů je pak lineární s délkou dotazu a počtem výskytů hledaného dotazu.

Díky této vlastnosti je přístup faktorového automatu použit i zde – slouží k rychlému výpočtu racionální jádrové funkce (kapitola 7.1) a rovněž k detekci sémantických entit (kapitola 8).

3.7 Shrnutí

V předcházejících odstavcích byly shrnuty různé přístupy nejen k porozumění řeči, ale i další metody z oblasti zpracování řeči. Uveďme nyní, jakým způsobem tyto metody ovlivnily nový model pro porozumění řeči vyvinutý v rámci této práce. První ze zmíněných technologií, *stochastické bezkontextové gramatiky*, jsou velmi srozumitelným a široce akceptovaným prostředkem pro návrh hlasových dialogových systémů, zde jsou použity pro zachycení expertní znalosti týkající se sémantických entit. Dále popsaný *parser se skrytým stavovým vektorem* je zástupce generativních modelů. Zároveň jde o jeden z referenčních modelů, který byl použit pro získání experimentálních výsledků a se kterým je vyvinutý model porovnáván. Druhý z referenčních modelů, *klasifikátory sémantických entit*, je použit jako zástupce třídy diskriminativních modelů. Přístup *transformation based learning* byl popsán pro ilustraci problémů, které mohou vzniknout při přenosu metod určených pro úlohy NLP do prostředí, v nichž figuruje automatické rozpoznávání řeči, které do procesu porozumění zavádí nezanedbatelnou neurčitost. Myšlenka *detekce klíčových slov pomocí hierarchických klasifikátorů* je velmi blízká modelu porozumění řeči popsanému v následujících kapitolách. Namísto klasifikace jednotlivých fonémů a následně predikce výskytu hledaných slov jsou však natrénovány klasifikátory pro predikci dílčích významů a následně jejich výstupy jsou použity v další vrstvě pro dekódování celých sémantických stromů. A konečně *detekce klíčových slov pomocí vážených konečných transducerů* posloužila jako inspirace pro rychlý výpočet racionálních jádrových funkcí a zároveň přístup založený na použití faktorového automatu posloužil i pro detekci sémantických entit ve vstupní promluvě.

Kapitola 4

Cíle disertační práce

Popis architektury hlasových dialogových systémů a jednotlivých metod pro porozumění řeči a zpracování mluvené řeči v předchozí kapitole pak slouží jako motivace ke stanovení jednotlivých cílů této disertační práce. Tyto cíle vyplynuly především z praktických zkušeností autora při výzkumu, vývoji a nasazení hlasových dialogových systémů a dalších technologií pro zpracování řeči – především systémů automatického rozpoznávání řeči a systémů pro indexaci a vyhledávání klíčových slov v audiovizuálních archívech.

Cíl 1: *Vyvinutí modelu porozumění schopného pracovat s neurčitostí vstupu i výstupu*

S ohledem na cílové nasazení v oblasti hlasových dialogových systému bylo prvním z cílů vyvinout model, který umožňuje efektivně pracovat s neurčitostí vzniklou při rozpoznávání řeči. A to nejen ve smyslu schopnosti generovat více hypotéz o významu vstupní promluvy, ale i v možnosti zpracovávat mřížku obsahující více hypotéz o slovním přepisu vstupní promluvy.

Cíl 2: *Využití fonémového rozpoznávače v oblasti porozumění řeči*

Při vývoji hlasového dialogového systému, nebo obecněji libovolného systému automatického rozpoznávání řeči, je největší překážkou potřeba získat dostatečné množství dat pro robustní jazykový model. Tento jazykový model musí dostatečným způsobem pokrývat slovník dané úlohy. Navíc možnosti přenesení znalostí mezi jednotlivými doménami jsou omezené. V projektech řešených autorem zaměřených na hledání klíčových slov a frází však byly s úspěchem použity metody pro rozpoznávání řeči na fonémové úrovni. Přestože tyto metody nedosahovaly přesnosti slovních modelů, tvoří jejich použití zajímavou alternativu k rozpoznávání na úrovni slov právě kvůli náročnosti přípravy slovního jazykového modelu. Proto dalším z cílů je výzkum v oblasti využití rozpoznávání řeči na fonémové úrovni za účelem porozumění řeči a získání významového popisu bez znalosti konkrétních slov vyskytujících se v dané úloze. Jelikož i fonémový rozpoznávač řeči vyžaduje jazykový model na fonémové úrovni, bude se tato práce věnovat i možnostem adaptace fonémového jazykového modelu.

Cíl 3: *Formulace plně pravděpodobnostního diskriminativního modelu*

Předchozí výzkum v oblasti hlasových dialogových systémů na pracovišti autora používal generativní modely pro porozumění řeči. Nicméně experimenty ukázaly, že diskriminativní modely umožňují dosáhnout vyšší přesnosti porozumění. Mezi další cíle zahrneme požadavek vyvinout statistický diskriminativní model, který však bude možné použít v plně pravděpodobnostním modelu hlasového dialogového systému. Tento cíl je formulován především s ohledem na budoucí výzkum v oblasti pravděpodobnostních modelů a rozhodovacích procesů pro řízení dialogu.

Cíl 4: *Návrh vhodné metody pro kombinaci statistického a znalostního přístupu*

Přestože statistický přístup k porozumění řeči je schopen naučit se cílové chování z trénovací množiny, tato množina musí mít dostatečný počet reprezentativních příkladů. Tento předpoklad není v praxi vždy splněn. Nabízí se proto využití znalostního přístupu k vyjádření základních, obecně platných znalostí o dané úloze. Využitím znalostí lze redukovat potřebný počet trénovacích dat. Proto bude část disertační práce věnována i tomu, jak vhodně tuto expertní znalost integrovat do plně pravděpodobnostního diskriminativního modelu.

Cíl 5: *Ověření modelu nad více cílovými doménami*

Posledním cílem disertační práce bude ověření vyvinutého modelu nad více než jedním sémanticky anotovaným korpusem dat z důvodu zabránění „přetrénování" modelu na určitou cílovou doménu.

Kapitola 5

Teoretický základ použitých metod

V této kapitole budou detailněji popsány teoretické základy metod a postupů použitých v disertační práci. Jedná se především o teorii klasifikátorů založených na *support vector machines* (SVM, kapitola 5.1). Jsou popsána i rozšíření SVM klasifikátorů pro klasifikaci do více cílových tříd a metoda pro odhad aposteriorní pravděpodobnosti příslušnosti daného vektoru příznaků do cílové třídy. Klasifikátory založené na SVM jsou použity ve skryté a výstupní vrstvě hierarchickém diskriminativním modelu (kapitoly 7.2 a 7.3).

Další popsanou teoretickou oblastí jsou *vážené konečné automaty* (kapitola 5.2). Jsou popsány algoritmy a operátory pro práci s váženými konečnými automaty. Je zde rovněž zmíněn faktorový automat jako nástroj pro efektivní indexaci všech podřetězců zdrojového automatu. Tyto struktury jsou použity pro reprezentaci slovních a fonémových mřížek na výstupu ze systému automatického rozpoznávání řeči. Rovněž slouží k efektivnímu výpočtu racionálních jádrových funkcí (kapitola 7.1) a také k detekci sémantických entit (kapitola 8).

Následuje výklad teorie *racionální jádrových funkcí* (kapitola 5.3), které umožňují vyčíslení jádrové funkce mezi dvěma mřížkami a tím pádem lze s jejich využitím natrénovat SVM klasifikátor nad trénovací množinou reprezentovanou mřížkami, nikoli příznakovými vektory. Racionální jádrové funkce jsou použity ve vstupní vrstvě hierarchického diskriminativního modelu (kapitola 7.1).

Další text popisuje *stochastické bezkontextové gramatiky* (kapitola 5.4). Jsou zmíněny i lexikalizované gramatiky, které slouží jako základ pro sémantické gramatiky použité v hierarchickém diskriminativním modelu (kapitola 7.3).

Výklad *n-gramových jazykových modelů pro rozpoznávání řeči* (kapitola 5.5) je využit v části věnované adaptaci fonémových jazykových modelů na novou doménu (kapitola 9.4.4).

Poslední dvě kapitoly *parser se skrytým vektorovým stavem* (kapitola 5.6) a *klasifikátory sémantických n-tic* (kapitola 5.7) popisují referenční modely, k nimž jsou vztaženy experimentální výsledky. Tyto modely byly vybrány záměrně – parser se skrytým vektorovým stavem jako reprezentant třídy generativních modelů, klasifikátory sémantických n-tic pak jako představitel diskriminativních modelů.

5.1 Support Vector Machines

Popis klasifikátorů založených na podpůrných vektorech (Support Vector Machines, SVMs) pro lineárně neseparabilní problém byl publikován Corrinou Cortes a Valdimirem Vapnikem v práci *Support-Vector Networks* [71]. Při popisu SVM však budeme vycházet z přehledné publikace *A Tutorial on Support Vector Machines for Pattern Recognition*, jejímž autorem je Christopher J.C. Burges [72].

Předpokládejme, že trénovací množina $\mathcal{T} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$ je tvořena příznakovými vektory $\mathbf{x}_i \in \mathbb{R}^n$ a odpovídajícími cílovými třídami y_i. Uvažujme nejprve úlohu binární klasifikace, kdy vektory trénovací množiny patří právě do jedné ze dvou možných tříd, tj. $y_i \in \{-1, 1\}$. Nejprve budeme uvažovat takovou trénovací množinu, která je lineárně separabilní, tj. v prostoru $\mathbb{R}^n$ existuje lineární nadrovina, která oddělí ty vektory $\mathbf{x}_i$, které náleží do třídy $y_i = -1$ od vektorů, které náleží do třídy $y_i = 1$. Pro tuto trénovací množinu odvodíme optimalizační úlohu pro získání maximum margin klasifikátoru (kapitola 5.1.1). Následně tyto úvahy zobecníme pro případ lineárně neseparabilního problému (kapitola 5.1.2). Dále se budeme zabývat problémem nelineární separace (kapitola 5.1.3) a krátce zmíníme Mercerovu podmínku (kapitola 5.1.4). Nakonec zodpovíme otázky klasifikace do více cílových tříd (kapitola 5.1.5) a odhadu aposteriorní pravděpodobnosti příslušnosti k dané třídě (kapitola 5.1.6).

5.1.1 Lineárně separabilní problém

Uvažujme nyní lineárně separabilní trénovací množinu $\mathcal{T}$ s příznakovými vektory náležícími dvěma třídám. Předpokládejme, že v prostoru $\mathbb{R}^n$ existuje nadrovina H oddělující body $\mathbf{x}_i$ náležící různým cílovým třídám a splňující $\mathbf{x} \cdot \mathbf{w} + b = 0$. Tento předpoklad můžeme formulovat jako:

$$\mathbf{x}_i \cdot \mathbf{w} + b \geq +1 \quad \text{pro } y_i = +1 \tag{5.1}$$

$$\mathbf{x}_i \cdot \mathbf{w} + b \leq -1 \quad \text{pro } y_i = -1 \tag{5.2}$$

Tyto nerovnice lze sloučit do jedné množiny nerovností díky požadavku na množinu cílových tříd $\{-1, +1\}$:

$$y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 \geq 0 \quad \forall i = 1 \ldots l \tag{5.3}$$

Pro oddělující nadrovinu H platí, že její vzdálenost k počátku je možné vyjádřit jako $\frac{|b|}{||\mathbf{w}||}$. Uvažujme nyní body $\mathbf{x}_i$, pro které v nerovnici (5.1) platí rovnost. Tyto body zcela jistě leží na nadrovině $H_1 : \mathbf{x}_i \cdot \mathbf{w} + b = 1$. Obdobně body, pro které platí rovnost v nerovnici (5.2) leží na nadrovině $H_2 : \mathbf{x}_i \cdot \mathbf{w} + b = -1$. Kolmá vzdálenost nadroviny H_1 k počátku je možné zapsat jako $\frac{|1-b|}{||\mathbf{w}||}$, obdobně pro H_2 pak jako $\frac{|-1-b|}{||\mathbf{w}||}$.

Označme d_+ a d_- nejmenší vzdálenost pozitivního příkladu $y_i = +1$, resp. negativního příkladu $y_i = -1$ k oddělující nadrovině. Algoritmus podpůrných vektorů (support vector algorithm) hledá takovou normálu $\mathbf{w}$ oddělující nadroviny H, která maximalizuje součet $(d_+ + d_-)$. Označme tuto vzdálenost anglicky jako *margin* a klasifikátor maximalizující součet $(d_+ + d_-)$ jako *maximum margin classifier*.

Vzhledem k tomu, že jak H, tak H_1 a H_2 mají stejnou normálu $\mathbf{w}$, pak pro vzdálenost d_+ mezi H a H_1 a pro vzdálenost d_- mezi H a H_2 (d_-) platí:

$$d_+ = d_- = \frac{1}{||\mathbf{w}||} \tag{5.4}$$

Nalezení oddělující nadroviny maximalizující součet $(d_+ + d_-)$ je pak ekvivalentní nalezení oddělující nadroviny minimalizující $||\mathbf{w}||$ při respektování množiny nerovností (5.3).

Jak bylo zmíněno výše, nalezení takové oddělující nadroviny H, která maximalizuje součet $(d_+ + d_-)$ je ekvivalentní minimalizaci $||\mathbf{w}||$ a tím také minimalizaci $\frac{1}{2}||\mathbf{w}||^2$ při splnění podmínek (5.3). Přejděme tedy k formulaci problému pomocí Langrangeových multiplikátorů $\alpha_i,\ i = 1 \dots l$, přičemž jednotlivá α_i odpovídají multiplikátorům rovnic (5.1) nebo (5.2) pro dvojici $(\mathbf{x}_i, y_i)$. Odpovídající Lagrangeova funkce má tvar:

$$\begin{aligned} L_P &\equiv \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{l} \alpha_i \left[y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1\right] \to \min_{\mathbf{w},b} \\ &\equiv \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{l} \alpha_i y_i(\mathbf{x}_i \cdot \mathbf{w} + b) + \sum_{i=1}^{l} \alpha_i \to \min_{\mathbf{w},b} \end{aligned} \tag{5.5}$$

Výše zmíněná Langrangeova funkce definuje primární úlohu konvexního kvadratického programování minimalizující L_P podle $\mathbf{w}$ a b za podmínek $\frac{\partial}{\partial \alpha_i} L_P = 0$, $\alpha_i \geq 0$ (množina omezujících podmínek $\mathcal{C}_1$). Řešení primární úlohy odpovídá řešení duálního problému – maximalizace L_P za podmínek $\frac{\partial}{\partial w_k} L_P = 0$, $\frac{\partial}{\partial b} L_P = 0$ a zároveň za podmínek $\alpha_i \geq 0$ (množina omezujících podmínek $\mathcal{C}_2$). Tato duální formulace problému má tu vlastnost, že maximum L_P s ohledem na $\mathcal{C}_2$ nastává při stejných hodnotách $\mathbf{w}$, b a α_i jako minimum L_P s ohledem na omezení $\mathcal{C}_1$.

Podmínky $\frac{\partial}{\partial w_k} L_P = 0$, $\frac{\partial}{\partial b} L_P = 0$ vedou na rovnice:

$$\mathbf{w} - \sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i = 0 \tag{5.6}$$

$$\sum_{i=1}^{l} \alpha_i y_i = 0 \tag{5.7}$$

Rovnice (5.6) a (5.7) mohou být dosazeny do (5.5). Pak lze vyjádřit duální podobu Langrangeovy funkce:

$$L_D \equiv \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \to \max_{\alpha_i} \tag{5.8}$$

při splnění podmínek (5.6) a (5.7).

V takto formulované úloze odpovídá každému trénovacímu příkladu $(\mathbf{x}_i, y_i)$ jeden Lagrangeův multiplikátor α_i. Ty vektory $\mathbf{x}_i$, pro které $\alpha_i > 0$, se nazývají *podpůrné vektory* a leží na jedné z nadrovin H_1 nebo H_2. Tyto podpůrné vektory leží nejblíže rozhodovací nadrovině H a jsou nezbytnou součástí trénovací množiny $\mathcal{T}$. Pokud by veškeré ostatní

trénovací příklady (pro které $\alpha_i = 0$) byly z množiny $\mathcal{T}$ odstraněny, stále by řešením optimalizačního problému byla táž nadrovina H.

Po optimalizaci výše zmíněného optimalizačního problému vzhledem k trénovací množině $\mathcal{T}$ je možné klasifikovat libovolný vektor $\mathbf{x}$ pomocí pravidla:

$$\hat{y} = \text{sgn}(\mathbf{w} \cdot \mathbf{x} + b) \tag{5.9}$$

kde funkce $\text{sgn}(x)$ je definována jako:

$$\text{sgn}(x) = \begin{cases} +1 & \text{pokud } x \geq 0 \\ -1 & \text{pokud } x < 0 \end{cases} \tag{5.10}$$

5.1.2 Lineárně neseparabilní problém

Aplikaci výše uvedené optimalizační úlohy na lineárně neseparabilní problémy vede k divergujícímu algoritmu – duální Lagrangeova funkce L_D roste nade všechny meze. Řešením je uvolnění omezení daných rovnicemi (5.1) a (5.2) pomocí zavedení tzv. *slack proměnných* $\xi_i,\ i = 1 \ldots l$ následujícím způsobem:

$$\mathbf{x}_i \cdot \mathbf{w} + b \geq +1 - \xi_i \quad \text{pro } y_i = +1 \tag{5.11}$$

$$\mathbf{x}_i \cdot \mathbf{w} + b \leq -1 - \xi_i \quad \text{pro } y_i = -1 \tag{5.12}$$

kde platí $\xi_i \geq 0\ \forall i$. Proměnné ξ_i vyčíslují míru chybné klasifikace vzhledem k oddělující nadrovině H. Tudíž $\sum_i \xi_i$ vyčísluje horní mez na počet chyb predikce nad trénovací množinou $\mathcal{T}$. Kriteriální funkce je pak modifikována takovým způsobem, aby tuto horní mez minimalizovala:

$$\min_{\mathbf{w},b,\xi} \left[\frac{1}{2}||\mathbf{w}||^2 + C \sum_{i=1}^{l} \xi_i \right] \tag{5.13}$$

Lagrangeova funkce je po zavedení slack proměnných modifikována zavedením množiny Lagrangeových multiplikátorů β_i:

$$L_P = \frac{1}{2}||\mathbf{w}||^2 + C \sum_{i=1}^{l} \xi_i - \sum_{i=1}^{l} \alpha_i \left[y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 + \xi_i \right] - \sum_{i=1}^{l} \beta_i \xi_i \rightarrow \min_{\mathbf{w},b,\xi} \tag{5.14}$$

Jedinečná vlastnost výše zmíněného způsobu zavedení slack proměnných spočívá v tom, že duální Lagrangeova funkce neobsahuje ani slack proměnné ξ_i, ani odpovídající Lagrangeovy multiplikátory β_i [71]. Potom:

$$L_D \equiv \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \rightarrow \max_{\alpha_i} \tag{5.15}$$

Za podmínek:

$$0 \leq \alpha_i \leq C \tag{5.16}$$

$$\sum_{i=1}^{l} \alpha_i y_i = 0 \tag{5.17}$$

Hledaná nadrovina H je pak dána normálou $\mathbf{w}$:

$$\mathbf{w} = \sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i \tag{5.18}$$

Výpočet parametru b vyplývá z Karushových-Kuhnových-Tuckerových podmínek na řešení výše uvedeného optimalizačního problému. Uveďme pouze jedinou z těchto podmínek určující hodnotu parametru b:

$$\alpha_i \left[y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 + \xi_i \right] = 0 \tag{5.19}$$

Na základě této rovnice stačí znalost jediného prvku trénovací množiny $\mathcal{T}$ pro určení prahu b. Nicméně jako numericky stabilnější se jeví výpočet b_i pro každý prvek trénovací množiny a následné zprůměrování vypočtených hodnot [72].

Zmiňme ještě význam parametru C. Tento parametr je volen jako součást kriteriální funkce před optimalizací oddělující nadroviny H. Vyšší hodnoty C způsobují vyšší váhu chyb nad trénovací množinou. Platí tedy, že čím vyšší hodnota C je použita, tím více se oddělující nadrovina H adaptuje na body trénovací množiny – dochází k jevu nazývanému *přetrénování*. Při trénování klasifikátoru se doporučuje použít metodu křížové validace a vyhodnocení klasifikační chyby pro různé hodnoty C [73], např. $C = \{2^k\},\ k = -2, -1, 0, 1, 2$.

5.1.3 Nelineární separace

Ve výše zmíněných rovnicích (5.15) pro optimalizační úlohu se vždy vektory trénovací množiny objevují ve tvaru skalárního součinu. Uvažujme nyní případ, kdy oddělující nadrovina H není lineární. Předpokládejme, že existuje zobrazení Ψ:

$$\Psi : \mathbb{R}^n \mapsto \mathcal{H} \tag{5.20}$$

Zobrazení Ψ realizuje vzájemně jednoznačné zobrazení z příznakového prostoru do Euklidovského prostoru $\mathcal{H}$ vyšší, potenciálně nekonečné, dimenze. Očekává se, že data, která nejsou v prostoru $\mathbb{R}^n$ lineárně separabilní již budou po zobrazení do vyšší dimenze prostoru $\mathcal{H}$ lineárně separabilní.

Nyní aplikujeme toto zobrazení na vektory $\mathbf{x}_i$ množiny $\mathcal{T}$ a v prostoru $\mathcal{H}$ začněme řešit algoritmus podpůrných vektorů. Platí, že po aplikaci zobrazení Ψ, závisí tento algoritmus pouze na skalárních součinech $\Psi(\mathbf{x}_i) \cdot \Psi(\mathbf{x}_j)$ v prostoru $\mathcal{H}$.

Tímto byl skalární součin přenesen z prostoru $\mathbb{R}^n$ do prostoru $\mathcal{H}$. Nazývejme tedy funkci, která dvojici vektorů $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^n$ přiřadí hodnotu skalárního součinu z prostoru $\mathcal{H}$, jako *jádrovou funkci* (angl. kernel function) $K(\mathbf{x}_i, \mathbf{x}_j)$:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \Psi(\mathbf{x}_i) \cdot \Psi(\mathbf{x}_j) \tag{5.21}$$

Tuto jádrovou funkci nyní můžeme použít v algoritmu podpůrných vektorů. Následuje ekvivalent rovnice (5.15) po zobrazení vektorů $\mathbf{x}_i$ a $\mathbf{x}_j$ do prostoru $\mathcal{H}$:

$$\begin{aligned} L_D &\equiv \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \Psi(\mathbf{x}_i)) \cdot \Psi(\mathbf{x}_j \\ &\equiv \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \rightarrow \max_{\alpha_i} \end{aligned} \tag{5.22}$$

Pro získání klasifikačního pravidla klasifikujícího vektor příznaků $\mathbf{x}$ do jedné z cílových tříd dosaďme nejprve rovnici (5.18) do (5.9).

$$\hat{y} = \operatorname{sgn}\left[(\sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i) \cdot \mathbf{x} + b\right] \tag{5.23}$$

Proveďme nyní aplikaci zobrazení Ψ na podpůrné vektory $\mathbf{x}_i$ a na vstupní vektor $\mathbf{x}$ a substituci jádrové funkce $K(\cdot,\cdot)$:

$$\begin{aligned}\hat{y} &= \operatorname{sgn}\left[(\sum_{i=1}^{l} \alpha_i y_i \Psi(\mathbf{x}_i) \cdot \Psi(\mathbf{x})) + b\right] \\ &= \operatorname{sgn}\left[\sum_{i=1}^{l} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b\right]\end{aligned} \tag{5.24}$$

Poznamenejme, že výpočet jádrové funkce se obejde bez explicitní znalosti zobrazení $\Psi(\cdot)$ i bez znalosti prostoru $\mathcal{H}$. Jediné, co musí být zaručeno pro konvergenci tohoto algoritmu, je existence prostoru $\mathcal{H}$ a zobrazení Ψ, která je dána tzv. Mercerovou podmínkou (kapitola 5.1.4). Náhrada skalárního součinu jádrovou funkcí se v angličtině nazývá *kernel trick*. Použití této náhrady je velmi časté nejen na poli SVM, ale například i pro shlukování [74] nebo pro analýzu hlavních komponent [75].

Uveďme nyní některé z používaných jádrových funkcí pro příznakové vektory $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^n$:

- *Polynomiální homogenní*:
$$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j)^d \tag{5.25}$$
- *Polynomiální nehomogenní*:
$$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^d \tag{5.26}$$
- *Radiální bázová funkce* (RBF):
$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\gamma^2}} \tag{5.27}$$

Na závěr definujme vzdálenost k rozhodovací nadrovině $d(\mathbf{x})$ jako argument funkce $\operatorname{sgn}(\cdot)$ z rovnice (5.24):

$$d(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b \tag{5.28}$$

Potom rozhodovací pravidlo odvozené z rovnice (5.24) nabývá tvaru:

$$\hat{y} = \operatorname{sgn} d(\mathbf{x}) \tag{5.29}$$

5.1.4 Mercerova podmínka

Existence prostoru $\mathcal{H}$ a zobrazení Ψ je dáno tzv. Mercerovou podmínkou, která říká, že zobrazení Ψ a prostor H existují tehdy a jen tedy, pokud pro libovolnou funkci $g(\mathbf{x})$ takovou, že:

$$\int g(\mathbf{x})^2 d\mathbf{x} \leq \infty \tag{5.30}$$

platí:

$$\int K(\mathbf{x}, \mathbf{y}) g(\mathbf{x}) g(\mathbf{y}) d\mathbf{x} d\mathbf{y} \geq 0 \tag{5.31}$$

Více o Mercerově podmínce v publikacích [10, 71, 72].

5.1.5 Klasifikace do více tříd

Výše uvedené optimalizační problémy řeší binární klasifikaci do dvou cílových tříd. Klasifikace do více než dvou cílových tříd je pak založena na využití binárních klasifikátorů. Předpokládejme, že trénovací množina $\mathcal{T}$ je složena z dvojic $(\mathbf{x}_i, y_i),\ i = 1 \cdots l$, $\mathbf{x}_i \in \mathbb{R}^n$, $y_i \in \{1 \cdots k\}$, kde k je počet různých cílových tříd. Existují tři základní schémata pro trénování takových klasifikátorů [76]:

- *One-against-all* – spočívá v natrénování k binárních klasifikátorů, které diskriminují příznakové vektory příslušející k dané cílové třídě oproti zbylým třídám [77]. Tento přístup vede ke k rozhodovacím funkcím a cílová třída pro bod $\mathbf{x}$ je vybrána jako třída odpovídající nejvzdálenější oddělující nadrovině $H^{(c)},\ c = 1 \ldots k$:

$$\hat{y} = \arg \max_{c=1\ldots k} \left[\sum_{i=1}^{l} \alpha_i^{(c)} y_i^{(c)} K(\mathbf{x}_i, \mathbf{x}) + b^{(c)} \right] \tag{5.32}$$

  kde $y_i^{(c)} = +1$, pokud $y_i = c$, jinak $y_i^{(c)} = -1$. Dále $\alpha_i^{(c)}$ a $b^{(c)}$ jsou parametry jednotlivých binárních klasifikátorů.

- *One-against-one* – při tomto postupu je natrénováno $\frac{k(k-1)}{2}$ binárních klasifikátorů, přičemž každý z těchto klasifikátorů diskriminuje vždy dvě různé třídy navzájem. Při trénování klasifikátoru diskriminujícího třídy m a n jsou použity pouze ty prvky trénovací množiny $\mathcal{T}$, pro které platí $y_i \in \{m, n\},\ i = 1 \ldots l$. Výběr cílové třídy pro neznámý příznakový vektor $\mathbf{x}$ se provede následující hlasovací strategií [78]:

  1. Pokud klasifikátor diskriminující třídy m a n predikuje, že příslušnou třídou je m, pak je počet hlasů pro třídu m zvýšen o 1, v opačném případě je o 1 zvýšen počet hlasů pro třídu n.
  2. Po predikci všemi $\frac{k(k-1)}{2}$ klasifikátory jsou hlasy pro jednotlivé třídy sečteny a vektor $\mathbf{x}$ je přiřazen do třídy s nejvyšším počtem hlasů.

- *Orientovaný acyklický graf* – obdobně jako při využití schématu one-against-one se trénuje $\frac{k(k-1)}{2}$ klasifikátorů, nicméně před fází predikce jsou tyto klasifikátory sestaveny do struktury orientovaného acyklického grafu, přičemž graf obsahuje právě $\frac{k(k-1)}{2}$ uzlů odpovídajících jednotlivým binárním klasifikátorům a k listových uzlů

odpovídajících cílovým třídám [79]. Výhoda spočívá v predikci pouze pomocí klasifikátorů ležících v uzlech na cestě z kořene stromu do uzlu odpovídající cílové třídy a tudíž v nižší výpočetní náročnosti predikce oproti metodě One-against-one.

V práci [76] bylo provedeno zhodnocení jednotlivých přístupů a jejich porovnání na různých datových množinách. Autoři zmiňují, že mezi těmito metodami nejsou statisticky významné rozdíly a pro praktickou implementaci preferují využití metod One-against-one nebo Orientovaného acyklického grafu z důvodu menší výpočetní náročnosti při trénování klasifikátoru, které je dosaženo menším počtem prvků trénovací množiny využitých pro trénování dílčích binárních klasifikátorů. Poznamenejme, že použitá implementace využívá schéma One-against-one [80].

5.1.6 Odhad aposteriorní pravděpodobnosti

Přestože výše definované SVM nepodporují přímo odhad aposteriorní pravděpodobnosti, je možné pomocí metody popsané J. Plattem [81] provést mapování vzdálenosti predikovaného bodu k rozhodovací nadrovině na aposteriorní pravděpodobnost. Předpokládejme, že cílovou třídu vektoru $\mathbf{x}$ považujeme za náhodnou proměnnou Y. Úkolem je získat odhad aposteriorní pravděpodobnosti podmíněný pozorováním vstupního příznakového vektoru $\mathbf{x}$, tj. $\mathrm{P}(\mathsf{Y} = 1|\mathbf{x})$. Za tímto účelem jsou natrénovány parametry γ a δ sigmoidální funkce:

$$\mathrm{P}(y = 1|\mathbf{x}) = p(\mathbf{x}) \approx \frac{1}{1 + \exp\left[\gamma \cdot d(\mathbf{x}) + \delta\right]} \tag{5.33}$$

kde $d(\mathbf{x})$ je dáno rovnicí (5.28) a parametry sigmoidy γ, δ jsou odhadovány metodou maximální věrohodnosti. Za tímto účelem je definována nová trénovací množina $(d(\mathbf{x}_i), t_i)$, kde t_i je cílová pravděpodobnost definovaná jako:

$$t_i = \frac{y_i + 1}{2} \in \{0, 1\} \tag{5.34}$$

Maximalizace věrohodnostní funkce vzhledem k parametrům γ, δ odpovídá minimalizaci jejího záporného logaritmu daného jako:

$$-\sum_i t_i \log(p_i) + (1 - t_i) \log(1 - p_i) \to \min_{\gamma,\delta} \tag{5.35}$$

kde

$$p_i = \frac{1}{1 + \exp\left[\gamma \cdot d(\mathbf{x}_i) + \delta\right]} \tag{5.36}$$

Minimalizaci je nutné provádět tak, aby nedošlo k vychýlení pravděpodobnostního rozdělení směrem k trénovacím datům a tudíž ke špatné schopnosti predikce na datech neviděných. Použitá implementace [80] používá pětinásobnou křížovou validaci pro odhad těchto parametrů a Newtonovu metodu pro optimalizaci kritéria z rovnice (5.35) [82].

Pro případ více cílových tříd je úloha odhadu aposteriorních pravděpodobností komplikovanější. Při použití strategie one-against-one je možné získat pouze párové odhady:

$$r_{ij} = \mathrm{P}(\mathsf{Y} = i|\mathsf{Y} \in \{i, j\}, \mathbf{x}) \tag{5.37}$$

Cílem je z odhadů r_{ij} získat odhad $p_i = \mathrm{P}(y = i|\mathbf{x}),\ i = 1, \dots k$, kde k je počet cílových tříd. Použitá implementace používá přístup popsaný v práci [83] spočívající v optimalizaci kritéria:

$$\min_{\mathbf{p}} \sum_{i=1}^{k} \sum_{j:j\neq i} (r_{ji}p_i - r_{ij}p_j)^2 \tag{5.38}$$

kde vektor $\mathbf{p} = [p_i],\ i = 1, \dots k$ reprezentuje pravděpodobnostní rozdělení nad množinou cílových tříd, z čehož vyplývá omezující podmínka pro minimalizaci:

$$\sum_{i=1}^{k} p_i = 1 \tag{5.39}$$

$$p_i \geq 0, i = 1, \dots k \tag{5.40}$$

Ve zmíněné práci bylo ukázáno, že výše uvedený optimalizační problém má jedinečné řešení, dané jako

$$\begin{bmatrix} \mathbf{Q} & \mathbf{e} \\ \mathbf{e}^\intercal & 0 \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ b \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} \tag{5.41}$$

kde $\mathbf{e}$ je sloupcový k dimenzionální vektor obsahující jedničky, $\mathbf{0}$ je sloupcový k dimenzionální vektor obsahující nuly, b je Lagrangeův multiplikátor omezení (5.39) a matice $\mathbf{Q}$ je dána jako:

$$\mathbf{Q} = [Q_{ij}], \quad Q_{ij} = \begin{cases} \sum_{s:s\neq i} r_{si}^2 & \text{pokud } i = j, \\ -r_{ji}r_{ij} & \text{pokud } i \neq j \end{cases} \tag{5.42}$$

Pro řešení rovnice (5.41) je v práci [83] uváděn jednoduchý iterativní algoritmus. Zároveň je pro tento algoritmus proveden i důkaz jeho konvergence.

5.1.7 Normalizace jádrových funkcí

Při řešení mnoha úloh klasifikace je vhodné provádět jisté předzpracování příznakových vektorů $\mathbf{x}$ před trénováním klasifikátoru. Jednou z možností je provádět normalizaci na jednotkovou normu ve vstupním prostoru $\mathbf{x} \in \mathbb{R}^n$:

$$\tilde{\mathbf{x}} = \frac{\mathbf{x}}{||\mathbf{x}||_2} \tag{5.43}$$

Po normalizaci leží vektory $\mathbf{x}$ na jednotkové kouli v prostoru $\mathbb{R}^n$. Algoritmus podpůrných vektorů je ale navržen k nalezení optimální oddělovací nadroviny v prostoru příznakových vektorů $\Psi(\mathbf{x})$ z prostoru $\mathcal{H}$, který je získán obecně nelineárním zobrazením prostoru $\mathbb{R}^n$, čímž může dojít k porušení normalizace.

Řešením je nenormalizovat vektory $\mathbf{x}$ ve vstupním prostoru $\mathbb{R}^n$, ale až po jejich zobrazení $\Psi(\mathbf{x})$ do příznakového prostoru $\mathcal{H}$ [84, 85]. Toho dosáhneme následující normalizací jádrové funkce:

$$\tilde{K}(\mathbf{x}_i, \mathbf{x}_j) = \frac{K(\mathbf{x}_i, \mathbf{x}_j)}{\sqrt{K(\mathbf{x}_i, \mathbf{x}_i) \cdot K(\mathbf{x}_j, \mathbf{x}_j)}} \tag{5.44}$$

Platí, že $\tilde{K}(\mathbf{x}_i, \mathbf{x}_i) = 1$. Odtud vyplývá, že vektory $\Psi(\mathbf{x})$ leží v prostoru $\mathcal{H}$ na jednotkové kouli. Důležité je, že jádrová funkce $\tilde{K}(\mathbf{x}_i, \mathbf{x}_i)$ rovněž splňuje Mercerovu podmínku, neboť tato jádrová funkce je skalárním součinem v prostoru $\mathcal{H}$:

$$\tilde{K}(\mathbf{x}_i, \mathbf{x}_j) = \tilde{\Psi}(\mathbf{x}_i) \cdot \tilde{\Psi}(\mathbf{x}_j) \tag{5.45}$$

kde normalizované zobrazení $\tilde{\Psi}$ je dáno jako:

$$\tilde{\Psi}(\mathbf{x}) = \frac{\Psi(\mathbf{x})}{||\Psi(\mathbf{x})||} = \frac{\Psi(\mathbf{x})}{\sqrt{K(\mathbf{x}, \mathbf{x})}} \tag{5.46}$$

Pro lineární jádrovou funkci je normalizace v příznakovém prostoru $\mathcal{H}$ ekvivalentní normalizaci ve vstupním prostoru $\mathbb{R}^n$, neboť v tomto případě jsou rovnice (5.43) a (5.44) ekvivalentní. Dále poznamenejme, že radiální bázová jádrová funkce je již z definice normalizována, tj. že $K_{\mathrm{RBF}}(\mathbf{x}_i, \mathbf{x}_i) = 1$. Autoři [84] popisují významné zlepšení na vybraných referenčních úlohách z oblasti klasifikace při použití výše zmíněné normalizace jádrové funkce.

5.2 Vážené konečné automaty

Vážené konečné automaty, lépe řečeno vážené konečné akceptory a transducery (viz níže) jsou matematickou strukturou velice široce používanou nejen na poli zpracování mluvené řeči, ale i v jiných oblastech, např. pro identifikaci písní [70, 86], statistický strojový překlad [87] nebo k detekci pojmenovaných entit [88]. Použití v řečových technologiích je velmi široké, s využitím vážených konečných automatů lze implementovat výslovnostní slovníky, jazykové modely i dekódovací strategie [6, 7, 40, 89]. Na vyšších úrovních pak byly vážené konečné automaty použity pro indexaci audio archívů [69], pro porozumění mluvené řeči [90] nebo pro řízení hlasových dialogových systémů [26]. V této práci byly vážené konečné automaty použity pro reprezentaci vstupních promluv, ať již ve formě řetězce slov, nebo slovní či fonémové mřížky [7, 36]. Dále bylo vážených konečných transducerů použito pro implementaci racionálních jádrových funkcí využívaných v SVM klasifikátorech pro porozumění mluvené řeči (kapitola 7).

5.2.1 Definice pojmů

Pro definici pojmů použijeme notaci zavedenou v práci [91]. *Polookruh* $\mathbb{K} = (\mathcal{K}, \oplus, \otimes, \bar{0}, \bar{1})$ je definován nad uzavřenou množinou hodnot $\mathcal{K}$ a je dán operacemi $\oplus$ a $\otimes$ a hodnotami $\bar{0}$, $\bar{1}$. Operace $\oplus$ je asociativní, komutativní s neutrálním prvkem $\bar{0}$. Operace $\otimes$ je asociativní s neutrálním prvkem $\bar{1}$, distributivní vzhledem k $\oplus$ a prvek $\bar{0}$ je vhledem k této operaci nulovým prvkem, tj. $a \otimes \bar{0} = \bar{0} \otimes a = \bar{0}, a \in \mathcal{K}$. Uveďme některé příklady polookruhů:

- *Pravděpodobnostní polookruh*, definovaný jako:

$$\mathcal{K} \equiv \mathbb{R}_+ \quad x \oplus y \equiv x + y \quad x \otimes y \equiv x \times y \quad \bar{0} \equiv 0 \quad \bar{1} \equiv 1$$

- *Logaritmický polookruh*, definovaný jako:

$$\mathcal{K} \equiv \mathbb{R} \cup \{-\infty, +\infty\} \quad x \oplus y \equiv -\log(e^{-x} + e^{-y}) \quad x \otimes y \equiv x + y \quad \bar{0} \equiv +\infty \quad \bar{1} \equiv 0$$

Logaritmický polookruh je izomorfní s pravděpodobnostním polookruhem použitím zobrazení $-\log(\cdot)$ z množiny hodnot pravděpodobnostního polookruhu do množiny hodnot polookruhu logaritmického.

- *Tropický polookruh*, definovaný jako:

$$\mathcal{K} \equiv \mathbb{R} \cup \{-\infty, +\infty\} \quad x \oplus y \equiv min\{x, y\} \quad x \otimes y \equiv x + y \quad \bar{0} \equiv +\infty \quad \bar{1} \equiv 0$$

Vážený konečný transducer $T = (\mathcal{A}, \mathcal{B}, \mathcal{Q}, I, \mathcal{F}, \mathcal{E}, \lambda, \rho)$ nad polookruhem $\mathbb{K}$ je dán:

- konečnou vstupní abecedou $\mathcal{A}$,
- konečnou výstupní abecedou $\mathcal{B}$,
- konečnou množinou stavů $\mathcal{Q}$,
- množinou počátečních stavů $I \subseteq \mathcal{Q}$,
- množinou koncových stavů $\mathcal{F} \subseteq \mathcal{Q}$,
- konečnou množinou přechodů $\mathcal{E} \subseteq \mathcal{Q} \times (\mathcal{A} \cup \{\epsilon\}) \times (\mathcal{B} \cup \{\epsilon\}) \times \mathbb{K} \times \mathcal{Q}$,
- váhovým ohodnocením počátečních stavů $\lambda : I \to \mathbb{K}$,
- váhovým ohodnocením koncových stavů $\rho : \mathcal{F} \to \mathbb{K}$.

Speciální symbol ϵ značí prázdný řetězec. Přechod označený tímto symbolem má i svůj speciální význam při kompozici transducerů (více v kapitole 5.2.5). Označme $\mathcal{E}[q]$ množinu přechodů ze stavu $q \in \mathcal{Q}$. Pro přechod $e \in \mathcal{E}$ označme $p[e]$ počáteční stav přechodu e, $n[e]$ koncový stav přechodu e, $i[e]$ vstupní symbol přechodu e, $o[e]$ výstupní symbol přechodu e a $w[e] \in \mathbb{K}$ odpovídající váhu přechodu.

Potom *cesta* $\pi = e_1 \dots e_k$ je posloupností po sobě následujících přechodů, pro které platí: $n[e_{i-1}] = p[e_i]$, $i = 2, \dots k$. Jako *řetězec* budeme nazývat automat, ve kterém existuje právě jedna cesta nenulové délky. Funkce n, p, w mohou být rozšířeny i pro cesty následujícím způsobem:

$$\begin{aligned} n[\pi] &= n[e_k] \\ p[\pi] &= p[e_1] \\ w[\pi] &= \bigotimes_{i=1}^{k} w[e_i] \end{aligned} \tag{5.47}$$

Dále, funkci w definujeme i pro libovolnou konečnou množinu cest $\mathcal{R}$ jako:

$$w[\mathcal{R}] = \bigoplus_{\pi \in \mathcal{R}} w[\pi] \tag{5.48}$$

Označme množinu cest ze stavu q do q' jako $\mathcal{P}(q, q')$. Jako $\mathcal{P}(q, x, y, q')$ označíme množinu cest z q do q' se vstupními symboly $x \in \mathcal{A}^*$ a výstupními symboly $y \in \mathcal{B}^*$. Definujme nyní funkce $\mathcal{P}(\cdot, \cdot)$ a $\mathcal{P}(\cdot, \cdot, \cdot, \cdot)$ i nad podmnožinami $\mathcal{R}, \mathcal{R}' \subseteq \mathcal{Q}$:

$$\begin{aligned} \mathcal{P}(\mathcal{R}, \mathcal{R}') &= \bigcup_{q \in \mathcal{R}, q' \in \mathcal{R}'} \mathcal{P}(q, q') \\ \mathcal{P}(\mathcal{R}, x, y, \mathcal{R}') &= \bigcup_{q \in \mathcal{R}, q' \in \mathcal{R}'} \mathcal{P}(q, x, y, q') \end{aligned} \tag{5.49}$$

Váhu přiřazenou transducerem T libovolné dvojici řetězců $(x, y) \in \mathcal{A}^* \times \mathcal{B}^*$ definujme jako:

$$T(x,y) = \bigoplus_{\pi \in \mathcal{P}(I,x,y,\mathcal{F})} \lambda\left[p[\pi]\right] \otimes w[\pi] \otimes \rho\left[n[\pi]\right] \tag{5.50}$$

Jinými slovy, váha přiřazená dvojici vstupní řetězec x a výstupní řetězec y je dána $\oplus$-součtem vah všech cest z libovolného stavu z množiny I do libovolného stavu z množiny $\mathcal{F}$ se vstupním řetězcem x a výstupním řetězcem y $\otimes$-vynásobených cenou odpovídajícího počátečního a koncového stavu. Je-li $\mathcal{P}(I, x, y, \mathcal{F}) = \emptyset$, pak $T(x, y) = \bar{0}$

Vážený konečný akceptor A je speciálním případem váženého konečného transduceru. Akceptor nedefinuje (váženou) relaci mezi vstupními a výstupními řetězci, ale pouze vahou oceňuje cestu akceptorem. Z praktických důvodů je vhodné akceptor zavést jako transducer se stejnou vstupní a výstupní abecedou $\mathcal{A}$ definující identickou relaci mezi vstupními a výstupními symboly, tj. $i[e] = o[e]\ \forall e \in \mathcal{E}$. Tím se například zjednodušuje zápis kompozice konečného transduceru a akceptoru.

V dalším textu bude rovněž používán termín *(vážený konečný) automat*. Tento termín bude použit na místech, kde není nutné explicitně rozlišovat, zda se jedná o vážený konečný transducer nebo o vážený konečný akceptor.

5.2.2 Algoritmy nad váženými konečnými transducery

Sjednocení – pokud T_1 převádí řetězec x na řetězec y s vahou a a T_2 převádí řetězec v na w s vahou b, potom sjednocení $T_1 \oplus T_2$ převádí x na y s vahou a a v na w s vahou b:

$$T_1 \oplus T_2(x,y) = T_1(x,y) \oplus T_2(x,y) \tag{5.51}$$

Konkatenace – pokud T_1 převádí řetězec x na řetězec y s vahou a a T_2 převádí řetězec v na w s vahou b, potom konkatenace $T_1 \otimes T_2$ převádí xv na yw s vahou $a \otimes b$:

$$T_1 \otimes T_2(x,y) = \bigoplus_{x=x_1x_2, y=y_1y_2} T_1(x,y) \otimes T_2(x,y) \tag{5.52}$$

Opakování – pokud T převádí řetězec x na řetězec y s vahou a, pak n-násobné opakování T^n převádí řetězec x^n na řetězec y^n s vahou $\otimes_{i=1}^{n} a$:

$$T^n(x,y) = \bigotimes_{i=1}^{n} T(x,y) \tag{5.53}$$

$$T^0(x,y) = \bar{1} \tag{5.54}$$

Kleeneho uzávěr transduceru T je sjednocení $0, 1, \ldots \infty$ jeho opakování:

$$T^*(x,y) = \bigoplus_{n=0}^{\infty} T^n(x,y) \tag{5.55}$$

Kleeneho plus transduceru T je sjednocení $1, 2, \ldots \infty$ jeho opakování:

$$T^+(x,y) = \bigoplus_{n=1}^{\infty} T^n(x,y) \tag{5.56}$$

Inverze – převádí-li transducer T řetězec x na y s vahou a, pak jeho inverze T^{-1}, převádí y na x s vahou a:

$$T^{-1}(x, y) = T(y, x) \tag{5.57}$$

Kompozice předpokládá, že $\mathbb{K}$ je komutativní. Převádí-li transducer T_1 řetězec x na z s vahou a a transducer T_2 řetězec z na y s vahou b, pak kompozice $T_1 \circ T_2$ převádí x na y s vahou $a \otimes b$:

$$T_1 \circ T_2(x, y) = \bigoplus_z T_1(x, z) \otimes T_2(z, y) \tag{5.58}$$

Hrany v T_1 s výstupním symbolem ϵ, popř. v T_2 se vstupním symbolem ϵ vyžadují speciální zpracování, více v [92].

Projekce transduceru na vstupní symboly $\Pi_1(T)$ transduceru T je definována jako [8]:

$$\Pi_1(T)(x) = \bigoplus_y T(x, y) \tag{5.59}$$

Obdobně projekce transduceru T na výstupní symboly $\Pi_2(T)$:

$$\Pi_2(T)(y) = \bigoplus_x T(x, y) \tag{5.60}$$

Faktorový automat $F(T)$ je minimální deterministický automat akceptující množinu všech faktorů automatu T. Faktor je označení podřetězce úspěšné cesty. Rozhodnutí, zda řetězec x je faktorem T, je s využitím $F(T)$ lineární v čase s délkou $|x|$ [70]. Pro konstrukci faktorového automatu nejprve označme $\alpha[q]$ nejkratší vzdálenost z počátečních stavů T do stavu q a $\beta[q]$ nejkratší vzdálenost ze stavu q do koncových stavů T [69]:

$$\alpha[q] = \bigoplus_{\pi \in \mathcal{P}(I,q)} \lambda(p[\pi]) \otimes w[\pi] \tag{5.61}$$

$$\beta[q] = \bigoplus_{\pi \in \mathcal{P}(q,\mathcal{F})} w[\pi] \otimes \rho(p[\pi]) \tag{5.62}$$

Tyto vztahy lze dále přeformulovat do rekurzivní podoby:

$$\alpha[q] = \bigoplus_{e \in \mathcal{E}: n[e]=q} \alpha[p[e]] \otimes w[e] \tag{5.63}$$

$$\beta[q] = \bigoplus_{e \in \mathcal{E}: p[e]=q} w[e] \otimes \beta[n[e]] \tag{5.64}$$

Následně tyto definice vzdáleností $\alpha[q]$ a $\beta[q]$ použijeme v algoritmu pro sestavení faktorového automatu $F(T)$ [69, 70]:

Vstup:

– Automat T

Výstup:

– Faktorový automat $F(T)$

Algoritmus vytvoření faktorového automatu:

1. Vytvoř $F(T)$ jako kopii T.
2. Spočítej vzdálenosti $\alpha[q]$ a $\beta[q]$ pro všechny stavy $q \in Q$.
3. Přidej do $F(T)$ stavy q_0 a q_1.
4. Nastav q_0 jako nový počáteční stav a q_1 jako jediný koncový stav s vahou $\bar{1}$.
5. Pro každý stav $q \in Q \setminus \{q_0, q_1\}$:
   - Přidej ϵ-přechod z q_0 do q s vahou $\alpha[q]$.
   - Přidej ϵ-přechod z q do q_1 s vahou $\beta[q]$.
6. Proveď odstranění ϵ-přechodů, determinizaci a minimalizaci $F(T)$.

5.2.3 Optimalizační algoritmy

Nyní jmenujme další speciální algoritmy, které nemění relaci mezi vstupními a výstupními řetězci transduceru nebo jazyk přijímaný akceptorem, nicméně mění jeho vnitřní strukturu z pohledu počtu stavů a přechodů tak, aby zpravidla bylo možné dosáhnout vyšší rychlosti a nebo nižších paměťových nároků při práci s těmito optimalizovanými automaty.

Odstranění ϵ-přechodů [93] vytváří ekvivalentní konečný automat $\mathrm{rmeps}\, T$ k automatu T, přičemž výsledný automat neobsahuje žádné přechody e, pro které platí $i[e] = o[e] = \epsilon$.

Determinizace [6] vytváří ekvivalentní konečný automat $\det T$ k automatu T, přičemž pro všechny uzly $q \in Q$ platí, že pro libovolný symbol $a \in \mathcal{A}$ existuje nejvýše jeden přechod e označený tímto symbolem na vstupní straně.

Algoritmus determinizace je založen na konstrukci *vážených podmnožin* p' množiny Q. Vážená podmnožina p' je množina párů $(q, x) \in Q \times \mathbb{K}$ [8, str. 25]. Jako $Q[p']$ označíme množinu stavů q z vážené podmnožiny p'. Potom $\mathcal{E}[Q[p']]$ reprezentuje množinu přechodů ze stavů $Q[p']$ a $i\left[\mathcal{E}\left[Q[p']\right]\right]$ množinu vstupních symbolů těchto přechodů. Stavy automatu po determinizaci mohou být identifikovány pomocí vážených podmnožin stavů původního automatu. Stav r automatu po determinizaci, který lze z počátečního stavu dosáhnout pomocí cesty π_r je popsán množinou párů $(q, x) \in Q \times \mathbb{K}$ takových, že q je z počátečního stavu dosažitelný pomocí cesty π_q a zároveň $i[\pi_r] = i[\pi_q]$ a váhy $\lambda(p[\pi_q]) \otimes w[\pi_q] = \lambda(p[\pi_r]) \otimes w[\pi_r] \otimes x$. Důležitým závěrem je fakt, že každý stav výstupního deterministického automatu odpovídá jedné vážené podmnožině p'.

Minimalizace [94] vytváří konečný automat $\min T$ s minimálním počtem stavů ekvivalentní k deterministickému automatu T.

5.2.4 Grafická reprezentace vážených konečných automatů

V rámci této práce budeme používat i grafickou reprezentaci vážených konečných automatů. Stavy jsou značeny kružnicemi, počáteční stav tučnou kružnicí, koncový stav dvojitou kružnicí. Uvnitř kružnic se nacházejí identifikátory stavů q, je-li $\rho(q) \neq \bar{0}$, tj. koncová váha ve stavu q je nenulová, pak je uvnitř kružnice uvedeno $q/\rho(q)$. Mezi stavy se nachází orientované přechody e reprezentované šipkou z počátečního stavu přechodu $p[e]$ do koncového stavu přechodu $n[e]$. Každý přechod e má přiřazeno ohodnocení ve tvaru $i[e] : o[e]/w[e]$, je-li $w[e] = \bar{1}$, pak pouze ve tvaru $i[e] : o[e]$. Ukázkou této grafické reprezentace může být automat zobrazený na obrázku 5.1 a dalších.

5.2.5 Speciální symboly ve vážených konečných automatech

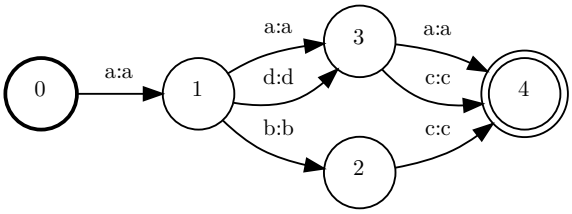
Jediným v kontextu konečných transducerů dosud zmíněným speciálním symbolem byl symbol ϵ, přičemž přechod v transduceru T_2 obsahující ϵ na svém vstupu může být při kompozici $T_1 \circ T_2$ realizován bez přijetí výstupního symbolu z odpovídajícího přechodu transduceru T_1. Poznamenejme, že ϵ-přechody obecně zvyšují neurčitost obsaženou v daném konečném transduceru, neboť při kompozici je nutné při prohledávání stavového prostoru expandovat větší množství uzlů. Použitá implementace vážených konečných transducerů OpenFST [91] navíc podporuje i další speciální symboly, které naopak umožňují kompaktnější reprezentaci a následně i rychlejší kompozici konečných automatů. Při popisu speciálních symbolů předpokládejme, že se uplatňují při kompozici transducerů $T_1 \circ T_2$ a že speciální symboly se vyskytují na vstupní straně transduceru T_2:

- ϵ symbol, který odpovídá prázdnému symbolu. Přechod označený na vstupu symbolem ϵ může být realizován bez přijetí odpovídajícího symbolu. Jedním z důsledků pak je, že posloupnosti symbolů $a\epsilon b\epsilon c$ a abc jsou si rovny. Přechody označené symbolem ϵ (tzv. ϵ-přechody) způsobují nedeterminismus automatu, neboť se mohou, ale nemusí realizovat při kompozici nebo při generování cest automatem. Pro odstranění ϵ-přechodů existuje algoritmus zmíněný v kapitole 5.2.3.
- σ symbol, který při kompozici může odpovídat libovolnému symbolu z $\Pi_2(T_1)$. σ-přechod tento symbol přijme a obsahuje-li tento přechod σ symbol i na svém výstupu, pak je přijatý symbol z T_1 kopírován na výstup kompozice $T_1 \circ T_2$.
- ρ symbol – přechod obsahující na vstupu tento symbol je realizován, pokud při kompozici není možné z daného uzlu pokračovat jinou cestou. ρ-přechod přijme odpovídající symbol z výstupu T_1 a obsahuje-li tento přechod ρ symbol i na výstupu, pak je přijatý symbol z T_1 kopírován na výstup kompozice $T_1 \circ T_2$.
- ϕ symbol je obdobou symbolu ρ, nicméně ϕ-přechod nepřijímá žádný symbol z výstupu T_1, realizuje se pouze pokud z daného uzlu nelze pokračovat jinou cestou.

Tabulka 5.1 shrnuje různé speciální symboly a dělí je do podskupin. První dělení se uplatní při kompozici $T_1 \circ T_2$ a hledání shody mezi symbolem na výstupu T_1 a speciálním symbolem na vstupu T_2 – jedná-li se o akceptující symbol, pak je při kompozici symbol z T_1 zpracován a pokračuje se dalším symbolem, v neakceptující variantě je symbol z T_1 znovu porovnáván s dalšími symboly na přechodech následujících za přechodem označeným speciálním symbolem.

Shoda při kompozici	Akceptuje symbol	Neakceptuje symbol
Shoduje se vždy	σ	ϵ
Pokud není jiná shoda	ρ	ϕ

Tabulka 5.1: Tabulka shrnující speciální symboly v kontextu vážených konečných automatů.

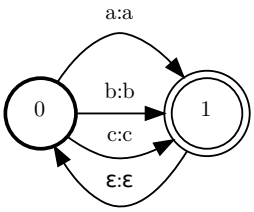


Obrázek 5.1: Vstupní akceptor T_1

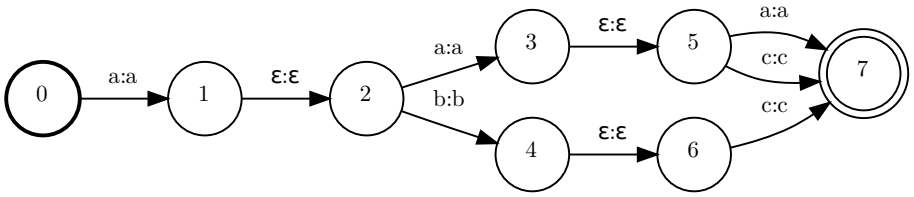
Druhé dělení je podle toho, kdy se přechod v T_2 označený na vstupní straně speciálním symbolem zrealizuje – pokud se symbol vždy shoduje, pak je přechod realizován při každé příležitosti, v opačném případě je přechod realizován pouze pokud ze stavu nevede žádný jiný přechod, který by bylo možné realizovat.

I přes to, že tyto speciální symboly mají široké použití v různých aplikacích vážených konečných automatů, např. v úloze jazykového modelování [40] nebo v úloze statistického překladu [87], příklady transducerové kompozice s využitím těchto speciálních symbolů v literatuře nejsou příliš časté. Uveďme zde tedy ukázkový vstupní transducer akceptující řetězce *aaa*, *aac*, *ada*, *adc* a *abc* (obrázek 5.1).

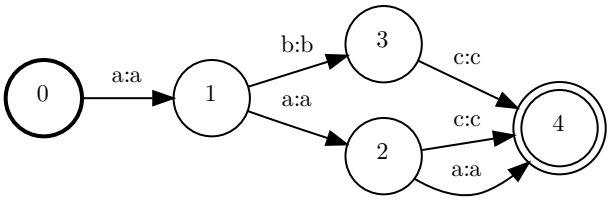
Přechod označený symbolem ϵ se může realizovat bez ohledu na vstupní symbol. V použité implementaci vážených konečných transducerů OpenFST [91] je používán v celé řadě algoritmů, např. pro konstrukci Kleeneho uzávěru. Na obrázku 5.2 je zobrazen akceptor T_2^ϵ, který přijímá libovolné řetězce složené ze znaků *a*, *b* a *c* o délce minimálně jeden znak. Jeho kompozicí s akceptorem T_1 získáme akceptor $T_1 \circ T_2^\epsilon$ (obrázek 5.3). Poněvadž přechod ze stavu 1 do stavu 0 transduceru T_2^ϵ má jako výstupní symbol opět ϵ, jsou všechny přechody v transduceru $T_1 \circ T_2^\epsilon$, které vznikly kompozicí s tímto přechodem opět označeny symbolem ϵ. Pro odstranění těchto přechodů lze použít algoritmus rmeps$(\cdot)$, jehož výsledek je vyobrazen na obrázku 5.4. Porovnáním 5.1 a 5.4 je vidět, že při kompozici byl odstraněn přechod v transduceru T_1 ze stavu 1 do stavu 3 označený symbolem *d*, protože není přijímán akceptorem T_2^ϵ.



Obrázek 5.2: Transducer T_2^ϵ se symbolem ϵ

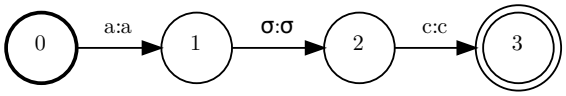


Obrázek 5.3: Transducer $T_1 \circ T_2^\epsilon$

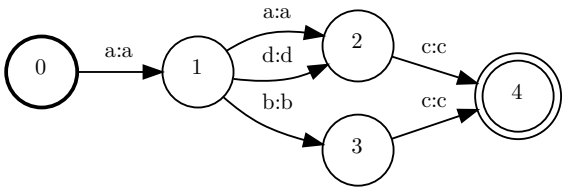


Obrázek 5.4: Transducer rmeps($T_1 \circ T_2^\epsilon$)

Symbol σ lze chápat jako zástupný symbol, který se shoduje s libovolným jiným symbolem v transduceru T_1. Transducer na obrázku 5.5 akceptuje řetězce $a\sigma c$, tj. všechny řetězce délky 3, které začínají symbolem a a končí symbolem c. Výsledná kompozice $T_1 \circ T_2^\sigma$ je zachycena na obrázku 5.6.

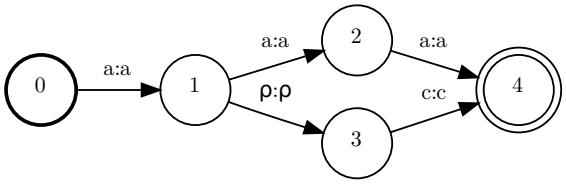


Obrázek 5.5: Transducer T_2^σ se symbolem σ

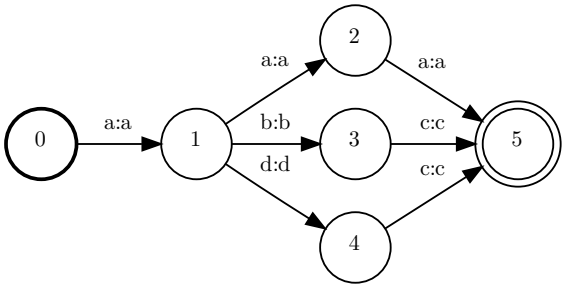


Obrázek 5.6: Transducer $T_1 \circ T_2^\sigma$

Symbol ρ je ústupovým symbolem, který se shoduje, pokud není nalezen žádný jiný přechod v T_1 z daného stavu, kterým by bylo možné při kompozici pokračovat. Transducer na obrázku 5.7 akceptuje buď řetězec aaa a nebo řetězec $a\rho c$, kterému odpovídá řetězec délky 3, kde první symbol je a, druhý symbol je libovolný symbol kromě a (neboť to je již obsaženo v řetězcu aaa) následovaný symbolem c.

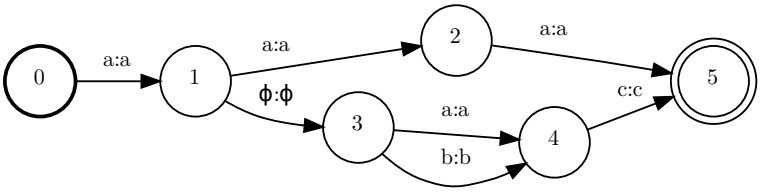


Obrázek 5.7: Transducer T_2^{ρ} se symbolem ρ

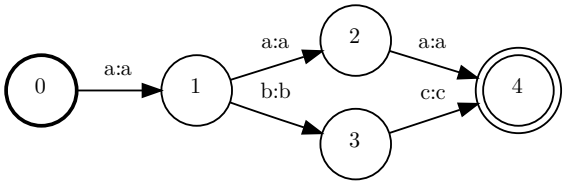


Obrázek 5.8: Transducer $T_1 \circ T_2^{\rho}$

Symbol ϕ je ústupovým symbolem a přechod takto označený se realizuje bez přijetí vstupního symbolu, pokud není nalezen žádný jiný přechod v T_1 z daného stavu, kterým by bylo možné při kompozici pokračovat. Akceptor na obrázku 5.10 nepřijímá řetězec *aac*, neboť přechod se symbolem ϕ se realizuje pouze pokud není jiné možnosti, avšak druhý symbol *a* je již obsažen na přechodu ze stavu 1 do stavu 2 transduceru T_2^{ϕ} a tudíž všechny cesty v T_1 začínající prefixem *aa* budou akceptovány cestou mezi stavy 0-1-2, ostatní cesty začínající prefixem *a* budou akceptovány cestou mezi stavy 0-1-3 (obrázek 5.9).



Obrázek 5.9: Transducer T_2^{ϕ} se symbolem ϕ



Obrázek 5.10: Transducer $T_1 \circ T_2^{\phi}$

V této práci jsou použity transducery obsahující na vstupu přechodu symbol σ, který slouží jako zástupný symbol za libovolný symbol z odpovídající výstupní abecedy $\mathcal{B}$ transduceru T_1. Díky tomu je možné použít jediný přechod označený tímto symbolem namísto $|\mathcal{B}|$ přechodů (v řešené úloze se může jednat o řádově až desetitisíce přechodů). Úspora z pohledu paměťového prostoru nutného pro reprezentaci T_2 i z pohledu výpočetních nároků při kompozici $T_1 \circ T_2$ je zřejmá.

Dále poznamenejme, že symbol ϕ je speciálním symbolem vhodným pro implementaci ústupového vyhlazování při práci s n-gramovými jazykovými modely reprezentovanými pomocí konečných automatů. U ústupového vyhlazování trigramového modelu je modelována pravděpodobnost $\mathrm{P}(w_n|w_{n-2}, w_{n-1})$ slova w_n za předpokladu, že tomuto slovu předchází historie w_{n-2}, w_{n-1}. Pokud ovšem trigram w_{n-2}, w_{n-1}, w_n při trénování jazykového modelu nebyl pozorován, je tato pravděpodobnost aproximována pravděpodobností bigramovou $b(w_n, w_{n-1})\mathrm{P}(w_n|w_{n-1})$, kde $b(w_n, w_{n-1})$ je ústupová váha. Symbol ϕ se uplatní právě při rozhodování, zda daná historie byla pozorována a tudíž je modelována trigramovou pravděpodobností, nebo je nutné aplikovat ústup realizovaný ϕ-přechodem s váhou $b(w_n, w_{n-1})$ na historii bigramovou [40, 95].

5.3 Racionální jádrové funkce

Klasická klasifikační úloha popsaná v části 5.1 je formulovaná pro trénovací množinu $\mathcal{T} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$, kde $\mathbf{x}_i \in \mathbb{R}^n$ je n-rozměrný příznakový vektor. Obecně však klasifikovaný obraz nemusí být reprezentován příznakovým vektorem, ale některou komplexnější strukturou. V oblasti rozpoznávání a porozumění řeči, ale třeba i v oblasti zpracování textu nebo výpočetní biologii, jsou obrazy často reprezentovány pomocí konečných automatů (akceptorů, případně transducerů). To umožňuje výhodnou reprezentaci více hypotéz spolu s jejich aposteriorními pravděpodobnostmi.

Klasifikátory SVM díky využití jádrových funkcí umožňují efektivní optimalizaci klasifikátoru ve vysokodimenzionálním prostoru $\mathcal{H}$. V této kapitole jsou popsány jádrové funkce založené na konečných transducerech – *racionální jádrové funkce* – které umožňují vyčíslení jádrové funkce mezi dvěma konečnými automaty (akceptory či transducery) [10]. Tyto racionální jádrové funkce využívají operaci kompozice konečných vážených transducerů pro efektivní výpočet hodnoty jádrové funkce.

Racionální jádrová funkce je pak jádrová funkce definovaná nad množinami vážených konečných transducerů. Jádrová funkce K nad $\mathcal{A}^* \times \mathcal{B}^*$ se nazývá racionální, pokud existuje vážený transducer $S = (\mathcal{A}, \mathcal{B}, \mathcal{Q}, I, \mathcal{F}, \mathcal{E}, \lambda, \rho)$ nad polookruhem $\mathbb{K}$ a funkce $\psi : \mathbb{K} \to \mathbb{R}$ taková, že pro všechna $x \in \mathcal{A}^*$ a $y \in \mathcal{B}^*$ platí:

$$K(x, y) = \psi\left(S(x, y)\right) \tag{5.65}$$

Jádrová funkce K je pak definována dvojicí (ψ, S), kde ψ je libovolná funkce zobrazující $\mathbb{K}$ do oboru reálných čísel. Racionální jádrové funkce pak mohou být rozšířeny na jádrové funkce nad váženými konečnými automaty [10].

Nechť A, B jsou vážené konečné akceptory nad polookruhem $\mathbb{K}$ a nad abecedou $\mathcal{A}$, respektive $\mathcal{B}$. Jádrová funkce $K(A, B)$:

$$K(A, B) = \psi\left(\bigoplus_{(x,y)\in\mathcal{A}^*\times\mathcal{B}^*} A(x) \otimes S(x, y) \otimes B(y)\right) \tag{5.66}$$

je pak definována pro všechna A a B taková, že $\oplus$-součet

$$\bigoplus_{(x,y)\in\mathcal{A}^*\times\mathcal{B}^*} A(x) \otimes S(x, y) \otimes B(y) \tag{5.67}$$

je konečný a náleží do $\mathbb{K}$. $\oplus$-součet je vždy konečný a náleží do $\mathbb{K}$, pokud A a B jsou acyklické vážené konečné akceptory, neboť $\oplus$-součet je prováděn přes konečnou množinu vstupně výstupních dvojic. Suma je rovněž definována pro všechny vážené automaty nad *uzavřeným polookruhem*[1], např. tropický polookruh.

Pro pravděpodobnostní polookruhy je suma dobře definována, pokud A, B a S reprezentují pravděpodobnostní rozdělení. Je-li $K(A, B)$ definováno, pak rovnice (5.66) může být ekvivalentně zapsána jako:

$$K(A, B) = \psi\left(w[A \circ S \circ B]\right) \tag{5.68}$$

5.3.1 Pozitivně definitní symetrické racionální jádrové funkce

Pro použití v rámci SVM klasifikátorů je nutné uvažovat jádrové funkce splňující Mercerovu podmínku (kapitola 5.1.4). Tato podkapitola dá obecný návrh, jak konstruovat pozitivně definitní symetrické racionální jádrové funkce.

Předpokládejme, že funkce $\psi : (\mathcal{K}, \oplus, \otimes, \bar{0}, \bar{1}) \to (\mathbb{R}, +, \times, 0, 1)$ je spojitý morfismus. Nyní předpokládejme, že $T = (\mathcal{A}, \mathcal{B}, \mathcal{Q}, I, \mathcal{F}, \mathcal{E}, \lambda, \rho)$ je vážený konečný transducer definovaný nad polookruhem $(\mathcal{K}, \oplus, \otimes, \bar{0}, \bar{1})$. Předpokládejme, že vážený transducer $T \circ T^{-1}$ je regulovaný[2], potom $(\psi, T \circ T^{-1})$ definuje pozitivně definitní symetrický racionální jádrovou funkci nad $\mathcal{A}^* \times \mathcal{A}^*$.

Důkaz: Označme jako S kompozici $T \circ T^{-1}$. Nechť K je racionální jádrová funkce definovaná pomocí S. Z definice kompozice lze psát:

$$K(x, y) = \psi(S(x, y)) = \psi\left(\bigoplus_{z \in \mathcal{B}*} T(x, z) \otimes T^{-1}(z, y)\right) \tag{5.69}$$

pro všechna $x, y \in \mathcal{A}^*$. Protože ψ je spojitý morfismus, pak pro všechna $x, y \in \mathcal{A}^*$ platí:

$$K(x, y) = \psi(S(x, y)) = \sum_{z \in \mathcal{B}*} \psi\left[T(x, z)\right] \cdot \psi\left[T^{-1}(z, y)\right] \tag{5.70}$$

Pro všechna $n \in \mathbb{N}$ a $x, y \in \mathcal{A}^*$ definujme $K_n(x, y)$ jako:

$$K_n(x, y) = \sum_{|z| \leq n} \psi\left[T(x, z)\right] \cdot \psi\left[T^{-1}(z, y)\right] \tag{5.71}$$

Suma je realizována přes všechny řetězce $z \in \mathcal{B}^*$ s délkou menší nebo rovnou n. Pro všechna $l \geq 1$ a pro $x_1, \ldots, x_l \in \mathcal{A}^*$ definujme matici $\mathbf{M}_n$ jako $\mathbf{M}_n = [K_n(x_i, x_j)]$, $1 \leq i \leq l$, $1 \leq j \leq l$.

[1] Polookruh $\mathbb{K} = (\mathcal{K}, \oplus, \otimes, \bar{0}, \bar{1})$ je uzavřený, pokud:
- pro všechna $a \in \mathbb{K}$ je nekonečná suma $\bigoplus_{n=0}^{\infty} a^n$ dobře definovaná a náleží do $\mathbb{K}$
- a pro tyto nekonečné sumy platí asociativní, komutativní a distributivní zákon. [96]

[2] Transducer T je regulovaný, pokud váha $T(x, y)$ přiřazená transducerem libovolnému páru vstupních a výstupních řetězců (x, y) je dobře definována a náleží do $\mathbb{K}$. Vážený konečný transducer bez ϵ-cyklů je regulovaný [91].

Nechť $z_1, z_2, \ldots, z_m, z_i \in \mathcal{B}^*$ je libovolně uspořádaná posloupnost řetězců délky menší nebo rovné n. Definujme matici $\mathbf{A}$ jako:

$$\mathbf{A} = \psi\left[T(x_i, z_j)\right], 1 \leq i \leq l, 1 \leq j \leq m \tag{5.72}$$

Potom z definice funkce K_n je $\mathbf{M}_n = \mathbf{A}\mathbf{A}^\intercal$. Vlastní čísla $\mathbf{A}\mathbf{A}^\intercal$ jsou nezáporná pro libovolnou matici $\mathbf{A}$, odtud K_n je pozitivně definitní symetrická jádrová funkce. Protože K je bodová limita posloupnosti K_n:

$$K(x, y) = \lim_{n\to\infty} K_n(x, y) \tag{5.73}$$

pak i K je pozitivně definitní symetrická jádrová funkce [10]. □

Při použití racionálních jádrových funkcí nedochází k explicitnímu vyčíslení příznakového vektoru $\mathbf{x}$ příslušejícího danému příkladu z trénovací množiny případně příslušejícího novému příkladu určenému ke klasifikaci. Není proto možné provádět normalizaci příznakových vektorů ve vstupním prostoru. S využitím racionálních jádrových funkcí však lze s výhodou využít normalizaci v příznakovém prostoru $\mathcal{H}$ dle rovnice (5.44).

5.3.2 n-gramové jádrové funkce

Současné modely pro porozumění řeči využívají pro trénování a dekódování významové reprezentace zpravidla nejlepší hypotézy, případně ohodnocený seznam n-nejlepších hypotéz odvozený ze slovní mřížky přiřazené vstupní promluvě, např. [55, 59, 65]. Racionální jádrová funkce vyjadřuje podobnost mezi dvěma váženými konečnými automaty. Je proto možné při klasifikaci využít přímo tuto jádrovou funkci namísto odvozování explicitní reprezentace příznakových vektorů a jejich použití při trénování klasifikátoru. Racionální jádrová funkce umožňuje použít např. celou slovní mřížku jako reprezentaci prvku z trénovací množiny a následně provádět klasifikaci nikoli příznakových vektorů, ale přímo mřížek.

Uvažujme mřížku A, která reprezentuje pravděpodobnostní distribuci $\mathrm{P}_A(a)$ nad všemi řetězci $a \in \mathcal{A}^*$. Označme počet výskytů podřetězce x v řetězci a jako $\mathrm{cnt}(a, x)$. Střední počet výskytů všech podřetězců x v automatu A s pravděpodobnostním rozložením $\mathrm{P}_A(a)$ lze získat jako:

$$\mathrm{cnt}(A, x) = \sum_{a\in\mathcal{A}^*} \mathrm{P}_A(a)\,\mathrm{cnt}(a, x) \tag{5.74}$$

Dvě mřížky mohou být považovány za podobné, pokud střední počty výskytů společných podřetězců jsou dostatečně vysoké. Definujme tedy jádrovou funkci k_n pro dvě mřížky A a B nad společnou abecedou $\mathcal{A}$:

$$k_n(A, B) = \sum_{\substack{x\in\mathcal{A}^* \\ |x|=n}} \mathrm{cnt}(A, x)\,\mathrm{cnt}(B, x) \tag{5.75}$$

Jádrová funkce k_n je pozitivně definitní symetrická jádrová funkce typu $T \circ T^{-1}$. Hodnota $\mathrm{cnt}(A, x)$ pro $|x| = n$ může být efektivně získána pomocí transduceru T_n definovaného jako [10]:

$$T_n = (\mathcal{A} \times \{\epsilon\})^* \otimes (\bigoplus_{x\in\mathcal{A}} \{x\} \times \{x\})^n \otimes (\mathcal{A} \times \{\epsilon\})^* \tag{5.76}$$

Funkce k_n pak může být psána rovněž jako:

$$\begin{aligned} k_n(A,B) &= \psi\left(w\left[(A \circ T_n) \circ (T_n^{-1} \circ B)\right]\right) \\ &= \psi\left(w\left[A \circ (T_n \circ T_n^{-1}) \circ B\right]\right) \end{aligned} \tag{5.77}$$

Jedná se o racionální jádrovou funkci s odpovídajícím transducerem $T_n \circ T_n^{-1}$, a tedy k_n je pozitivně definitní symetrická jádrová funkci. Dále využijeme poznatku, že součet dvou jádrových funkcí k_n a k_m je rovněž pozitivně definitní symetrická jádrová funkce. Pak definujeme *n-gramovou racionální jádrovou funkci* K_n jako součet všech funkcí k_m pro $1 \leq m \leq n$:

$$K_n = \sum_{m=1}^{n} k_m \tag{5.78}$$

V praxi jsou automaty A, B, T_n často definovány nad logaritmickým polookruhem namísto pravděpodobnostního polookruhu, neboť numerická implementace pak vede na numericky stabilní algoritmus.[3] Poněvadž pravděpodobnostní a logaritmický polookruh jsou izomorfní, lze pro zobrazení hodnot z logaritmického polookruhu do množiny reálných čísel použít následující tvar funkce ψ:

$$\psi(w[A]) = \exp(-w[A]) \tag{5.79}$$

Poznamenejme, že n-gramovou racionální jádrovou funkci K_n, definovanou v rovnici (5.78), lze modifikovat tak, že funkce bude vyčíslovat střední počty výskytů společných n-gramů řádu n až m kde $1 \leq n \leq m$:

$$K_{n,m} = \sum_{i=n}^{m} k_i \tag{5.80}$$

Tato racionální jádrová funkce je definována transducerem $T_{n,m} \circ T_{n,m}^{-1}$, kde $T_{n,m}$ je dáno jako:

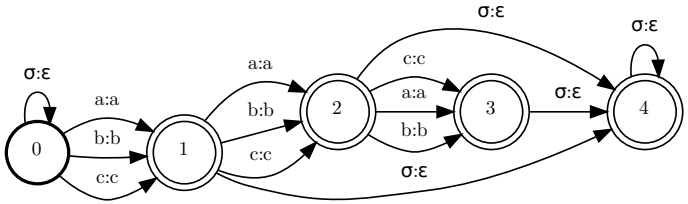
$$T_{n,m} = \bigoplus_{i=n}^{m} T_i \tag{5.81}$$

a T_i je definováno v rovnici (5.76). Váhy cest v kompozici $\psi(A \circ T_{n,m})$ převedené do prostoru reálných čísel funkcí $\psi(\cdot)$ pak vyjadřují střední počet výskytů jednotlivých n-gramů délky n až m v akceptoru $\Pi_2(A)$ (tj. mezi řetězci výstupních symbolů automatu A).

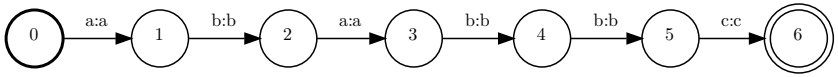
Na obrázku 5.11 je zobrazen transducer $T = T_{1,3} = \bigoplus_{i=1}^{3} T_i$ reprezentující racionální jádrovou funkci nad abecedou *abc*. Tento transducer generuje ze vstupních řetězců všechny n-gramy délky 1, 2 a 3. Ukázkový vstup A odpovídající řetězci *ababbc* je zobrazen na obrázku 5.12. Jejich kompozicí je získán transducer $A \circ T$, který je zobrazen na obrázku 5.13. Předpokládejme nyní druhý ukázkový vstup B odpovídající řetězci *abc* (bez vyobrazení). Obdobně i pro něj získáme kompozici $T^{-1} \circ B$ (obrázek 5.14).

Výslednou kompozici $(A \circ T) \circ (T^{-1} \circ B)$ vidíme na obrázku 5.15. Z obrázku vyplývá, že transducery A a B sdílí následující n-gramy: $2\times a$, $3\times b$, $1\times c$, $2\times ab$ a $1\times bc$. Odtud hodnota n-gramové racionální jádrové funkce mezi akceptory A a B dané transducerem

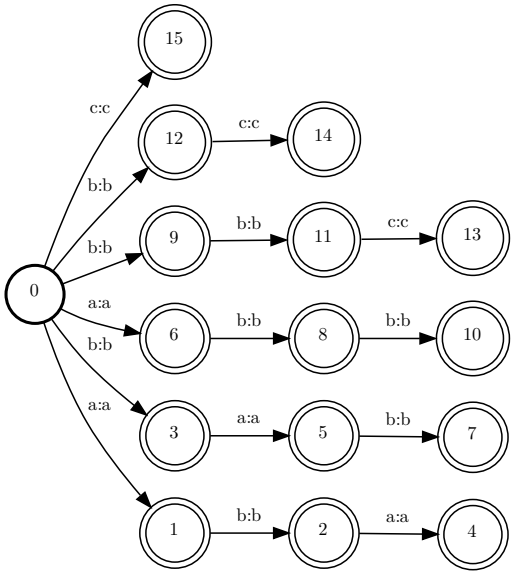
[3] Algoritmus determinizace nad pravděpodobnostním polookruhem používá numerické dělení. Při dělení malými čísly dochází k numerické nestabilitě způsobené omezeným počtem platných cifer při reprezentaci reálných čísel v paměti počítače.



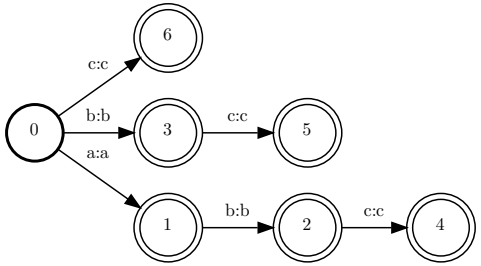
Obrázek 5.11: Transducer $T = T_{1,3}$ definující n-gramovou racionální jádrovou funkci. Symbol σ může při kompozici odpovídat libovolnému symbolu.



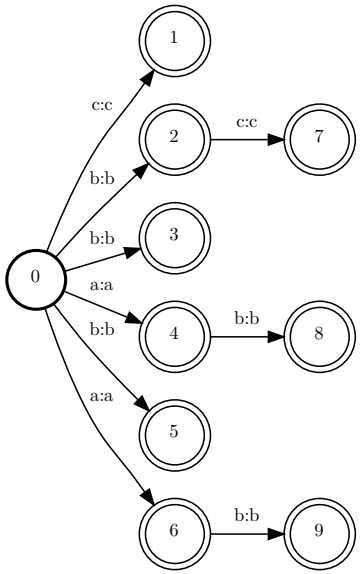
Obrázek 5.12: Ukázkový vstup A – řetězec *ababbc*.



Obrázek 5.13: Kompozice $\Pi_2(A \circ T)$, pro lepší názornost byla provedena projekce na výstupní symboly a odstranění ϵ-přechodů.



Obrázek 5.14: Kompozice $\Pi_1(T^{-1} \circ B)$, pro lepší názornost byla provedena projekce na vstupní symboly a odstranění ϵ-přechodů.



Obrázek 5.15: Kompozice $(A \circ T) \circ (T^{-1} \circ B)$, bylo provedeno odstranění ϵ-přechodů.

$T = T_{1,3}$ z obrázku 5.11 je $K(A,B) = 2+3+1+2+1 = 9$. Dále bez detailnějšího výpočtu uveďme hodnotu $K(A,A) = 25$ a hodnotu $K(B,B) = 6$. Potom normalizovaná racionální jádrová funkce nabývá hodnoty $\tilde{K}(A,B) = \frac{K(A,B)}{\sqrt{K(A,A)\cdot K(B,B)}} = \frac{9}{\sqrt{25\cdot 6}} \approx 0.735$.

5.3.3 Příklady další racionálních jádrových funkcí

Jedním z dalších typů racionálních jádrových funkcí je tzv. *mismatch string kernel* [10, 97]. Definujme Σ jako konečnou abecedu, a řetězce $z_1, z_2 \in \Sigma^*$ předpokládejme o stejné délce, tj. $|z_1| = |z_2|$. Označme celkový počet symbolů, ve kterých se z_1 a z_2 liší, jako $d(z_1, z_2)$. Potom pro libovolné $m \in \mathbb{N}$ definujme omezenou vzdálenost d_m mezi dvěma řetězci stejné délky jako:

$$d_m(z_1, z_2) = \begin{cases} 1 & \text{pokud } d(z_1, z_2) \leq m \\ 0 & \text{jinak} \end{cases} \tag{5.82}$$

Definujme množinu všech faktorů řetězce x délky k jako:

$$\mathcal{F}_k(x) = z : x \in \Sigma^* z \Sigma^*, |z| = k \tag{5.83}$$

Potom pro libovolné $k, m \in \mathbb{N}, m \leq k$ je (k,m)-mismatch string kernel $K_{(k,m)}$ funkce definovaná nad dvěma posloupnostmi $x, y \in \Sigma^*$:

$$K_{(k,m)} = \sum_{\substack{z_1 \in \mathcal{F}_k(x) \\ z_2 \in \mathcal{F}_k(y) \\ z \in \Sigma^k}} d_m(z_1, z) d_m(z, z_2) \tag{5.84}$$

Tento typ racionálních jádrových funkcí byl poprvé použit pro klasifikaci proteinů v práci [97]. Její výhodou je, že lze s výhodou využít líné implementace kompozičních algoritmů nad konečnými automaty a tím pádem vytvářet transducer odpovídající racionální jádrové funkci až podle potřeby při kompozici transducerů.

Jiným typem racionálních jádrových funkcí nad řetězci jsou tzv. *Haussleroy konvoluční jádrové funkce* [10, 98]. Transducer odpovídající této jádrové funkci je definován pro $0 \leq \gamma \leq 1$ jako:

$$K_H = (1-\gamma)\left[K_2 \oplus (\gamma K_1 \oplus K_2)^*\right] \tag{5.85}$$

kde $K_1(x,y)$ je transducer nad pravděpodobnostním polookruhem modelující pravděpodobnost záměny posloupnosti symbolů x za y a $K_2(z,\epsilon)$ je transducer nad pravděpodobnostním polookruhem modelující pravděpodobnost vložení posloupnosti symbolů z. Pro úplnost dodejme, že Haussleroy jádrové funkce lze sestavit i pro obecnější struktury, jako jsou stromy nebo grafy [98].

5.3.4 Algoritmický výpočet racionální jádrové funkce

Výpočet racionální jádrové funkce $K(A,B)$ pro dva libovolné acyklické konečné automaty je založen na kompozici automatů $A \circ S \circ B$ a na obecném algoritmu nejkratší vzdálenosti v polookruhu $\mathbb{K}$ pro výpočet $\oplus$-součtu vah všech úspěšných cest kompozicí automatů. Poznamenejme, že pro pozitivně definitní symetrické racionální jádrové funkce je transducer S ve tvaru $S = T \circ T^{-1}$.

Při daném acyklickém automatu M je nejkratší vzdálenost ze stavu q do množiny koncových stavů $\mathcal{F}$ definována jako $\oplus$-součet všech cest z q do $\mathcal{F}$, viz rovnice (5.62). Je-li M acyklický automat, pak složitost algoritmu je lineární, tj. $\mathcal{O}(|Q| + (S_\oplus + S_\otimes)|E|)$, kde $S_\oplus$ a $S_\otimes$ je maximální čas potřebný k výpočtu $\oplus$, resp. $\otimes$ [96].

Algoritmus výpočtu $K(A, B)$ se pak skládá z následujících kroků [10]:

1. Sestavení transduceru $N = A \circ S \circ B$
2. Výpočet $w[N]$ jako nejkratší vzdálenosti z počátečních stavů I_N do koncových stavů $\mathcal{F}_N$.
3. Výpočet $\psi(w[N])$

Jsou-li A a B acyklické, pak složitost algoritmu nejkratší vzdálenosti je lineární a celková složitost algoritmu je $\mathcal{O}(|S||A||B| + \Phi)$, kde $|S|$, $|A|$ a $|B|$ jsou velikosti odpovídajících automatů a Φ je nejhorší složitost výpočtu $\psi(x), x \in \mathbb{K}$.

Pro n-gramové jádrové funkce platí, že velikost transduceru $|S|$ je omezena $\mathcal{O}(n|\mathcal{A}|)$, ale v praxi může být díky využití líné implementace kompozice (lazy composition) a symbolu σ reprezentujícího všechny prvky vstupní (resp. výstupní) abecedy omezena na $\mathcal{O}(n)$. A protože složitost kompozice automatů je kvadratická [6] a obecný algoritmus pro určení nejkratší vzdálenosti je lineární pro acyklické grafy [96], pak nejhorší časová složitost algoritmu při použití nad mřížkami je $\mathcal{O}(n^2|A||B|)$.

5.4 Stochastické bezkontextové gramatiky

Stochastické bezkontextové gramatiky jsou velmi používaný matematický nástroj pro modelování strukturálních vlastností přirozeného jazyka. Jsou ideálním nástrojem pro zápis expertních znalostí popisujících např. jazykový model nebo model porozumění. Formálně můžeme *bezkontextovou gramatiku* definovat jako čtveřici $G = (\mathcal{N}, \Sigma, \mathcal{R}, S)$ [3], kde:

- $\mathcal{N}$ je množina neterminálních symbolů,
- Σ je množina terminálních symbolů, přičemž $\mathcal{N} \cap \Sigma = \emptyset$,
- $\mathcal{R}$ je množina pravidel, přičemž každé pravidlo je ve tvaru $A \to \beta$, kde $A \in \mathcal{N}$ a β je řetězec libovolné délky složený z terminálních a neterminálních symbolů, formálně $\beta \in (\Sigma \cup \mathcal{N})^*$, kde symbol * značí Kleeneho uzávěr, tj. množinu všech řetězců libovolné délky nad množinou $(\Sigma \cup \mathcal{N})$.
- $S \in \Sigma$ je startovací symbol.

Jazykem $\mathcal{L}_G$ definovaným gramatikou G rozumíme obecně nekonečnou množinu řetězců Σ^*, kterou získáme postupným odvozením ze symbolu S. Formálně, pokud $A \to \beta$ je pravidlo z $\mathcal{R}$ a α a γ jsou řetězce z množiny $(\Sigma \cup \mathcal{N})^*$, potom říkáme, že řetězec $\alpha A \gamma$ *přímo generuje* řetězec $\alpha\beta\gamma$, nebo že $\alpha A \gamma \Rightarrow \alpha\beta\gamma$ je *přímá derivace*.

Pokud platí, že $\alpha_1, \alpha_2, \ldots \alpha_m$ jsou řetězce $(\Sigma \cup \mathcal{N})^*$ a $m \geq 1$ takové, že

$$\alpha_1 \Rightarrow \alpha_2, \alpha_2 \Rightarrow \alpha_3, \ldots \alpha_{m-1} \Rightarrow \alpha_m \tag{5.86}$$

pak říkáme, že α_1 *generuje* α_m, nebo že $\alpha_1 \Rightarrow^* \alpha_m$ je *derivací*.

Definujme nyní jazyk $\mathcal{L}_G$ generovaný gramatikou G jako množinu řetězců terminálních symbolů, kterou generuje ze startovací symbol S gramatiky G.

$$\mathcal{L}_G = \{w | w \in \Sigma^* \ \wedge \ S \Rightarrow^* w\} \tag{5.87}$$

Stochastické bezkontextové gramatiky jsou rozšířením bezkontextových gramatik – umožňují ke každému pravidlu přiřadit zároveň i podmíněnou pravděpodobnost použití tohoto pravidla. Formálně tento typ gramatiky opět definujeme jako čtveřici $G = (\mathcal{N}, \Sigma, \mathcal{R}, S)$, kde $\mathcal{N}$, Σ a S mají stejný význam jako u bezkontextových gramatik.

Množina $\mathcal{R}$ je tentokráte množinou pravidel ve tvaru $A \rightarrow \beta \ [p]$, kde $\beta \in (\Sigma \cup \mathcal{N})^*$ a $0 \leq p \leq 1$ je podmíněná pravděpodobnost vyjadřující pravděpodobnost expanze neterminálního symbolu A na řetězec β za podmínky výskytu neterminálu A v derivačním stromu:

$$\mathrm{P}(A \rightarrow \beta | A) = p \tag{5.88}$$

přičemž platí $\sum_\beta \mathrm{P}(A \rightarrow \beta | A) = 1$.

Předpokládejme, že pro derivaci $S \Rightarrow^* W$ lze použít n přímých derivací

$$S = \alpha_0 \Rightarrow \alpha_1, \alpha_1 \Rightarrow \alpha_2, \ldots \alpha_{n-1} \Rightarrow \alpha_n = W \tag{5.89}$$

přičemž každá přímá derivace $\alpha_{i-1} \Rightarrow \alpha_i$ vznikla aplikováním pravidla $A_i \rightarrow \beta_i \ [p_i]$. Definujme pravděpodobnost přímé derivace jako

$$\mathrm{P}(\alpha_{i-1} \Rightarrow \alpha_i) = p_i \tag{5.90}$$

potom pravděpodobnost této derivace $S \Rightarrow^* W$ definujeme jako

$$\mathrm{P}(S \Rightarrow^* W) = \prod_{i=1}^{n} \mathrm{P}(\alpha_{i-1} \Rightarrow \alpha_i) = \prod_{i=1}^{n} p_i \tag{5.91}$$

(Stochastické) bezkontextové gramatiky jsou vhodným nástrojem pro zanesení expertních znalostí do jazykového modelu pro rozpoznávání řeči v rámci modulu statistického rozpoznávání řeči. Problematické však může být přímé expertní určení konkrétních pravděpodobností přiřazených jednotlivým pravidlům. Řešení může spočívat v jejich odhadu z trénovacích dat [99, 100].

Zmiňme se krátce ještě o modifikaci bezkontextových gramatik – o *lexikalizovaných gramatikách*. Citujme zde definici uvedenou v práci [101]:

Gramatika se nazývá lexikalizovaná, pokud se skládá z

- konečné množiny elementárních pravidel konečné velikosti a každé z pravidel obsahuje nenulový počet lexikálních jednotek,
- a konečné množiny operací pro vytváření odvozených struktur s využitím těchto pravidel.

Jako lexikální jednotku nazývejme sekvenci písmen, číslic a dalších specifických znaků používaným pro označení určitého pojmu. V přirozených jazycích je lexikální jednotka reprezentována zpravidla slovem ze slovníku, obvykle ve formě substantiva nebo substantivního spojení [102].

Dodejme, že bezkontextové gramatiky obecně lexikalizované nejsou. Omezíme-li však pravidla na tvar $A \rightarrow a\beta$, kde A je neterminální symbol, a terminální symbol a β je buď prázdná množina nebo konečná množina terminálních a neterminálních symbolů, pak již můžeme mluvit o lexikalizované gramatice, neboť každé pravidlo obsahuje lexikální jednotku reprezentovanou terminálním symbolem a. Další z možných zástupců lexikalizovaných modelů je popsán například v práci [103].

5.5 n-gramové jazykové modely

Při návrhu jazykového modelu s využitím statistického přístupu se používají metody, které umožňují trénování parametrů pravděpodobnostního rozdělení $\mathrm{P}(\mathsf{U})$ z trénovacích dat. Poznamenejme, že v této práci budou použity jak slovní, tak fonémové jazykové modely. Proto místo standardního zápisu pravděpodobnosti přiřazené jazykovým modelem $\mathrm{P}(\mathsf{W})$ (W – *words*) budeme používat $\mathrm{P}(\mathsf{U})$ (U – *utterance*).

Trénovací data jsou v tomto případě reprezentována rozsáhlým množstvím textových dat. V oblasti hlasových dialogových systémů je nutné pro každou cílovou aplikaci se specifickým jazykovým modelem nalézt způsob, jak získat dostatečné množství trénovacích dat vhodných pro jazykový model zajišťující dostatečné pokrytí jak z pohledu rozpoznávacího slovníku tak i mezislovních statistik. V principu je možné použít buď data získaná z *dialogů člověk-člověk* nebo *simulovaných dialogů*.

Nevýhodnou dialogů člověk-člověk je poněkud odlišná mluva od cílového hlasového dialogového systému používajícího automatické rozpoznávání řeči. Především, v dialozích člověk-člověk nedochází tak často k chybnému porozumění druhé straně dialogu a proto v těchto dialozích téměř chybí pokrytí případů zotavení z chyb, které je však nutné pro dialogové systému s automatickým rozpoznáváním řeči.

Oproti tomu při použití simulovaných dialogů je možné simulovat i chyby automatického rozpoznávání řeči. V tomto případě se velmi často používá přístup tzv. Wizard-of-Oz [56, 104, 105]. Zde roli hlasového agenta hraje člověk, který s uživatelem komunikuje prostřednictvím akustického kanálu, do kterého jsou vloženy systémy automatického rozpoznávání a syntézy řeči. Tímto jsou do hlasového dialogu zaneseny chyby způsobené těmito systémy a výsledné řečové záznamy obsahují i příklady zotavení z chyb.

V praxi nejčastěji používaným typem statistických jazykových modelů jsou tzv. *n-gramové jazykové modely*, které modelují pravděpodobnost následující jednotky podmíněnou historií $n-1$ předcházejících jednotek [3, 30]. Předpokládejme, že náhodná proměnná U se skládá z N dílčích náhodných proměnných odpovídajících N samostatným jednotkám, tj. $\mathsf{U} = \mathsf{U}_1, \mathsf{U}_2, \ldots \mathsf{U}_N$. Tyto jednotky mohou být definovány jako slova, ale také jako fonémy, nebo slabiky. V následujícím výkladu budeme předpokládat, že U_i odpovídá i-tému slovu, nicméně tytéž vztahy platí i v případě, kdy bychom modelovali pravděpodobnost nikoli slovních, ale například fonémových posloupností U.

Chceme-li nyní vyčíslit pravděpodobnost N po sobě jdoucích slov $w_i,\ i = 1, 2, \ldots N$, lze s použitím řetězového pravidla psát:

$$\begin{aligned}\mathrm{P}(\mathsf{U}) &= \mathrm{P}(\mathsf{U}_1 = w_1, \mathsf{U}_2 = w_2, \ldots \mathsf{U}_N = w_N) \\ &= \mathrm{P}(\mathsf{U}_1, \mathsf{U}_2, \ldots \mathsf{U}_N) \\ &= \mathrm{P}(\mathsf{U}_1)\mathrm{P}(\mathsf{U}_2|\mathsf{U}_1)\mathrm{P}(\mathsf{U}_3|\mathsf{U}_2, \mathsf{U}_1) \ldots \mathrm{P}(\mathsf{U}_N|\mathsf{U}_{N-1} \ldots \mathsf{U}_2\mathsf{U}_1) \\ &= \prod_{k=1}^{N} \mathrm{P}(\mathsf{U}_k|\mathsf{U}_1^{k-1})\end{aligned} \tag{5.92}$$

Tímto jsme vyjádřili pravděpodobnost $\mathrm{P}(\mathsf{U})$ jako součin podmíněných pravděpodobností jednotlivých slov. Každá podmíněná pravděpodobnost vyjadřuje pravděpodobnost výskytu slova w_k za podmínky, že mu předcházela slova w_1^{k-1}. Namísto modelování pravděpodobnosti $\mathrm{P}(\mathsf{U}_k|\mathsf{U}_1^{k-1})$ se použije tzv. n-gramová aproximace, kdy se předpokládá, že pravděpodobnost výskytu každého slova je podmíněna pouze předchozími $n-1$ posledními slovy:

$$\begin{aligned}\mathrm{P}(\mathsf{U}_k|\mathsf{U}_1^{k-1}) &\approx \mathrm{P}(\mathsf{U}_k|\mathsf{U}_{k-n+1}^{k-1}) \\ \mathrm{P}(\mathsf{U}) &\approx \prod_{k=1}^{N} \mathrm{P}(\mathsf{U}_k|\mathsf{U}_{k-n+1}^{k-1})\end{aligned} \tag{5.93}$$

Parametry n-gramového jazykového modelu se nejčastěji odhadují pomocí metody maximální věrohodnosti. Jako n-gram označme posloupnost n po sobě jdoucích slov $w_1 w_2 \ldots w_n$ a počet výskytů určitého n-gramu v trénovacím korpusu $\mathcal{T}$ jako $\mathrm{cnt}(\mathcal{T}, w_1 w_2 \ldots w_n)$. Pravděpodobnost výskytu slova w_k podmíněnou $n-1$ předcházejícími slovy $w_{k-n+1} w_{k-n+2} \ldots w_{k-1}$ odhadneme jako poměr četnosti výskytu daného n-gramu a jeho historie délky $n-1$:

$$\begin{aligned}\mathrm{P}(\mathsf{U}_k|\mathsf{U}_{k-n+1}^{k-1}) &= \frac{\mathrm{cnt}(\mathcal{T}, w_{k-n+1} w_{k-n+2} \ldots w_{k-1} w_k)}{\mathrm{cnt}(\mathcal{T}, w_{k-n+1} w_{k-n+2} \ldots w_{k-1})} \\ &= \frac{\mathrm{cnt}(\mathcal{T}, w_{k-n+1} w_{k-n+2} \ldots w_{k-1} w_k)}{\sum_{w_k} \mathrm{cnt}(\mathcal{T}, w_{k-n+1} w_{k-n+2} \ldots w_{k-1} w_k)}\end{aligned} \tag{5.94}$$

Zabývejme se nyní volbou parametru n. Předpokládejme, že v úloze rozpoznávání řeči pracujeme s množinou slov – rozpoznávacím slovníkem $\mathcal{V}$. Označme $|\mathcal{V}|$ počet slov ve slovníku, potom počet všech možných n-tic nad tímto slovníkem je $|\mathcal{V}|^n$. Množství dat nutných k robustnímu odhadu pravděpodobností všech n-gramů je enormní již pro malé hodnoty n i při malých hodnotách $|\mathcal{V}|$. Další faktor, který je nutné brát v úvahu, je existence *neviděných n-gramů*, tj. n-gramů, které se v trénovacích datech vůbec nevyskytly.

Těmto n-gramům totiž jazykový model trénovaný z těchto dat přiřadí nulovou pravděpodobnost výskytu (v případě, že $\mathrm{cnt}(\mathcal{T}, w_{k-n+1} w_{k-n+2} \ldots w_{k-1}) \neq 0$) nebo dokonce odhad pravděpodobnosti nemusí být definován, tj. $\mathrm{cnt}(\mathcal{T}, w_{k-n+1} w_{k-n+2} \ldots w_{k-1}) = 0$! Pro odhad pravděpodobnosti neviděných n-gramů se používají různé metody *vyhlazování* jazykových modelů. Tyto metody obecně odečítají část pravděpodobnostní masy pozorovaným n-gramům a tuto pravděpodobnostní masu přerozdělují mezi n-gramy v trénovacích datech nepozorované [3].

Pro vyhlazování jazykových modelů bylo použito výhradně *Witten-Bellova vyhlazování* [106]. Základní myšlenkou tohoto vyhlazování je učinit četnost nových jevů, které se vyskytly v trénovacích datech s danou historií h, proporcionální k odhadu nepozorovaných jevů $n(h)$. Jako odhad četnosti nepozorovaných jevů $n(h)$ je použit počet různých slov následujících v trénovacím korpusu historii h [30]. Witten-Bellovo vyhlazování má výhody v dobře definovaném odhadu pravděpodobností i pro fonémové jazykové modely. Při použití těch vyhlazovacích postupů (například Good-Turingovo vyhlazování [107]), které používají „počty počtů" (count-of-counts, tj. počet různých n-gramů s četností 1, 2, 3, atd.) může dojít k dělení nulou, neboť pro fonémové jazykové modely je například count-of-count pro unigramy a počet výskytů 1 s vysokou pravděpodobností nulový (tj. žádný z fonémů se neobjeví s četností 1).

5.6 Parser se skrytým vektorovým stavem

Parser se skrytým vektorovým stavem je rozšířením konečně stavového taggeru popsaného v kapitole 3.2 (str. 14). Uveďme zde detailněji popis struktury tohoto generativního modelu.

Mějme slovník $\mathcal{V}$ a posloupnost slov $W = (w_1, w_2, \ldots, w_T)$, kde $w_t \in \mathcal{V}$. Předpokládejme, že nad množinou možných posloupností slov je definována náhodná proměnná W s pozorovanou hodnotou W.

Parser se skrytým vektorovým stavem každému slovu w_t přiřadí právě jednu sémantickou značku c_t z konečné množiny sémantických značek $\mathcal{C}$. Značky c_t pak tvoří posloupnost $C = (c_1, c_2, \ldots, c_T)$ a nad množinou možných posloupností C definujme náhodnou proměnnou C. Nyní již lze formulovat pravděpodobnostní generativní model popisující vztah C a W (hodnoty náhodných proměnných jsou pro přehlednost vynechány):

$$\begin{aligned} \mathrm{P}(\mathsf{C}|\mathsf{W}) &= \frac{\mathrm{P}(\mathsf{W}|\mathsf{C})\mathrm{P}(\mathsf{C})}{\mathrm{P}(\mathsf{W})} \\ &\propto \mathrm{P}(\mathsf{W}|\mathsf{C})\mathrm{P}(\mathsf{C}) \end{aligned} \tag{5.95}$$

Pro určení nejpravděpodobnější posloupnosti sémantických značek $\hat{C}$ pak použijeme klasifikaci podle maximální aposteriorní pravděpodobnosti:

$$\hat{C} = \arg\max_{C} \mathrm{P}(\mathsf{W} = W|\mathsf{C} = C)\mathrm{P}(\mathsf{C} = C) \tag{5.96}$$

Takto definovaný generativní model se skládá ze dvou dílčích modelů:

- *Sémantického modelu* $\mathrm{P}(\mathsf{C} = C)$, který modeluje apriorní pravděpodobnost výskytu dané posloupnosti sémantických značek C.
- *Lexikálního modelu* $\mathrm{P}(\mathsf{W} = W|\mathsf{C} = C)$ vyčíslujícího pravděpodobnost pozorování posloupnosti slov W za podmínky, že posloupnost sémantických značek je právě C.

Popišme nyní vnitřní strukturu sémantických značek c_t. V případě, kdy tyto značky žádnou vnitřní strukturu nemají (jedná se o skalární hodnoty), je model definovaný rovnicí (5.96) ekvivalentní konečně stavovému taggeru.

Pokud však připustíme, že sémantické značky c_t jsou tvořeny vektory, můžeme o tomto modelu uvažovat jako o zásobníkovém automatu se stavem c_t. Posloupnost sémantických značek přiřazených vstupním slovům je pak tvořena vektory $\mathbf{c}_t$, tj. $C = (\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_T)$ a tyto vektory jsou tvořeny prvky $\mathbf{c}_t = (c_t[1], c_t[2], \ldots, c_t[D_t])$. Celé číslo D_t určuje hloubku zásobníku odpovídajícího slovu w_t. O prvcích zásobníku pak předpokládejme, že náleží konečné množině sémantických konceptů $\mathcal{C}'$. Uvažujme nyní promluvu z obrázku 2.2 (strana 11) a odpovídající zarovnaný sémantický strom. Tento strom můžeme zapsat pomocí linearizovaného zápisu jako:

$$\text{DEPARTURE}(v\ kolik\ to\ jede\ \text{TO}(na\ \text{STATION}(cheb))\ \text{TRAIN\_TYPE}(rychlíky))$$

Této promluvě pak lze přiřadit posloupnost sémantických vektorů $\mathbf{c}_t$:

$$C = \left(\begin{bmatrix} \text{DEPARTURE} \end{bmatrix}, \cdots, \begin{bmatrix} \text{DEPARTURE} \\ \text{TO} \end{bmatrix}, \begin{bmatrix} \text{DEPARTURE} \\ \text{TO} \\ \text{STATION} \end{bmatrix}, \begin{bmatrix} \text{DEPARTURE} \\ \text{TRAIN\_TYPE} \end{bmatrix} \right)$$

Parser se skrytým vektorovým stavem (Hidden vector state parser) [47, 55] modeluje přechody mezi jednotlivými stavy zásobníkového automatu stochasticky. Přechod mezi dvěma stavy zásobníkového automatu lze obecně rozložit na operace „*odstranění n konceptů ze zásobníku*" (pop) a „*uložení m nových konceptů na zásobník*" (push). HVS parser pak předpokládá, že nově vložené koncepty jsou podmíněny koncepty, které se nacházely na zásobníku před vložením nových konceptů. Sémantický model tedy není lexikalizovaný, neboť přechod mezi dvěma různými stavy HVS parseru nezávisí na vstupních slovech.

Takto konstruovaný model má nekonečné množství stavů. Je vhodné proto uvažovat několik omezení na přechody mezi jednotlivými stavy. Prvním uvažovaným omezením zajišťujícím konečnost množiny $\mathcal{C}$ je omezení maximální hloubky zásobníku na $D_{\max}$ sémantických konceptů. Dalším omezením, které He a Young uvažují, je položení $m = 1$ znamenající vložení právě jednoho konceptu na zásobník při každém přechodu mezi stavy. Tato omezení zajišťují lineární závislost počtu parametrů modelu na hloubce zásobníku, počtu různých sémantických konceptů $|\mathcal{C}'|$ a velikosti slovníku $|\mathcal{V}|$.

HVS parser pak aproximuje apriorní pravděpodobnost $\mathrm{P}(\mathsf{C} = C)$ pomocí sémantického modelu, kde je zavedena náhodná proměnná pop_t reprezentující počet odstraněných sémantických konceptů ze zásobníku $\mathbf{c}_t$ při přechodu mezi stavy $\mathbf{c}_{t-1}$ a $\mathbf{c}_t$. Popis tohoto modelu pak vypadá následovně:

$$\mathrm{P}(\mathsf{C} = C) = \prod_{t=1}^{T} \mathrm{P}(\mathsf{pop}_t | \mathbf{c}_{t-1}) \mathrm{P}(c_t[1] | \mathbf{c}_t[2, \ldots, D_t]) \tag{5.97}$$

Lexikální model HVS parseru je pak definován jako:

$$\mathrm{P}(\mathsf{W} = W | \mathsf{C} = C) = \prod_{t=1}^{T} \mathrm{P}(w_t | \mathbf{c}_t) \tag{5.98}$$

Poznamejme, že díky předpokladu $m = 1$, tj. počet uložených konceptů na zásobník je vždy roven jedné, není nutné v původním HVS parseru explicitně modelovat operaci uložení nového konceptu na zásobník. Pokud se stav zásobníku mezi dvěma slovy nezměnil, pak je použito $\mathsf{pop}_t = 1$ a $c_t[1] = c_{t-1}[1]$. Takto definovaný model generuje sémantické stromy s pravým větvením.

F. Jurčíček v pracích [46, 108] provedl modifikaci sémantického modelu HVS parseru z rovnice (5.97) tak, aby tento model umožňoval generování stromů s levo-pravým větvením (HVS parser s levo-pravým větvením, Left-Rigth Branching HVS parser, LRB-HVS parser)[4]. LRB-HVS parser umožňuje pravděpodobnostní modelování počtu sémantických konceptů vložených v daném časovém okamžiku na zásobník. Toho dosáhl zavedením nové skryté proměnné push_t, která určuje počet konceptů vložených na zásobník při přechodu ze stavu $\mathbf{c}_{t-1}$ do stavu $\mathbf{c}_t$. Sémantický model LRB-HVS modelu je pak dán rovnicí:

$$\begin{aligned} \mathrm{P}(\mathsf{C}) = &\prod_{1}^{T} \mathrm{P}(\mathsf{pop}_t|\mathbf{c}_{t-1})\mathrm{P}(\mathsf{push}_t|\mathbf{c}_{t-1})\cdot \\ &\cdot \begin{cases} 1 & \text{pokud } \mathsf{push}_t = 0 \\ \mathrm{P}(c_t[1]|\mathbf{c}_t[2,\dots D_t]) & \text{pokud } \mathsf{push}_t = 1 \\ \mathrm{P}(c_t[1]|\mathbf{c}_t[2,\dots D_t])\mathrm{P}(c_t[2]|\mathbf{c}_t[3,\dots D_t]) & \text{pokud } \mathsf{push}_t = 2 \end{cases} \end{aligned} \tag{5.99}$$

V práci [46] bylo ukázáno, že hodnotu proměnné push_t lze omezit na nejvýše dva sémantické koncepty uložené na zásobník při přechodu mezi dvěma stavy. Poznamenejme, že HVS parser jako generativní model používá celou řadu předpokladů o podmíněné nezávislosti jednotlivých náhodných proměnných. Tyto předpoklady zde nejsou uvedeny, pro jejich přehled odkažme do publikací [46, 47, 55].

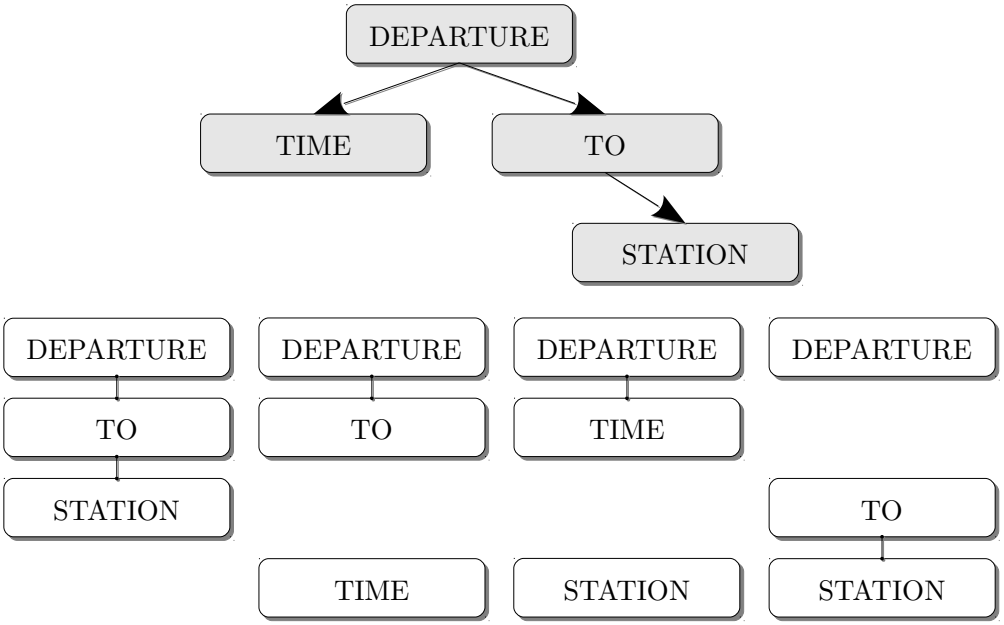
5.7 Klasifikátory sémantických n-tic

Klasifikátory sémantických n-tic (Semantic tuple classifiers, STC) je přístup k porozumění mluvené řeči vyvinutý na Univerzitě v Cambridge v roce 2009 [59]. Obecný popis tohoto modelu již byl zmíněn v kapitole 3.3 (str. 15). Autoři prezentují jednoduchou techniku, která využívá množiny natrénovaných klasifikátorů, které diskriminují jednotlivé sémantické koncepty. Struktura cílových tříd a předzpracování vstupní promluvy umožňuje rekonstrukci sémantického stromu, při trénování není nutná informace o zarovnání konceptů sémantického stromu se vstupní promluvou, STC model se trénuje z abstraktních sémantických anotací.

Vstupem trénovacího algoritmu STC je množina promluv a odpovídajících abstraktních sémantických stromů. Každý sémantický strom je rozdělen na tzv. *sémantické n-tice*, které mohou být chápány jako podposloupnosti o maximální délce k konceptů. Podposloupnosti jsou tvořeny z posloupností konceptů na cestě z kořene sémantického stromu do libovolného uzlu. Příkladem budiž obrázek 5.16 zobrazující dekompozici abstraktního sémantického stromu na sémantické n-tice.

Předpokládejme, že trénovací množina se skládá z dvojic vstupní promluva u_i a odpovídající abstraktní sémantický strom s_i, tj. $\mathcal{T} = \{(u_i, s_i)\}_{i=1}^{l}$. Říkejme, že sémantická n-tice $\mathbf{t}$ náleží do abstraktního sémantického stromu s (tj. $\mathbf{t} \in s$), je-li $\mathbf{t}$ podposloupnost libovolné cesty z kořenového uzlu stromu s do některého z uzlů stromu s. V opačném případě budeme psát $\mathbf{t} \notin s$. Množinu všech sémantických n-tic budeme označovat jako $\mathcal{S} = \{\mathbf{t} \in s_i; s_i \in \mathcal{T}\}$. Obdobně pro sémantické n-tice délky k označme $\mathcal{S}_k = \{\mathbf{t} \in \mathcal{S}; |\mathbf{t}| = k\}$.

[4] Rozdíly mezi stromy s pravým a levo-pravým větvením, stejně jako důsledky na přesnost porozumění jsou diskutovány v práci [46].



Obrázek 5.16: Sémantický strom (šedá barva) s abstraktní sémantickou anotací DEPARTURE(TIME, TO(STATION)) a jeho dekompozice na sémantické n-tice délky 1 až 3 (bílá barva).

5.7.1 Trénovací algoritmus

Algoritmus pro trénování modelu založeného na klasifikátorech sémantických n-tic pak sestává z následujících kroků [59]:

Vstup:

- Trénovací množina $\mathcal{T} = \{(u_i, s_i)\}_{i=1}^{l}$
- Maximální délka sémantických n-tic k
- Databáze doménově závislých sémantických entit

Výstup:

- Množina klasifikátorů sémantických n-tic $\mathcal{C} = \{C_j\}$
- Doménová gramatika G

Algoritmus trénování klasifikátorů sémantických n-tic:

1. Náhrada všech výskytů sémantických entit odpovídajícími identifikátory.
2. Výpočet lexiko-syntaktických příznaků $\mathbf{x}_i$ pro každou promluvu z množiny $\mathcal{T}$.
3. Pro všechny sémantické n-tice $\mathbf{t}_j \in \mathcal{S}_k$:
   (a) Vytvořit trénovací množinu pro trénování klasifikátoru $\mathcal{T}_j = \{(\mathbf{x}_i, y_i^j\}$, kde $y_i^j = 1$ pokud $\mathbf{t}_j \in s_i$, jinak $y_i^j = -1$.
   (b) Využití trénovací množiny $\mathcal{T}_j$ pro trénování binárního klasifikátoru realizujícího funkci $\hat{y}^j = C_j(\mathbf{x})$. V původní práci [59] byly použity binární klasifikátory založené na SVM.
4. Sestavení doménové gramatiky, která generuje všechny sémantické stromy z trénovací množiny $\mathcal{T}$.

5.7.2 Dekódovací algoritmus

Vstup:

- Neznámá promluva u
- Množina sémantických klasifikátorů $\mathcal{C}$
- Doménová gramatika G
- Databáze doménově závislých sémantických entit

Výstup:

- Sémantický strom $\hat{s}$ odpovídající u

Algoritmus dekódování pomocí klasifikátorů sémantických n-tic:

1. Náhrada všech výskytů sémantických entit v promluvě u odpovídajícími identifikátory tříd.

2. Výpočet lexiko-syntaktických příznaků $\mathbf{x}$ z promluvy u, přičemž se odstraní ty příznaky, které nebyly pozorovány ve fázi trénování.

3. Pro každé $C_j \in \mathcal{C}$ predikce $\hat{y}^j = C_j(\mathbf{x})$. Sestavení množiny sémantických n-tic odpovídajících promluvě u:

$$\hat{S} = \{\mathbf{t}_j \in S_k : \hat{y}^j = 1\} \tag{5.100}$$

4. Nalezení odpovídajícího abstraktního sémantického stromu $\hat{s}$ k množině predikovaných sémantických n-tic $\hat{S}$. Algoritmus nejprve vytvoří výstupní strom $\hat{s}$ obsahující pouze kořenový uzel odpovídající startovacímu symbolu doménové gramatiky G. Dále nastaví proměnnou r ukazující na tento uzel. Autoři v [59] popisují dva možné módy algoritmu:

   (a) *Mód s vysokou přesností* – pro každou n-tici $\mathbf{t} = (t_1, t_2, \ldots t_n) \in \hat{S}$, pro kterou $t_1 = r$, přidej do stromu s uzly $t_2, \ldots t_n$ uspořádané tak, že $r = t_1$ a t_{i-1} je předchůdce t_i. Odstraň $\mathbf{t}$ z $\hat{S}$. Rekurzivně opakuj nastavováním r na všechny nezpracované uzly z $\hat{s}$.

   (b) *Mód s vysokou úplností* – vytvoř strom $\hat{s}$ podle (a). Zbývající n-tice z $\hat{S}$, které již není možné do $\hat{s}$ přidat, zpracuj následujícím způsobem:

   Rekurzivně pro všechny uzly r ze stromu $\hat{s}$ a pro všechny zbývající n-tice $\mathbf{t} = (t_1, t_2, \ldots t_n) \in \hat{S}$ hledej v doménové gramatice n-tici $(r, t_1, \ldots t_n)$. Pokud taková n-tice existuje, přidej do stromu s uzly $t_1, \ldots t_n$ uspořádané tak, že r je předchůdce t_1 a t_{i-1} je předchůdce t_i.

5. Zarovnání abstraktního sémantického stromu s odpovídajícími identifikátory lexikálních tříd.

6. Zpětné nahrazení identifikátorů tříd za odpovídající slova vstupní promluvy.

Doménově závislá databáze sémantických entit přináší do úlohy expertní znalost. Je prezentována jako seznam dvojic *sémantický identifikátor/posloupnost slov*, např. STATION = *Plzeň* nebo STATION = *Ústí nad Labem*. Využití expertní znalosti umožňuje rozšíření modelu na úlohy, kde se vyskytuje velký počet různých hodnot jednotlivých konceptů. Před vyčíslením příznakového vektoru pro libovolnou vstupní promluvu,

jak ve fázi trénování, tak ve fázi dekódování, je provedena náhrada podposloupností vstupní promluvy za odpovídající sémantické identifikátory. V případě víceznačného mapování nebo překryvů je vybrán ten sémantický identifikátor, který se shoduje v největším počtu slov.

Parametr l určující maximální délku sémantické n-tice řídí zároveň i vyvážení mezi přesností jednotlivých sémantických klasifikátorů a nejednoznačností generovaného sémantického stromu. Příliš dlouhé sémantické n-tice vedou na triviální rekonstrukci sémantického stromu, ale za cenu nízké přesnosti predikce výskytu dané sémantické n-tice. Autoři původního STC modelu používali fixní délku l sémantických n-tic. Na referenční databázi ATIS [109] bylo nastaveno $l = 3$. Příznakový vektor $\mathbf{x}(u)$ byl získán jako četnost různých n-gramů ($1 \leq n \leq 3$) v promluvě u, což vede na příznakový vektor obsahující desítky tisíc položek. Autoři proto použili SVM klasifikátory s lineární jádrovou funkcí. Celkový počet klasifikátorů byl přibližně 250 a doba zpracování neznámé promluvy byla průměrně méně než 200 ms.

Zkušenosti získané při reimplementaci STC modelu byly hlavní motivací pro výzkum popsaný v této disertační práci. Výsledkem je pak diskriminativní model porozumění řeči, který STC model rozšiřuje takřka ve všech směrech:

- Náhrada posloupností slov identifikátory lexikálních tříd v rozpoznané promluvě. V rámci této práce došlo k úpravě tohoto nahrazování pomocí detekce sémantických entit, která může být navíc realizována nad neurčitým výstupem z rozpoznávání řeči ve tvaru mřížky.
- Možnost využít celé slovní (a dokonce fonémové) mřížky pro trénování a přiřazování sémantických stromů. Původní STC model oproti tomu umožňoval zpracování pouze první nejlepší slovní hypotézy.
- Modelování vzájemně korelovaných výstupů jednotlivých klasifikátorů sémantických entit. V této práci je navrženo rozšíření, které přidává další vrstvu diskriminativních klasifikátorů, která tuto korelaci bere v úvahu.
- Schopnost generovat více výstupních sémantických stromů (významových hypotéz) s přiřazenými aposteriorními pravděpodobnostmi. Navíc takto generované sémantické stromy jsou v rámci dále popsané struktury diskriminativního modelu kombinovány s expertní znalostí zahrnutou v detekci sémantických entit.

Kapitola 6

Diskriminativní model pro porozumění mluvené řeči

Tato práce vychází z předchozích výsledků výzkumu na pracovišti autora, především na závěrech prací [46, 58], které byly věnovány generativním modelům porozumění řeči. Jednalo se o různé modifikace parseru se skrytým vektorovým stavem (viz kapitoly 3.2 a 5.6). Výzkum v oblasti hlasových dialogových systémů však ukázal, že i přes dobré výsledky tohoto modelu nad referenčním přepisem (tj. přepisem dané promluvy) je přesnost porozumění z rozpoznaných promluv nedostatečná pro robustní řízení dialogu.

Současný stav ukazuje, že přesnost generativních modelů může být snadno překonána modely diskriminativními [2, 59, 65]. Nižší přesnost generativních modelů je způsobena především příliš silnými předpoklady nezávislosti jednotlivých náhodných proměnných, které vyplývají z toho, že sémantické modely generativních parserů jsou jak pro HVS parser, tak pro konečně stavové modely nelexikalizované, tj. modelují pouze apriorní pravděpodobnost dané posloupnosti sémantických značek.

Dalším důvodem pro nižší přesnost generativních modelů jsou především řídká trénovací data. Pro ilustraci použijme HVS parser jako zástupce třídy generativních modelů. Jeho lexikální model je dán jako:

$$\mathrm{P}(\mathsf{U}|\mathsf{C}) \approx \prod_{t=1}^{T} \mathrm{P}(w_t|\mathbf{c}_t) \tag{6.1}$$

kde $\mathrm{P}(w_t|\mathbf{c}_t)$ je pravděpodobnost pozorování slova w_t za podmínky aktuálního vektorového stavu $\mathbf{c}_t$. Uvažujme příklad, kdy slova w_t náleží slovníku automatického rozpoznávače řeči o velikosti $|\mathcal{V}| = 20\,000$ slov a počet různých vektorových stavů $\mathbf{c}_t$ položme roven 100. Pak pro odhad parametrů lexikálního modelu je nutné určit 2 000 000 parametrů. Porovnáme-li však toto číslo s počtem přibližně 22 000 tokenů (slovních jednotek) v trénovací sadě korpusu HHTT (viz kapitola 9.1), zjistíme, že na jeden token trénovacích dat připadá řádově 90 parametrů lexikálního modelu. Přestože lze použít různé způsoby vyhlazování, nebude tento model příliš robustní a to zejména pro málo četné vektory sémantických konceptů $\mathbf{c}_t$.

Rozpracujme nyní ideu diskriminativních modelů, které modelují přímo aposteriorní pravděpodobnost daného významu při známém vstupu. Přestože diskriminativních modelů pro porozumění řeči existuje celá řada, pokusíme se popsat jeden z možných nových pohledů na tuto problematiku. Jádrem tohoto pohledu je konceptový model, který ke vstupní

promluvě přiřazuje abstraktní sémantický strom. Následně je konceptový model doplněn o detekci sémantických entit a o model zarovnání, které provádějí částečné zarovnání lexikální realizace vstupní promluvy s abstraktním sémantickým stromem. Tuto myšlenku v nejjednodušším provedení lze nalézt již u modelu popsaného dříve – u klasifikátorů sémantických entit. Zde samotné klasifikátory a výstupní heuristika přiřazují výstupní sémantický strom, který je následně s pomocí lexikálních tříd a jednoduchých pravidel zarovnán se vstupní promluvou. Tuto myšlenku dále rozvineme do podoby plně pravděpodobnostního diskriminativního modelu, porozumění řeči je pak prováděno na dvou úrovních:

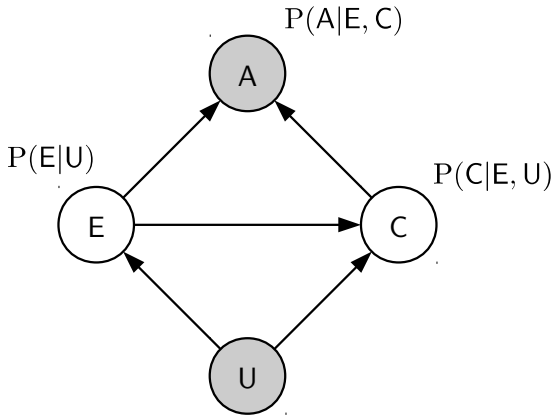
- *Porozumění nižší úrovně* (lokální) – hledá v promluvě významové entity, které reprezentují elementární prvky významu. Tyto entity jsou hledány striktně lokálně, bez ohledu na výskyt dalších entit nebo konceptů.
- *Porozumění vyšší úrovně* (globální) – přiřazuje promluvě význam jako celku. Tento význam je reprezentován abstraktním sémantickým stromem. Při globálním pohledu jsou již známy významové (sémantické) entity nalezené v promluvě a lze je tudíž použít při predikci sémantického stromu.

V této kapitole bude ustanoven pravděpodobnostní rámec kombinující porozumění nižší a vyšší úrovně do jediného diskriminativního modelu. V rámci tohoto modelu označme jako T náhodnou proměnnou nad množinou (částečně) zarovnaných sémantických stromů. Uvažujme, že proměnná T se skládá z dílčích náhodných proměnných E a C. Náhodná proměnná E je definována nad všemi různými posloupnostmi sémantických entit. Přesněji E je definováno nad množinou posloupností $e = \{e_1, e_2, \ldots\}$, přičemž každá posloupnost e se skládá z dílčích sémantických entit e_i. Každá sémantická entita reprezentuje nějaký konkrétní objekt zmíněný v dané promluvě a významný z pohledu sémantické analýzy. Pro příklad jmenujme sémantické entity typu čas, které se mohou skládat z údaje o hodinách a minutách (více v kapitole 8).

Náhodná proměnná C je pak definována nad množinou různých významů, například nad množinou všech možných sémantických stromů nebo nad množinou možných sémantických konceptů přiřazených dané promluvě. Náhodnou proměnnou C budeme uvažovat jako náhodnou proměnnou definovanou nad všemi možnými abstraktními sémantickými stromy.

Aposteriorní pravděpodobnost $\mathrm{P}(\mathsf{T} = t|\mathsf{U})$ pak odpovídá sdružené pravděpodobnosti $\mathrm{P}(\mathsf{E} = e, \mathsf{C} = c|\mathsf{U})$. Dále zaveďme binární náhodnou proměnnou A, která nabývá hodnoty 1 pokud je možné posloupnost sémantických entit e a abstraktní sémantický strom c zarovnat tak, aby sémantické entity odpovídaly sémantickým konceptům. Je-li $\mathsf{A} = 1$, pak zarovnaný sémantický strom t skládající se ze sémantických entit e a abstraktního sémantického stromu c je *validní*. V úloze porozumění mluvené řeči nás budou zajímat právě validní zarovnané sémantické stromy, tj. případy, kdy $\mathsf{A} = 1$. Potom:

$$\mathrm{P}(\mathsf{T} = t|\mathsf{U} = u) = \mathrm{P}(\mathsf{E} = e, \mathsf{C} = c|\mathsf{U} = u, \mathsf{A} = 1) \tag{6.2}$$



Obrázek 6.1: Bayesovská síť vyjadřující vzájemné vztahy náhodných proměnných U (promluva, pozorovaná proměnná), E (posloupnost sémantických entit), C (abstraktní sémantický strom) a A (proměnná určující validitu zarovnaného stromu T = (E, C), pozorovaná proměnná).

Pro přehlednost v dalším odvození vynechejme konkrétní hodnoty náhodných proměnných:

$$\begin{aligned}
\mathrm{P}(\mathsf{E},\mathsf{C}|\mathsf{U},\mathsf{A}) &= \mathrm{P}(\mathsf{C}|\mathsf{E},\mathsf{U},\mathsf{A}) \cdot \mathrm{P}(\mathsf{E}|\mathsf{U},\mathsf{A}) \\
&= \frac{\mathrm{P}(\mathsf{C}|\mathsf{E},\mathsf{U},\mathsf{A}) \cdot \mathrm{P}(\mathsf{A}|\mathsf{E},\mathsf{U})}{\mathrm{P}(\mathsf{A}|\mathsf{E},\mathsf{U})} \cdot \mathrm{P}(\mathsf{E}|\mathsf{U},\mathsf{A}) \\
&= \frac{\mathrm{P}(\mathsf{A},\mathsf{C}|\mathsf{E},\mathsf{U})}{\sum_{\mathsf{C}} \mathrm{P}(\mathsf{A},\mathsf{C}|\mathsf{E},\mathsf{U})} \cdot \mathrm{P}(\mathsf{E}|\mathsf{U},\mathsf{A}) \\
&= \frac{\mathrm{P}(\mathsf{A}|\mathsf{C},\mathsf{E},\mathsf{U}) \cdot \mathrm{P}(\mathsf{C}|\mathsf{E},\mathsf{U})}{\sum_{\mathsf{C}} \mathrm{P}(\mathsf{A}|\mathsf{C},\mathsf{E},\mathsf{U}) \cdot \mathrm{P}(\mathsf{C}|\mathsf{E},\mathsf{U})} \cdot \mathrm{P}(\mathsf{E}|\mathsf{U},\mathsf{A}) \\
&\approx \frac{\mathrm{P}(\mathsf{A}|\mathsf{C},\mathsf{E}) \cdot \mathrm{P}(\mathsf{C}|\mathsf{E},\mathsf{U})}{\sum_{\mathsf{C}} \mathrm{P}(\mathsf{A}|\mathsf{C},\mathsf{E}) \cdot \mathrm{P}(\mathsf{C}|\mathsf{E},\mathsf{U})} \cdot \mathrm{P}(\mathsf{E}|\mathsf{U})
\end{aligned} \tag{6.3}$$

Tento vztah náhodných proměnných může být ekvivalentně zachycen pomocí Bayesovské sítě na obrázku 6.1. Tato Bayesovská síť vyjadřuje množinu výroků o podmíněné nezávislosti jednotlivých náhodných proměnných, přičemž šipka od proměnné A k proměnné B vyjadřuje, že B podmiňuje pravděpodobnostní distribuci A. Pozorované náhodné proměnné jsou pak označeny šedým podbarvením. Více o Bayesovských sítích pak v [29, 31, 110].

Odtud je vidět, že úlohu nalezení modelu modelujícího pravděpodobnostní rozdělení P(T|U) lze dekomponovat na úlohu nalezení následujících dílčích modelů:

- P(C|E, U) = P(C = c|E = e, U = u) – *konceptový model* predikující pravděpodobnost abstraktního sémantického stromu c při pozorování vstupní promluvy u a odpovídající posloupnosti sémantických entit e. Konceptový model je tvořen *hierarchickým diskriminativním modelem* popsaným v kapitole 7. Tento model pro vstupní promluvu u reprezentovanou pomocí váženého konečného akceptoru (mřížky) predikuje pravděpodobnostní distribuci nad různými abstraktními sémantickými stromy c, tj. rozdělení $P(\mathsf{C} = c|\mathsf{U} = u)$. Struktura tohoto modelu umožňuje přidání dalších příznaků, čehož je s výhodou použito pro podmínění pravděpodobnostního rozdělení $P(\mathsf{C} = c|\mathsf{U} = u)$ posloupností sémantických entit e a tím modelování P(C|E, U) (kapitola 7.3.3).

- $\mathrm{P}(\mathsf{E}|\mathsf{U}) = \mathrm{P}(\mathsf{E} = e|\mathsf{U} = u)$ – *model detekce sémantických entit* přiřazující posloupnost sémantických entit e vstupní promluvě u. Je použit přístup, kdy sémantické entity jsou popsány pomocí expertně navržených bezkontextových gramatik. Algoritmus detekce sémantických entit (kapitola 8) pak na jejich základě nalezne v mřížce u všechny podřetězce, které patří do jazyka generovaného těmito gramatikami. Jelikož se tyto generované podřetězce mohou překrývat, je nutné nejprve nalézt množinu jednoznačně přiřazených sémantických entit (kapitola 8.1). Pro úlohu porozumění řeči je pak vhodné z této množiny opět sestavit mřížku sémantických entit E, která modeluje možné posloupnosti sémantických entit e a odpovídající pravděpodobnosti $\mathrm{P}(\mathsf{E} = e|\mathsf{U} = u)$ (kapitola 8.2).

- $\mathrm{P}(\mathsf{A}|\mathsf{E},\mathsf{C}) = \mathrm{P}(\mathsf{A} = 1|\mathsf{E} = e, \mathsf{C} = c)$ – *model zarovnání* určující, zda posloupnost sémantických entit e lze zarovnat s abstraktním sémantickým stromem c. Model zarovnání byl založený na jednoduchém pravidle: sémantický strom definovaný pomocí e a c je validní ($\mathsf{A} = 1$) právě tehdy, když všechny sémantické entity z e lze přiřadit odpovídajícím sémantickým konceptům v abstraktním sémantickém stromu c. Tato definice modelu zarovnání *nevyžaduje* zarovnání všech konceptů z c na sémantické entity e.

  Podle tohoto pravidla se například sémantická entita typu *time* (časový údaj) se zarovnává se sémantickým konceptem TIME. Obdobně entity typu *station* (stanice) se zarovnávají s konceptem STATION. Požadujeme-li pouze validní sémantické stromy a objeví-li se v posloupnosti e sémantická entita *station*, pak se nutně v odpovídajícím abstraktním sémantickém stromu c musí objevit koncept STATION. Opačnou relaci nepožadujeme, pokud se v sémantickém stromu objeví koncept TIME, není nutné aby se vyskytl v posloupnosti sémantických entit e. To umožní detekovat případy, kdy se ve významu promluvy vyskytl časový údaj (byl predikován konceptovým modelem), který ale neodpovídá bezkontextové gramatice pro časy (nebyl predikován detekcí sémantických entit).

Výše zmíněná struktura diskriminativního modelu efektivně kombinuje statistický a znalostní (expertní) přístup k návrhu porozumění řeči. Statistický přístup je zastoupen hierarchickým diskriminativním modelem (kapitola 7, str. 61). Znalostní přístup pak algoritmem pro detekci sémantických entit (kapitola 8, str. 85)

Poznamenejme, že v následujícím textu budeme uvažovat, že pravděpodobnostní distribuce nad rozpoznanými slovy $\mathrm{P}(\mathsf{U} = u)$ bude reprezentována mřížkou (váženým konečným akceptorem) U a obdobně pravděpodobnostní distribuce $\mathrm{P}(\mathsf{E} = e|\mathsf{U} = u)$ bude reprezentována mřížkou (váženým konečným akceptorem) E.

Kapitola 7

Hierarchický diskriminativní model

Hierarchický diskriminativní model vznikl při experimentech s modelem STC. Již prvotní implementace modelu STC předčila referenční HVS parser v přesnosti predikce sémantických stromů. I přes velice slibné výsledky měl původní STC model některé nevýhody – především nemožnost získání více výstupních hypotéz s přiřazenými aposteriorními pravděpodobnostmi, dále pak generování výstupního sémantického stromu založené na heuristice a zarovnání vygenerovaného sémantického stromu s lexikální realizací promluvy založené na lexikálních třídách. Mezi další nevýhody STC modelu patří nemožnost trénování a predikce z neurčitého vstupu reprezentovaného mřížkou. Navíc výstupy jednotlivých dílčích klasifikátorů v modelu STC jsou korelovány a tato korelace není žádným způsobem uvažována při rekonstrukci sémantického stromu.

Nově vyvinutý hierarchický diskriminativní model (HDM, Hierarchical Discriminative Model) rozšiřuje a doplňuje původní STC model takovým způsobem, že eliminuje výše zmíněné nevýhody tohoto modelu. Uveďme výčet vlastností HDM v porovnání s modelem STC:

- Model HDM využívá racionální jádrové funkce pro vyčíslení podobnosti mezi dvěma vstupními promluvami. Jak bylo řečeno v kapitole 2.2, možným výstupem systému automatického rozpoznávání řeči je kromě nejlepší hypotézy (1-best) i mřížka hypotéz v podobě acyklického váženého konečného automatu. Model STC (kapitola 5.7) využívá lexiko-sémantické příznaky (např. četnost n-gramů), což vede na dimenze příznakového vektoru v řádech desítek tisíc prvků. Je však třeba zdůraznit, že v klasifikátorech SVM se nakonec použije pouze skalární součin těchto rozměrných vektorů.

  S využitím teorie popsané v sekci 5.3 je možné použít mřížky ze systému rozpoznávání řeči přímo jako prvky trénovací množiny bez nutnosti explicitně vyčíslovat příznakový vektor $\mathbf{x}(u)$ pro každou promluvu u. Navíc díky operacím definovanými nad váženými konečnými transducery lze získat efektivní implementaci algoritmu výpočtu hodnot racionální jádrové funkce, což je vzhledem k předpokládanému nasazení v hlasových dialogových systémech, kde rychlost odezvy je tou nejvyšší prioritou, vlastnost velice vítaná.

- Využití SVM jako klasifikátorů v modelu STC vede na nutnost uchovávat podpůrné vektory pro jednotlivé klasifikátory. Obecně existuje možnost, že každý z prvků trénovací množiny $\mathcal{T}$ je podpůrným vektorem v některém z trénovaných klasifikátorů. Použití racionálních jádrových funkcí umožňuje efektivně reprezentovat trénovací množinu ve formě minimálního deterministického váženého konečného transduceru a výpočet jádrové funkce pak lze realizovat prostou kompozicí vstupní mřížky a transduceru pro výpočet jádrové funkce.

- Racionální jádrové funkce umožňují jednotné zpracování různých typů vstupních dat. Jedná se především o slovní přepis a první nejlepší slovní hypotézu ze systému rozpoznávání řeči, nicméně tato disertační práce se zabývá i využitím slovních mřížek již ve fázi trénování sémantického modelu nikoli až ve fázi dekódování významu. Navíc je zde studována i možnost využití fonémového rozpoznávače pro porozumění z nejlepší fonémové hypotézy a fonémové mřížky.

- Rekonstrukce sémantického stromu na výstupu STC modelu je heuristika, která pracuje nad binárními výstupy klasifikátorů, tj. pracuje pouze s informací, že daná sémantická n-tice ve výstupním stromu má/nemá být přítomna. HDM oproti tomu používá druhou vrstvu diskriminativních klasifikátorů, které jsou schopny kombinovat výstupy jednotlivých klasifikátorů sémantických n-tic. Dekódovací algoritmus použitý v HDM pak umožňuje získání pravděpodobnostní distribuce nad možnými abstraktními sémantickými stromy. Korelace výstupů jednotlivých klasifikátorů sémantických n-tic je tak uvažovaná již při trénování modelu.

- Původní implementace STC používá sémantické n-tice o dané délce l, přičemž tato délka je volena tak, aby bylo dosaženo optimálního vyvážení mezi přesností jednotlivých klasifikátorů a nejednoznačností rekonstrukce sémantického stromu (kapitola 5.7.1). HDM namísto toho používá všechny sémantické n-tice délky nejvýše l, tj. $1 \leq n \leq l$. Je tudíž možné kombinovat výstupy různých klasifikátorů sémantických entit, které jsou trénovány z různých podmnožin trénovací množiny. Navíc lze použít vektor doplňkových příznaků, který může popisovat další vlastnosti vstupní promluvy, například výskyt různých sémantických entit, případně kontextovou nebo historickou informaci o probíhajícím dialogu.

  Příklad: klasifikátor predikující výskyt sémantické n-tice (STATION) sdílí část trénovacích dat s klasifikátory predikujícími (TO, STATION) a (FROM, STATION), poskytuje tak obecně robustnější predikci než „specializované" klasifikátory. Výstupní vrstva je pak schopna na základě informace, že výstupní strom by měl obsahovat n-tici (STATION) a na základě vzdálenosti vstupní promluvy k rozhodovacím nadrovinám klasifikátorů predikujících (TO, STATION) a (FROM, STATION) rozhodnout, že výstupní strom obsahuje například n-tici (TO, STATION), přestože první vrstva tuto n-tici vůbec nepredikovala.

- Výstupní vrstva HDM predikuje pravděpodobnosti expanze některého uzlu sémantického stromu na danou množinu následnických uzlů. Je proto možné vyčíslit pravděpodobnost nejlepšího sémantického stromu. Je možné získat i dalších n-nejpravděpodobnějších sémantických stromů. Jak odhad pravděpodobnosti predikovaného stromu, tak n-nejpravděpodobnějších stromů lze s výhodou použít v subsystému řízení dialogu založeného na statistických metodách a efektivně tak reprezentovat neurčitost na úrovni významu vstupní promluvy.

- Zarovnání abstraktního sémantického stromu generovaného HDM s původní lexikální realizací je možné díky uvažování HDM jako konceptového modelu $\mathrm{P}(\mathsf{C}|\mathsf{E},\mathsf{U})$ v rámci struktury popsané v kapitole 6. Poznamenejme, že jak konceptový model reprezentovaný HDM, tak model detekce sémantický entit (kapitola 8) a model zarovnání umožňují reprezentaci pozorované promluvy u pomocí odpovídající slovní mřížky U a tudíž lze v rámci celé struktury modelu pracovat s neurčitostí způsobenou subsystémem rozpoznávání řeči.

Pro účely popisu HDM si vypůjčíme terminologii z teorie umělých neuronových sítí, konkrétně z popisu dopředných perceptronových sítí. HDM se skládá ze tří vrstev (obrázek 7.9):

1. *Vstupní vrstva*, která efektivně vyčísluje hodnoty racionální jádrové funkce vzájemně mezi promluvami z trénovací množiny, případně mezi novou dosud neviděnou promluvou a všemi promluvami z trénovací množiny. Výstupem vstupní vrstvy je vektor hodnot racionálních jádrových funkcí použitý pro trénování a predikci pomocí skryté vrstvy.

2. *Skrytá vrstva* odpovídá modelu STC. Ten na základě vektoru hodnot jádrových funkcí generuje výstup skryté vrstvy. Ten je tvořený vektorem vzdáleností vstupní promluvy k oddělovacím nadrovinám binárních SVM klasifikátorů klasifikujících přítomnost jednotlivých sémantických n-tic.

3. *Výstupní vrstva* predikující na základě příznakového vektoru získaného z výstupu skryté vrstvy pravděpodobnosti jednotlivých sémantických pravidel. Z těchto sémantických pravidel je pak sestaven výstupní abstraktní sémantický strom.

Přestože zkratka HDM označuje hierarchický diskriminativní model, budeme používat i slovní spojení *HDM model* nebo *model HDM*. Věřím, že laskavý čtenář toto „zdvojení" slova model – jednou jako součást zkratky a jednou jako samostatné slovo – promine v zájmu zachování čitelnosti textu.

7.1 Vstupní vrstva

Vstupní vrstva slouží k výpočtu hodnot jádrové funkce nad dvěma promluvami $u_j, u_k \in \mathcal{T}$ z trénovací množiny, případně k výpočtu hodnot jádrové funkce mezi neviděnou promluvou u a promluvami $u_k \in \mathcal{T}$. V případě modelu STC jsou vstupní promluvy reprezentovány pomocí příznakového vektoru obsahujícího lexikálně-syntaktické příznaky – nazvěme tento vektor $\mathbf{x}(u)$. Jádrová funkce v případě STC je lineární funkcí danou jako:

$$K(u_j, u_k) = \mathbf{x}(u_j) \cdot \mathbf{x}(u_k) \tag{7.1}$$

Autoři [59] popisují použití pouze lineární jádrové funkce, ale obecně může být použita libovolná jádrová funkce splňující Mercerovu podmínku (kapitola 5.1.4).

Jeden z cílů disertační práce je navrhnout model porozumění, který umožňuje využít nejednoznačný výstup systému automatického rozpoznávání řeči ve formě slovní, nebo fonémové mřížky. Vyčíslení příznakových vektorů pro obecnou slovní nebo fonémovou mřížku může

být výpočetně náročný proces. Lze však s výhodou použít teorii racionálních jádrových funkcí (kapitola 5.3).

Racionální jádrové funkce jsou definovány nad dvojicí vážených konečných automatů. V této kapitole budeme předpokládat, že tyto automaty jsou slovní resp. fonémové řetězce, popř. mřížky. Jinými slovy budeme předpokládat, že se jedná o acyklické vážené konečné akceptory.

V tomto případě lze racionální jádrovou funkci definovat pomocí transduceru $T \circ T^{-1}$ a funkce ψ, kde T je vážený konečný automat nad polookruhem $\mathbb{K}$ definující tuto jádrovou funkci a funkce ψ provádí zobrazení hodnot z polookruhu $\mathbb{K}$ do prostoru reálných čísel. Kapitola 5.3.4 popisuje efektivní algoritmus pro výpočet jádrové funkce mezi dvěma váženými acyklickými akceptory. Poznamenejme, že při použití SVM jako klasifikátoru je nutné vyčíslit jádrovou funkci mezi klasifikovaným vektorem a všemi podpůrnými vektory. Počet podpůrných vektorů závisí na velikosti trénovací množiny a z definice SVM mohou podpůrné vektory pokrývat celou trénovací množinu.

V této prvotní implementaci se omezíme pouze na n-gramové racionální jádrové funkce definované transducerem $T_{n,m}$ z rovnice (5.81). Použití tohoto transduceru je ekvivalentní použití příznakového vektoru, který obsahuje střední četnosti n-gramů o délce n až m ve mřížce U, spolu s lineární jádrovou funkcí. I přesto však tyto závěry nejsou platné pouze pro třídu n-gramových racionálních jádrových funkcí, ale lze je aplikovat i na další racionální jádrové funkce popsané v kapitole 5.3.3.

Pokud jsou vstupní promluvy reprezentované pomocí řetězců nebo mřížek, je postupné vyčíslování jádrové funkce mezi neznámým vstupem a každým prvkem trénovací množiny neefektivní, neboť mnoho podřetězců vstupních mřížek se vyskytuje velmi často v různých mřížkách (častá slova, výplňová slova apod.). Výpočet racionální jádrové funkce lze často dekomponovat tak, že výpočet nad těmito shodnými částmi mřížek lze provádět pouze jednou a výsledek přiřadit všem mřížkám, které tuto část dat sdílejí. S výhodou lze použít optimalizační algoritmy definované nad váženými konečnými automaty, především determinizaci a minimalizaci. Postup prezentovaný v tomto odstavci je inspirován postupem při optimalizaci indexu v úloze spoken term detection (STD) [69] – optimální index je v tomto případě reprezentován pomocí minimálního deterministického váženého konečného automatu (více lze nalézt v kapitole 3.6). Obdobně je postupováno při výpočtu racionální jádrové funkce, kdy celá trénovací množina $\mathcal{T}$ je reprezentována jako sjednocení mřížek z trénovací množiny, přičemž každé mřížce je přiřazen symbol umožňující identifikaci jednotlivých prvků. Tento automat pak již může být optimalizován jako běžný vážený konečný automat.

7.1.1 Efektivní výpočet racionální jádrové funkce

Pro výpočet jádrové funkce $K(U_k, U_j)$, kde U_k a U_j jsou acyklické vážené konečné akceptory odpovídající promluvám u_k, u_j, je použita kompozice $U_k \circ T \circ T^{-1} \circ U_j$ z rovnice (5.77). Algoritmus pro efektivní výpočet této kompozice je založen na vhodném pořadí kompozice a na optimalizaci mezivýsledků. Optimalizaci lze pro danou trénovací množinu $\mathcal{T}$ předpočítat a následně použít pro vyčíslení racionální jádrové funkce mezi neznámou promluvou U a všemi prvky trénovací množiny $U_j \in \mathcal{T}$.

Následuje popis algoritmu pro předzpracování trénovací množiny do podoby minimálního deterministického váženého konečného transduceru, který je následně použit pro rychlý výpočet racionální jádrové funkce:

Vstup:

- Množina mřížek $\mathcal{T} = \{U_k\}_{k=1}^{l}$
- Transducer T definující racionální jádrovou funkci

Výstup:

- Transducer R reprezentující prvky $\mathcal{T}$

Algoritmus předzpracování trénovací množiny do transduceru R:

1. Kompozice mřížky (akceptoru) U_k a transduceru T:

$$R_k = T^{-1} \circ U_k \tag{7.2}$$

   Tato kompozice je vyčíslena pro každý prvek trénovací množiny, neboť každý prvek této množiny může být potenciálním podpůrným vektorem při následném trénování skryté vrstvy, která je založena na SVM.

2. Projekce na R_k na vstupní symboly, tj. výpočet $\Pi_1(R_k)$.

3. Rozšíření abecedy symbolů $\Pi_1(R_k)$ o indexy promluv z trénovací množiny. Následně je provedena konkatenace akceptoru $\Pi_1(R_k) \otimes I(k)$, kde $I(k) = \{k\} \times \{k\}$. Jinými slovy dojde k přidání identifikátoru k-té promluvy na konec každé cesty akceptorem $\Pi_1(R_k)$. Tento krok je významný pro pozdější přiřazení hodnoty jádrové funkce konkrétnímu páru (U, U_k), bez přidání identifikátoru $I(k)$ by v následném kroku při determinizace automatu došlo ke sloučení cest se stejnými vstupními symboly, ale příslušející různým mřížkám.

4. Sjednocení všech těchto rozšířených akceptorů napříč všemi prvky U_k trénovací množiny $\mathcal{T}$, odstranění ϵ-přechodů, determinizace a minimalizace:

$$\begin{aligned} \bar{R} &= \min\left[\det\left[\operatorname{rmeps}\bigoplus_{k=1}^{l}\Pi_1(R_k)\otimes I(k)\right]\right] \\ &= \min\left[\det\left[\operatorname{rmeps}\bigoplus_{k=1}^{l}\Pi_1(T^{-1}\circ U_k)\otimes I(k)\right]\right] \end{aligned} \tag{7.3}$$

   Nyní je $\bar{R}$ minimální deterministický vážený konečný akceptor nad množinou $\mathcal{A} \cup \{k\}_{k=1}^{l}$.

5. Převedení akceptoru $\bar{R}$ na transducer R aplikací následujícího algoritmu:

   - Inicializace $R \leftarrow \bar{R}$
   - Pro všechny přechody e z akceptoru $\bar{R}$:
     - Je-li $i[e] \in \{k = 1, 2, \ldots, l\}$, pak $i[e] = \epsilon$.
     - Jinak $o[e] = \epsilon$.

Výsledný transducer R definuje relaci mezi řetězci akceptovanými transducerem T^{-1} a číselnými indexy k prvků trénovací množiny $\mathcal{T}$. Je nutné poznamenat, že tento transducer je minimální a deterministický vzhledem ke vstupním symbolům s výjimkou přechodů vedoucích do koncových stavů transduceru R, které po aplikaci výše uvedeného algoritmu mají vstupní symboly ϵ a výstupním symbolem je identifikátor prvku trénovací množiny k.

Transducery R a T definují parametry vstupní vrstvy. Zatímco T reprezentuje racionální jádrovou funkci, transducer R efektivním, minimálním a deterministickým způsobem uchovává prvky trénovací množiny. Výpočet jádrové funkce $K(U, U_k)\ \forall\ U_k \in \mathcal{T}$ mezi vstupní mřížkou U a všemi mřížkami trénovací množiny U_k je pak možné realizovat následujícím algoritmem:

Vstup:

- Mřížka U
- Transducer T a funkce ψ definující racionální jádrovou funkci
- Transducer R reprezentující prvky trénovací množiny $\mathcal{T}$

Výstup:

- Hodnoty racionální jádrové funkce $K(U, U_k)$ pro $k = 1, 2, \ldots l$

Algoritmus efektivního výpočtu racionální jádrové funkce:

1. Projekce kompozice $U \circ T$ na výstupní symboly, dále odstranění ϵ-přechodů a determinizace:
$$L = \det\left[\text{rmeps}\,\Pi_2(U \circ T)\right] \tag{7.4}$$
2. Výpočet kompozice $LR = L \circ R$.
3. Výpočet racionální jádrové funkce $K(U, U_k)$ procházením všech cest transducerem LR. Hodnota této funkce je rovna $\oplus$-sumě vah všech cest z počátečního do koncového stavu transduceru LR, které mají výstupní symbol k. Pro zobrazení z prostoru polookruhu $\mathbb{K}$ do reálných čísel je použita funkce ψ z rovnice (5.79):
$$K(U, U_k) = \psi\left(\bigoplus_{\substack{\pi \in \mathcal{P}(I_{LR}, \mathcal{F}_{LR}) \\ o[\pi]=k}} \lambda[p[\pi]] \otimes w[\pi] \otimes \rho[n[\pi]]\right) \tag{7.5}$$

Je důležité zdůraznit, že algoritmus procházející transducer vyčísluje celý vektor $K = [K(U, U_k)]_{k=1}^{l}$ naráz v jednom průchodu. Algoritmus počítá pouze ty prvky vektoru, pro které existuje řetězec symbolů přijímaný jak akceptorem $\Pi_2(U \circ T)$, tak akceptorem $\Pi_1(T^{-1} \circ U_k)$. Jinými slovy – jsou vyčíslovány pouze prvky $K(U, U_k)$, pro které je kompozice
$$U \circ T \circ T^{-1} \circ U_k \tag{7.6}$$
neprázdný transducer.

Pro ilustraci uveďme mezivýsledky algoritmu pro trénovací množinu sestávající se ze dvou fonémových mřížek U_1 a U_2 (obrázky 7.1 a 7.2). Všechny váhy vynesené v obrázcích jsou z pravděpodobnostního polookruhu, chybějící váha odpovídá váze 1. Obrázek 7.3 zobrazuje transducer T definující 3-gramovou racionální jádrovou funkci nad množinou symbolů (fonémů) mřížek U_1 a U_2. Na obrázku 7.4 je transducer R získaný aplikací výše uvedeného

algoritmu na trénovací množinu obsahující mřížky U_1 a U_2. Neviděnou mřížku U zachycuje obrázek 7.5. Akceptor L vygenerovaný z U je uveden na obrázku 7.6. Konečně výsledná kompozice určující hodnoty racionální 3-gramové jádrové funkce $K(U, U_1)$ a $K(U, U_2)$ je zachycena na obrázku 7.7.

7.1.2 Analýza výpočetní složitosti

Pro výpočet racionální jádrové funkce je nutné nejprve vyčíslit kompozici $U \circ T$, kde T je transducer definující racionální jádrovou funkci a U je mřížka z výstupu systému automatického rozpoznávání řeči. Akceptor U je acyklický automat, tj. délka cest π tímto akceptorem je vždy shora omezená, tj. $|\pi| < +\infty \quad \forall \pi \in \mathcal{P}_U(I, \mathcal{F})$.

Transducer T však obecně acyklický není, předpokládejme však, že T neobsahuje cykly tvořené přechody se vstupním symbolem ϵ. Pak platí, že pro libovolný řetězec x, který tento transducer převádí na řetězec y, platí: $|y| \leq |x| + k \ \forall x, y$, kde $k < +\infty$ je počet symbolů z y, které vznikly z přechodů se vstupním symbolem ϵ. Tuto nerovnost lze popsat také tak, že při převodu řetězce x pomocí transduceru T na řetězec y bylo použito:

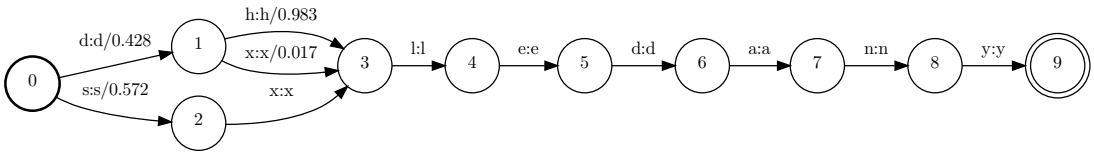
- k přechodů se vstupním symbolem ϵ,
- $|x| - |y| + k$ přechodů s výstupním symbolem ϵ,
- $|y| - k$ přechodů, které nemají vstupní ani výstupní symbol ϵ.

V případě, že T neobsahuje žádné přechody se vstupním symbolem ϵ, je $k = 0$.

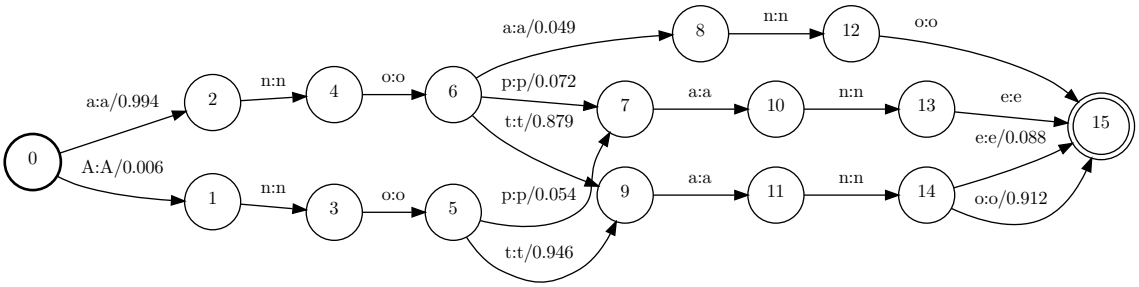
Pro kompozici $U \circ T$ potom platí, že i *výsledek kompozice je acyklický konečný automat*, protože libovolný řetězec přijímaný U je zároveň vstupní řetězec T a tudíž $+\infty > |u| = |x| + k \geq |y|$. Poznamenejme, že předpoklad o neexistenci cyklů se vstupními symboly ϵ v transduceru T je splněn jak pro n-gramové racionální jádrové funkce používané v této práci, tak i pro další druhy racionálních jádrových funkcí, např. mismatch-kernel [10] nebo Hausslerovy konvoluční jádrové funkce [98] (viz též kapitola 5.3.3).

Algoritmická složitost operace kompozice vážených konečných transducerů $T_1 \circ T_2$ má časovou složitost $\mathcal{O}(|\mathcal{Q}_1| \cdot |\mathcal{Q}_2| \cdot D_1(\log D_2 + M_2))$, kde $|\mathcal{Q}_1|$, $|\mathcal{Q}_2|$ je počet stavů transducerů T_1, resp. T_2, D_1 je maximální počet přechodů z jednoho stavu a M_2 je maximální počet přechodů z jednoho stavu označených stejným vstupním symbolem, přičemž se předpokládá, že automat T_1 má přechody neuspořádané a T_2 má přechody uspořádané vzhledem k vstupním uzlům, tj. pro libovolný výstupní symbol T_1 je možné odpovídající vstupní symbol T_2 dohledat v čase $\mathcal{O}(\log D_2)$ [91, 92]. Pro analýzu kompozice $U \circ T$ nás zajímá časová složitost pouze vzhledem k parametrům akceptoru U. Z výše uvedeného vyplývá, že asymptotická časová složitost kompozice je $\mathcal{O}(|\mathcal{Q}_U|D_U)$. Paměťová složitost kompozice $T_1 \circ T_2$ je $\mathcal{O}(|\mathcal{Q}_1| \cdot |\mathcal{Q}_2| \cdot D_1 M_2)$ [91] a pro případ kompozice vstupní mřížky U s transducerem T pak opět $\mathcal{O}(|\mathcal{Q}_U|D_U)$.

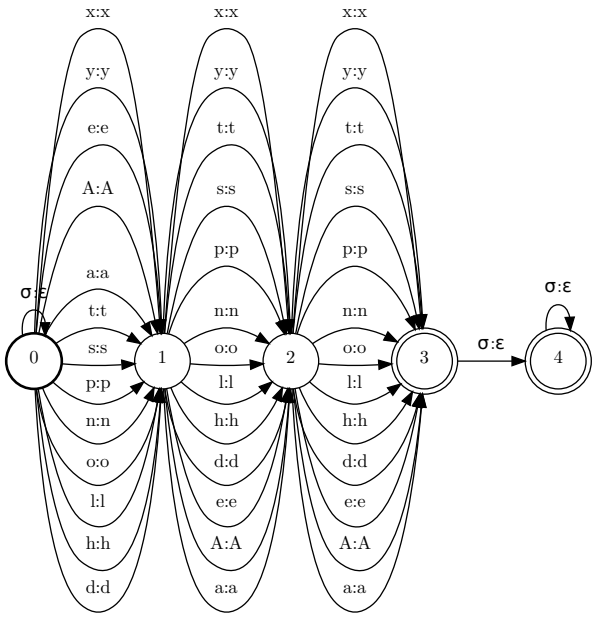
Další z operací nutných pro vyčíslení jádrové funkce je odstranění ϵ-přechodů. Tento algoritmus má pro acyklické automaty, což je i případ akceptoru $A = \Pi_2(U \circ T)$, složitost $\mathcal{O}(|\mathcal{Q}_A|^2 + |\mathcal{Q}_A| \cdot |\mathcal{E}_A|)$, kde $|\mathcal{E}_A|$ počet přechodů akceptoru A. Asymptotická paměťová složitost je pak $\mathcal{O}(|\mathcal{Q}_A| \cdot |\mathcal{E}_A|)$ [93]. Poznamenejme, že výstupem algoritmu odstranění ϵ-přechodů pro acyklický akceptor je opět acyklický akceptor.



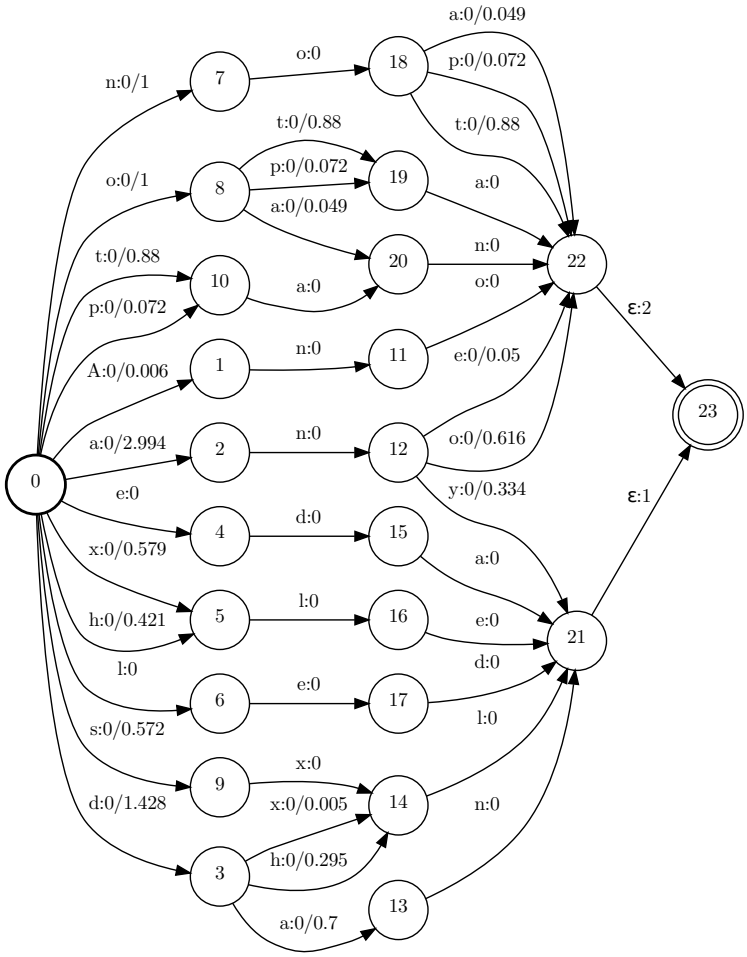
Obrázek 7.1: Fonémová mřížka promluvy U_1



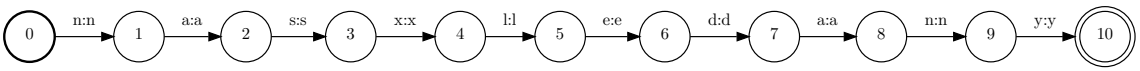
Obrázek 7.2: Fonémová mřížka promluvy U_2



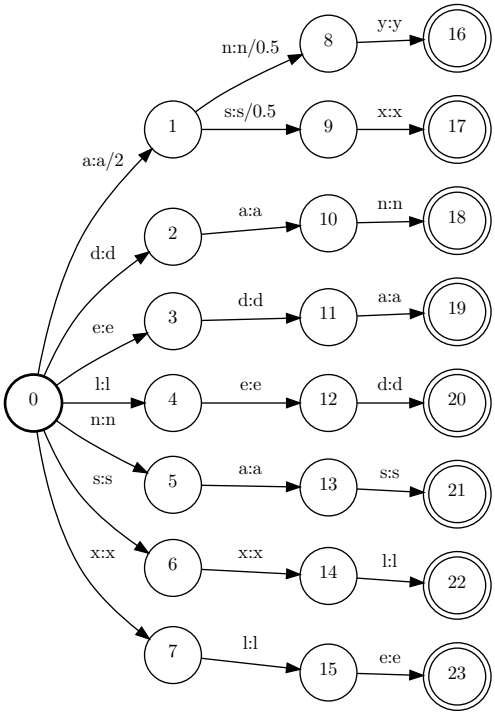
Obrázek 7.3: Transducer T definující racionální jádrovou funkci.



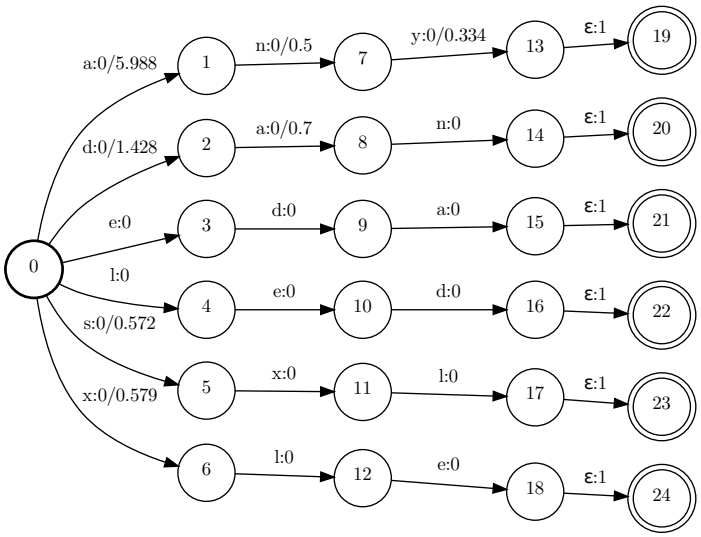
Obrázek 7.4: Transducer R, přechody ze stavů 21 a 22 do stavu 23 jsou přechody odkazující na mřížku U_1, resp. U_2.



Obrázek 7.5: Fonémová mřížka promluvy U.



Obrázek 7.6: Akceptor $L = \det[\text{rmeps}\,\Pi_2(U \circ T)]$



Obrázek 7.7: Transducer $L \circ R$. Je zřejmé, že mřížka U má společné n-gramy pouze s mřížkou U_1 – všechny cesty transducerem mají výstupní symbol 1 reprezentující mřížku U_1. Hodnota jádrové funkce $K(U, U_1) = 5.988 \cdot 0.5 \cdot 0.334 + 1.428 \cdot 0.7 + 1 + 1 + 0.572 + 0.579 = 5.15$.

Operace	Časová	Paměťová
$A = \Pi_2(U \circ T)$	$\mathcal{O}(\lvert\mathcal{Q}_U\rvert \cdot D_U)$	$\mathcal{O}(\lvert\mathcal{Q}_U\rvert \cdot D_U)$
$B = \text{rmesp } A = \text{rmesp } [U \circ T]$	$\mathcal{O}(\lvert\mathcal{Q}_A\rvert^2 + \lvert\mathcal{Q}_A\rvert \cdot \lvert\mathcal{E}_A\rvert)$	$\mathcal{O}(\lvert\mathcal{Q}_A\rvert \cdot \lvert\mathcal{E}_A\rvert)$
$L = \det B = \det \text{rmesp } [U \circ T]$	$\mathcal{O}(\lvert\mathcal{Q}_U\rvert \cdot \lvert\mathcal{E}_U\rvert)$	$\mathcal{O}(\lvert\mathcal{Q}_U\rvert \cdot \lvert\mathcal{E}_U\rvert)$
$L \circ R$	$\mathcal{O}(\lvert\mathcal{Q}_L\rvert \cdot D_L)$	$\mathcal{O}(\lvert\mathcal{Q}_L\rvert \cdot D_L)$

Tabulka 7.1: Asymptotická časová a paměťová složitost jednotlivých operací algoritmu výpočtu racionální jádrové funkce ve vstupní vrstvě HDM pro n-gramovou racionální jádrovou funkci.

Pro algoritmus determinizace je dále nutné studovat i podmínky, za nichž se algoritmus zastaví. Citujme zde závěr například z [8] říkající, že je-li automat acyklický a polookruh, nad nímž je definován nemá nulový součet (angl. zero-sum-free semiring), tj. polookruh splňuje

$$a \oplus b = \bar{0} \;\Rightarrow\; a = b = \bar{0} \quad \forall a, b \tag{7.7}$$

pak je automat determinizovatelný, algoritmus determinizace se zastaví. Výše uvedená podmínka pro automat $\Pi_2(U \circ T)$ definovaný nad logaritmickým (ale i tropickým a pravděpodobnostním) polookruhem platí, tudíž tento automat je determinizovatelný. Algoritmická a paměťová složitost determinizace je pro obecný konečný akceptor exponenciální. Nicméně pro transducer T reprezentující n-gramovou racionální jádrovou funkci je výsledek kompozice $U \circ T$ podmnožinou faktorového automatu. Například pro $n = 5$ projekce kompozice $\Pi_2(U \circ T)$ akceptuje podřetězce řetězců akceptovaných U o délce 5.

Je-li x řetězec nad abecedou $\mathcal{A}^*$, pak konstrukce faktorového automatu $F(x)$ je lineární v čase s délkou řetězce x a počet stavů $F(x)$ je rovněž lineární s délkou řetězce x [111]. Tento závěr se dále nechá zobecnit pro případ obecného acyklického konečného automatu [69, 70, 86]. Použijeme-li tento závěr, tj. že $|F(U)|$ je lineární s $|U|$, pak velikost automatu L jako výsledku determinizace musí opět být lineární s $|U|$ (pro racionální n-gramovou jádrovou funkci). Vzhledem k algoritmu pro determinizaci konečných automatů (kapitola 5.2.3) pak musí algoritmus skončit rovněž v čase lineárním s $|U|$, neboť počet stavů $F(U)$ je roven počtu vážených podmnožin p' vytvářených v průběhu determinizace a pro každý stav $\text{rmeps}\,\Pi_2(U \circ T)$ vznikne nejvýše $|\mathcal{B}|$ podmnožin p'.

Poslední krok algoritmu spočívá v kompozici $L \circ R$. Zde je analýza složitosti obdobná jako u kompozice $U \circ T$, tj. jak paměťová, tak časová složitost je pro pevné R asymptoticky $\mathcal{O}(|\mathcal{Q}_L| \cdot D_L)$.

Celkový přehled o asymptotické časové a paměťové složitosti dává tabulka 7.1. Z této tabulky a z předchozí analýzy pro případ n-gramových racionálních jádrových funkcí vyplývá, že paměťová složitost, tj. počet stavů a přechodů roste lineárně s počtem stavů a přechodů mřížky U a časová složitost není horší než kvadratická. Poznamenejme, že asymptotická složitost vyčísluje odhad složitosti pro $|U| \to +\infty$, nicméně v úloze rozpoznávání řeči v hlasových dialogových systémech bývá velikost mřížky U omezená, proto nás kromě asymptotické složitosti bude zajímat i časová a paměťová složitost na konkrétních úlohách. Této problematice je věnována celá kapitola 10.1.

$\|\mathbf{t}\|$	*počet*	*různých*	F [%]
1	581	14	87.5
2	560	21	77.6
3	164	12	53.1
4	35	3	49.0

Tabulka 7.2: Přesnost predikce vyjádřená F-mírou (kapitola 9.3) pro sémantické n-tice různé délky $|\mathbf{t}|$. Sloupec *počet* říká, kolik pozitivních příkladů sémantických n-tic dané délky se v datech vyskytuje, sloupec *různých* pak udává počet různých sémantických n-tic dané délky. Čísla byla určena na development sadě korpusu HHTT (kapitola 9.1) za využití skryté vrstvy HDM modelu.

7.2 Skrytá vrstva

Jak již bylo řečeno, HDM je rozšířením modelu STC, přičemž jako skrytá vrstva modelu HDM je použit model STC s několika modifikacemi. Tento model oproti původní implementaci zmiňované v [59] používá klasifikátory sémantických n-tic, přičemž délky těchto n-tic jsou voleny od jedné až po $n_{\max}$. Takto definované sémantické n-tice se překrývají a výstupy binárních klasifikátorů jsou silně korelované, např. pokud je v trénovacím stromu s_i přítomna n-tice (DEPARTURE, TO, STATION), pak jsou v tomto trénovacím stromu přítomny i n-tice (DEPARTURE, TO), (TO, STATION), (DEPARTURE), (TO) a (STATION) a každé z nich odpovídá právě jeden binární klasifikátor.

Experimenty (tabulka 7.2) ukázaly, že zpravidla čím menší je délka n odpovídající sémantické n-tice, tím robustnější klasifikace je možné dosáhnout. V původním STC modelu však krátké n-tice (především pro $n = 1$) přináší problém s nejednoznačně rekonstruovatelným sémantickým stromem.

Protože již první úroveň sémantického stromu může obsahovat více sémantických konceptů – například anotace TIME, TO(STATION) – je do každého sémantického stromu navíc vložen nový kořenový uzel S. Z výše zmíněné anotace se pak stane S(TIME, TO(STATION)) a jsou zde obsaženy následující sémantické n-tice: (S), (TIME), (TO), (STATION), (S, TIME), (S, TO), (TO, STATION), (S, TO, STATION). Kořenový uzel je vkládán při generování množiny $\mathcal{S}$ (viz níže) a jeho vložení umožňuje rozlišit, zda daný sémantický koncept se v sémantickém stromu vyskytuje jako přímý následovník S nebo je zanořen hlouběji ve stromu.

Označme $\mathcal{S} = \{\mathbf{t} \in s_i : s_i \in \mathcal{T}\}$ množinu všech sémantických n-tic $\mathbf{t}$ získaných ze stromů s_i trénovací množiny $\mathcal{T}$. Experimenty s přesností klasifikace výskytu sémantických n-tic ukázaly, že přesnost klasifikace málo četných n-tic je nízká, proto pro trénování skryté vrstvy HDM modely jsou vybrány pouze n-tice četnější než definovaný práh N. Označme $\mathrm{cnt}(\mathcal{T}, \mathbf{t})$ počet výskytů n-tice $\mathbf{t}$ ve stromech z trénovací množiny $\mathcal{T}$. Pro další trénování je pak zvolena pouze podmnožina:

$$\mathcal{S}_N = \{\mathbf{t} \in \mathcal{S} : \mathrm{cnt}(\mathcal{T}, \mathbf{t}) \geq N\}' \tag{7.8}$$

Typický histogram zobrazující četnosti sémantických n-tic je zobrazen na obrázku 10.13, str. 137.

Pro každý prvek $\mathbf{t} \in \mathcal{S}_N$ definujme množinu pozitivních a negativních trénovacích příkladů:

$$\begin{aligned}
\mathcal{T}_{\mathbf{t}}^{+} &= \{(U_k, +1) : \mathbf{t} \in s_k \wedge (U_k, s_k) \in \mathcal{T}\} \quad \mathbf{t} \in \mathcal{S}_N, k = 1, 2, \ldots, l \\
\mathcal{T}_{\mathbf{t}}^{-} &= \{(U_k, -1) : \mathbf{t} \notin s_k \wedge (U_k, s_k) \in \mathcal{T}\} \quad \mathbf{t} \in \mathcal{S}_N, k = 1, 2, \ldots, l \\
\mathcal{T}_{\mathbf{t}} &= \mathcal{T}_{\mathbf{t}}^{+} \cup \mathcal{T}_{\mathbf{t}}^{-}
\end{aligned} \tag{7.9}$$

Samotná skrytá vrstva HDM se skládá z binárních SVM klasifikátorů:

$$f_{\mathbf{t}}(U, \alpha(\mathbf{t})) : \mathcal{U} \to \{-1, +1\}, \quad \mathbf{t} \in \mathcal{S}_N, \quad U \in \mathcal{U} \tag{7.10}$$

získaných optimalizací kritéria (5.22) nad trénovací množinou $\mathcal{T}_{\mathbf{t}} = \mathcal{T}_{\mathbf{t}}^{+} \cup \mathcal{T}_{\mathbf{t}}^{-}$. Množina $\mathcal{U}$ značí množinu všech možných acyklických automatů – reprezentuje prostor všech mřížek na výstupu systému automatického rozpoznávání řeči. Jako jádrové funkce při trénování a predikci je použita racionální jádrová funkce $K(U_i, U_j)$.

Poznamenejme, že pro predikci dosud neviděné promluvy U je nutné nejprve vyčíslit racionální jádrovou funkci $K(U, U_k)$, $\forall U_k \in \mathcal{T}$ a následně tyto hodnoty použít pro klasifikaci přítomnosti n-tice $\mathbf{t}$ ve výstupním stromu $\hat{s}$. Označme nyní $d_{\mathbf{t}}$ vzdálenost promluvy k rozhodovací nadrovině $H_{\mathbf{t}}$ klasifikátoru klasifikujícího přítomnost sémantické n-tice $\mathbf{t}$:

$$d_{\mathbf{t}} = \sum_{k=1}^{l} \alpha_k^{\mathbf{t}} y_k^{\mathbf{t}} K(U, U_k) + b^{\mathbf{t}} \tag{7.11}$$

Podle rozhodovacího pravidla definovaného v rovnici (5.9) pak můžeme určit, zda se sémantická n-tice $\mathbf{t}$ vyskytuje ($\hat{y}_{\mathbf{t}} = 1$) nebo nevyskytuje ($\hat{y}_{\mathbf{t}} = -1$) v sémantickém stromu odpovídajícím promluvě U:

$$\hat{y}_{\mathbf{t}} = f_{\mathbf{t}}(U, \alpha(\mathbf{t})) = \operatorname{sgn} d_{\mathbf{t}} = \operatorname{sgn}\left[\sum_{k=1}^{l} \alpha_k^{\mathbf{t}} y_k^{\mathbf{t}} K(U, U_k) + b^{\mathbf{t}}\right] \tag{7.12}$$

kde $\alpha(\mathbf{t}) = (\alpha_1^{\mathbf{t}}, \ldots, \alpha_l^{\mathbf{t}}, y_1^{\mathbf{t}}, \ldots, y_l^{\mathbf{t}}, b^{\mathbf{t}})$ jsou parametry binárního klasifikátoru sémantické n-tice $\mathbf{t}$ – hodnoty parametrů $\alpha_k^{\mathbf{t}}$ určují normálu oddělující nadroviny $H^{\mathbf{t}}$ v prostoru $\mathcal{H}^{\mathbf{t}}$ (rovnice (5.15)), hodnoty $y_k^{\mathbf{t}} \in \mathcal{T}_{\mathbf{t}}$ jsou cílové třídy promluv U_k v množině $\mathcal{T}_{\mathbf{t}}$ a $b^{\mathbf{t}}$ posun oddělující nadroviny vzhledem k počátku.

Poznamenejme, že každý z binárních SVM klasifikátorů má navíc jeden metaparametr $C^{\mathbf{t}}$, který regularizuje klasifikátor sémantické n-tice $\mathbf{t}$. Možné způsoby nastavování tohoto metaparametru pro jednotlivé klasifikátory $f_{\mathbf{t}}$ budou diskutovány a vyhodnoceny v kapitole 10.2.2, strana 135.

7.3 Výstupní vrstva

Problém s nejednoznačností sémantického stromu při použití sémantických n-tic pro malá n je možné vyřešit použitím další vrstvy diskriminativních klasifikátorů. Na abstraktní sémantický strom je možné se dívat jako na derivační strom gramatiky. Abstraktní sémantické stromy jsou však, obdobně jako u STC modelu, uvažovány jako neuspořádané – neexistuje v nich relace uspořádání mezi následovníky jednoho uzlu. Derivační pravidla popisující generování takového stromu tedy musí tuto vlastnost brát v úvahu.

Proto každé z derivačních pravidel předpokládejme ve tvaru $A \to \beta\ (p)$, kde A je sémantický koncept, β je množina sémantických konceptů – následovníků sémantického konceptu A a $p = \mathrm{P}(A \to \beta|u)$ je pravděpodobnost realizace derivace při dané promluvě u.

Jako *sémantickou gramatiku promluvy* u nazývejme trojici $G_u = (\Theta, \mathcal{R}_u, S)$, kde:

- Θ je množina sémantických konceptů,
- $\mathcal{R}_u$ je množina derivačních pravidel ve tvaru $A \to \beta\ (p)$, kde $A \in \Theta$. Toto pravidlo říká, že ve stromu s pro vstupní promluvu u se vyskytuje uzel označený konceptem A s potomky β s pravděpodobností p:

$$p = \mathrm{P}(A \to \beta|u) = \mathrm{P}(\beta|A, u) \tag{7.13}$$

  Platí:

$$\sum_{\beta} \mathrm{P}(A \to \beta|u) = \sum_{\beta} \mathrm{P}(\beta|A, u) = 1 \tag{7.14}$$

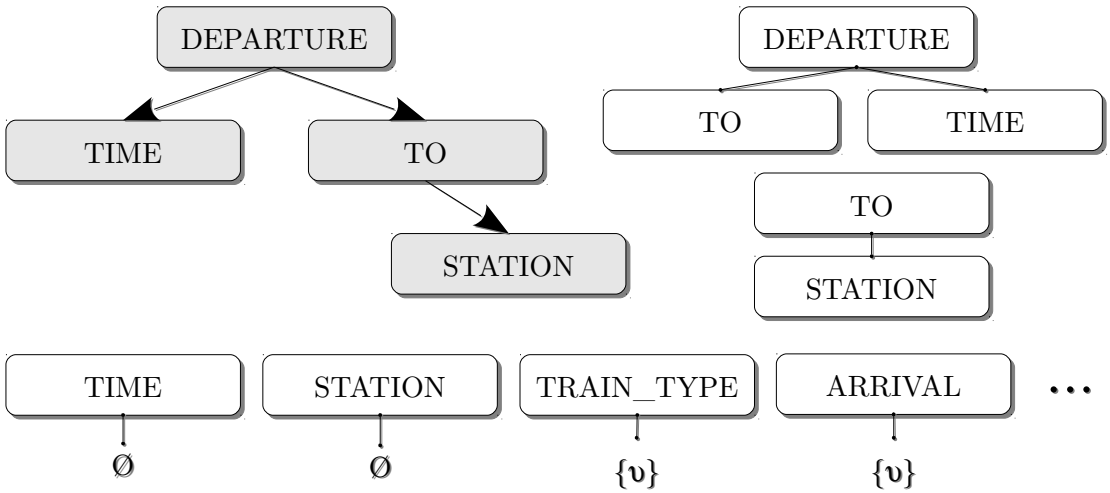
- Množina β je podmnožinou množiny všech sémantických konceptů Θ doplněná o speciální symbol ν (viz níže).

$$\beta \subset \{\nu\} \cup \Theta \tag{7.15}$$

- $S \in \Theta$ je kořenový sémantický koncept, startovací symbol gramatiky G_u.

Tato definice sémantické gramatiky promluvy u je obdobná definici bezkontextových gramatik (kapitola 5.4) s následujícími rozdíly:

- Symboly gramatiky nejsou děleny na terminální a neterminální, existuje pouze jediná množina sémantických konceptů Θ.



Obrázek 7.8: Sémantický strom z obrázku 5.16 dekomponovaný na pravidla sémantické gramatiky R. Pro všechny ostatní sémantické koncepty nevyskytující se v tomto stromu platí pravidlo $A \to \{\nu\}$, např. TRAIN_TYPE $\to \{\nu\}$.

- Pravděpodobnosti přiřazené pravidlům sémantické gramatiky jsou parametrizovány promluvou u. Tímto se do sémantické gramatiky zavádí lexikalizace obdobně jako u lexikalizovaných bezkontextových gramatik popsaných v kapitole 5.4 nebo v [101].
- Pravidlo ve tvaru $A \to \emptyset$ (p) znamená, že s pravděpodobností p koncept A je v abstraktním sémantickém stromu pro promluvu u listovým uzlem (nemá potomky).
- Pravidlo ve tvaru $A \to \{\nu\}$ (p) znamená, že s pravděpodobností p se koncept A nenachází v abstraktním sémantickém stromu pro promluvu u.

Předpokládejme nyní, že pro danou promluvu u jsme získali sémantickou gramatiku $G_u = (\Theta, \mathcal{R}_u, S)$. Cílem dekódovacího algoritmu je z této gramatiky vygenerovat nejpravděpodobnější abstraktní sémantický strom.

Pojmem *částečný abstraktní sémantický strom* budeme rozumět posloupnost pravidel $r = (r_1, r_2, \dots r_n)$, $r_i \subseteq \mathcal{R}_u$, kde r_1 je ve tvaru $S \to \beta$ a pro libovolný koncept A_i takový, že $r_i = A_i \to \beta_i$, existuje pravidlo $r_j = A_j \to \beta_j, 1 \leq j < i$ takové, že $A_i \in \beta_j$. Jinými slovy, koncept A_i je do derivačního stromu přidán pravidlem r_j a jeho potomci jsou určeni pravidlem r_i, přičemž musí platit $j < i$. Částečný sémantický strom odpovídá mezivýsledku při postupném generování derivačního stromu (popsáno k kapitole 7.3.1).

Množinu sémantický konceptů v částečném stromu r označme jako Θ_r:

$$\Theta_r = \bigcup_{A \to \beta \in r} \beta \tag{7.16}$$

Dále označme $L(A)$ jako počet prvků r_i takových, že $r_i = A \to \beta$ a $R(A)$ počet prvků r_i takových, že $r_i = C \to \beta,\ A \in \beta$. Jinými slovy $L(A)$ říká, kolikrát se ve stromu r objevil sémantický koncept A na levé straně nějakého pravidla. $R(A)$ pak vyčísluje, kolikrát se sémantický koncept A objevil na pravé straně nějakého pravidla.

Nyní můžeme označit množinu $\mathcal{E}_r$ expandovaných sémantických konceptů v částečném stromu r jako podmnožinu všech sémantických konceptů, pro které platí $L(A) = R(A)$:

$$\mathcal{E}_r = \{A : A \to \beta \in r \wedge L(A) = R(A)\} \tag{7.17}$$

Množinu neexpandovaných sémantických konceptů v částečném stromu r jako $\mathcal{M}_r$:

$$\mathcal{M}_r = \Theta_r \setminus \mathcal{E}_r \tag{7.18}$$

O částečném sémantickém stromu r budeme říkat, že je plně expandovaný, pokud $\mathcal{M}_r = \emptyset$.

Definujme množinu všech možných následovníků konceptu A jako $\mathcal{B}_A$:

$$\mathcal{B}_A = \{\beta : A \to \beta \in \mathcal{R}_u\} \tag{7.19}$$

Pak $\mathrm{P}(A \to \beta|u)$ je pravděpodobnostní rozdělení nad množinou $\mathcal{B}_A$ podmíněné sémantickým konceptem A a vstupní promluvou u.

Nakonec definujme pravděpodobnost částečného sémantického stromu $\mathrm{P}(r|u)$ jako součin pravděpodobností

$$\mathrm{P}(r|u) = \prod_{A \to \beta \in r} \mathrm{P}(A \to \beta|u) \tag{7.20}$$

7.3.1 Algoritmus určení abstraktního sémantického stromu

Pro určení nejpravděpodobnějšího abstraktního sémantického stromu ze sémantické gramatiky promluvy u je použit algoritmus prohledávání s nejmenší cenou. Cena $\text{cost}(r)$ částečného sémantického stromu r je definována jako záporný logaritmus pravděpodobnosti $\text{P}(r|u)$:

$$\begin{aligned}\text{cost}(r) &= -\ln \text{P}(r|u) = -\ln \prod_{A\to\beta\in r} \text{P}(A\to\beta|u) \\ &= \sum_{A\to\beta\in r} -\ln \text{P}(A\to\beta|u)\end{aligned} \tag{7.21}$$

Pro hledání je použita standardní podoba algoritmu hledání s nejmenší cenou [112]:

Vstup:

– Sémantická gramatika promluvy u: $G_u = (\Theta, \mathcal{R}_u, S)$

Výstup:

– Plně expandovaný sémantický strom r

Algoritmus určení abstraktního sémantického stromu:

1. Nastav OPEN = [ ]
2. Pro všechna pravidla $r_1 \in \mathcal{R}_u : r_1 = S \to \beta$ přidej r_1 do seznamu OPEN.
3. Opakuj dokud je seznam OPEN neprázdný:
   (a) Najdi v OPEN částečný strom r s nejnižší $\text{cost}(r)$ a odeber r z OPEN
   (b) Je-li $\mathcal{M}_r = \emptyset$, pak ukonči algoritmus **s návratovou hodnotou** r
   (c) Jinak přiřaď do A libovolný prvek $\mathcal{M}_r$
   (d) Pro každé pravidlo $r_i \in \mathcal{R}_u : r_i = A \to \beta$ vytvoř nový strom r', který vznikne z r přidáním r_i na konec a přidej r' do seznamu OPEN.

Poznamenejme, že pravidla $A \to \emptyset$ a $A \to \{\nu\}$ slouží k definování vazby mezi sémantickým stromem a lexikální realizací promluvy u. Pomocí těchto pravidel je definována pravděpodobnost, že daný sémantický koncept se v promluvě vyskytuje, resp. nevyskytuje. Navíc pokud zvolíme množinu pravidel $\mathcal{R}_u$ tak, že neobsahuje pravidla ve tvaru $\nu \to \beta$, dosáhneme tím nemožnosti plně expandovat stromy obsahující pravidlo $A \to \{\nu\}$ a dekódovaný abstraktní sémantický strom tak nemůže obsahovat uzly ν.

Výše zmíněný algoritmus umožňuje snadné rozšíření na verzi, která vrací n-nejlepších abstraktních sémantických stromů. Stačí v kroku 3(b) neukončit algoritmus, ale pokračovat v generování dalších plně expandovaných sémantických stromů r. Pro získání pravděpodobnostní distribuce $\text{P}(\mathsf{C} = r|\mathsf{U} = u)$ nad těmito n-nejlepšími sémantickými stromy je nutné nejprve říci, že pravděpodobnost $\text{P}(r|u)$ v rovnici (7.20) lze vyčíslit i pro plně neexpandované sémantické stromy. Nelze je tedy přímo použít k přiřazení aposteriorní pravděpodobnosti. Pro vyčíslení pravděpodobnostního rozdělení nad n-nejlepšími sémantickými stromy $r_1, r_2, \ldots r_n$ proto použijeme aproximaci:

$$\text{P}(\mathsf{C} = r_i|\mathsf{U} = u) \approx \frac{\text{P}(r_i|u)}{\sum_{k=1}^{n} \text{P}(r_k|u)} \tag{7.22}$$

Model P(C|U) pak lze přímo použít jako konceptový model v diskriminativním modelu pro porozumění řeči z kapitoly 6.

7.3.2 Omezení na sémantické stromy generované HDM

Ze způsobu, jakým je definována sémantická gramatika pro promluvu u vyplývá množina omezení na třídu derivačních stromů generovaných HDM:

1. Abstraktní sémantické stromy jsou neuspořádané, tj. HDM, podobně jako STC, neumožňuje diskriminovat mezi dvěma abstraktními sémantickými anotacemi lišícími se pouze pořadím sémantických konceptů:

   DEPARTURE(FROM(STATION), TIME)
   DEPARTURE(TIME, FROM(STATION))

2. Více výskytů jednoho sémantického konceptu A v různých uzlech sémantického stromu sdílí stejné derivační pravidlo $A \rightarrow \beta$ a tudíž i stejnou pravděpodobnost derivace $\mathrm{P}(A \rightarrow \beta|u)$. Například jedna gramatika modeluje pouze jedinou pravděpodobnost $\mathrm{P}(\text{STATION} \rightarrow \emptyset|u)$, přestože abstraktní sémantická anotace obsahuje dva uzly se sémantickým konceptem STATION:

   DEPARTURE(FROM(STATION), TO(STATION))

3. Sémantická gramatika neumožňuje diskriminovat mezi dvěma abstraktními sémantickými anotacemi lišícími se pouze počtem uzlů se stejným sémantickým konceptem a stejným rodičovským uzlem, např. abstraktní sémantické anotace:

   DEPARTURE(TIME, FROM(STATION), TIME)
   DEPARTURE(FROM(STATION), TIME)

   Stromy odpovídajícím těmto anotacím v obou případech obsahují pravidlo DEPARTURE $\rightarrow$ {TIME, FROM} (p).

   Možným řešením, jak se vyhnout tomuto omezení, je použití přístupu z STC modelu [59], kde jsou používány indexované koncepty pro odlišení různých uzlů sémantického stromu se stejným konceptem, např:

   DEPARTURE(TIME-1, FROM(STATION), TIME-2)

4. Sémantická gramatika neumožňuje generovat abstraktní sémantické stromy s uzly obsahujícími stejný sémantický koncept, ale lišícími se následovnickými uzly, např:

   ARRIVAL(FROM), DEPARTURE(FROM(STATION), TO(STATION))

   Toto omezení v sobě zahrnuje nemožnost generovat abstraktní sémantické stromy, kde pro některý uzel se sémantickým konceptem A jeho podstrom rekurzivně obsahuje A, např.:

ACCEPT(DEPARTURE(ACCEPT(STATION)))

Poznamenejme, že obzvlášť druhý případ by neměl být povolen anotačním schématem, neboť si lze velmi obtížně představit lexikální realizaci takové promluvy.

I přes výše uvedená omezení lze konstatovat, že nejsou na překážku dobře pracujícímu modelu porozumění, neboť jejich důsledkům lze předcházet vhodným návrhem anotačního schématu. Mnohá tato omezení jsou společná s modelem STC a vyplývají především ze způsobu reprezentace vstupní promluvy ve formě příznakového vektoru obsahujícího četnost jednotlivých n-gramů (více v [59]).

Pro experimentální ověření byly použity dva sémantické korpusy – HHTT (kapitola 9.1) a TIA (kapitola 9.2). Anotační schéma sémantického korpusu TIA bylo navrženo již s ohledem na tato omezení. U sémantického korpusu HHTT pak bylo nutné upravit stromy porušující omezení 3., tj. potomci jednoho uzlu sémantického stromu označení stejnými koncepty byly sloučeni do jednoho uzlu a množiny jejich potomků byly sjednoceny. Stromy porušující omezení 4. se v sémantickém korpusu objevovaly pouze v jednotkách případů a při trénování výstupní vrstvy byly odpovídající trénovací příklady vyloučeny z trénovací množiny.

7.3.3 Určení množiny pravidel $\mathcal{R}_u$

Sémantická gramatika využívá množinu pravidel $\mathcal{R}_u$ zahrnující pro každé pravidlo pravděpodobnostní rozdělení $\mathrm{P}(A \to \beta|u)$ podmíněné promluvou u. V modelu HDM je nejprve určena množina $\mathcal{R}$ obsahující všechny možné expanze sémantických konceptů vyskytující se v trénovací množině $\mathcal{T}$ a následně je použita sada diskriminativních klasifikátorů pro přiřazení aposteriorní pravděpodobnosti $\mathrm{P}(A \to \beta|u)$ těmto pravidlům a tedy k získání parametrizované množiny pravidel $\mathcal{R}_u$:

Vstup:

– Trénovací množina $\mathcal{T}$

Výstup:

– Množina $\mathcal{R}$ pravidel vyskytujících se v trénovací množině

Algoritmus určení množiny $\mathcal{R}$:

1. Inicializace $\mathcal{R} = \emptyset$.
2. Pro každý strom $s_k \in \mathcal{T}$ projdi všechny uzly A stromu s_k a získej množinu následovníků β. Nemá-li A následovníky, použij $\beta = \emptyset$. Do $\mathcal{R}$ přidej pravidlo $A \to \beta$.
3. Pro každý sémantický koncept $A \in \Theta$ přidej do $\mathcal{R}$ pravidlo $A \to \{\nu\}$.

Nyní množina $\mathcal{R}$ obsahuje všechny sémantické koncepty A a ke každému sémantickému konceptu množiny jeho následovníků napříč všemi stromy v trénovacích datech. Nad touto množinou je nyní nutné definovat pravděpodobnostní rozložení $\mathrm{P}(A \to \beta|u)$. Poněvadž trénovací množina $\mathcal{T}$ je konečná a všechny sémantické stromy jsou konečné, je konečný i počet různých derivací konceptu A, tj. $|\mathcal{B}_A| < \infty \quad \forall A \in \Theta$.

Podmíněnou pravděpodobnost $\mathrm{P}(A \rightarrow \beta|u) = \mathrm{P}(\beta|A,u), \beta \in \mathcal{B}_A$ je možné odhadnout pomocí metod strojového učení. Tato pravděpodobnost je podmíněna sémantickým konceptem A a vstupní promluvou u. Vyplývá z toho tedy potřeba natrénovat pro každý koncept $A \in \Theta$ jeden klasifikátor diskriminující mezi prvky $\mathcal{B}_A$ a mající na vstupu příznakový vektor odvozený od promluvy u, potažmo od mřížky U.

Klasifikátor pro koncept A na základě příznakového vektoru získaného z mřížky U provádí klasifikaci do jedné ze tříd $\beta \in \mathcal{B}_A$. Musí ale zároveň poskytovat odhad aposteriorní pravděpodobnosti pro všechny cílové třídy, což je právě hledaná podmíněná pravděpodobnost $\mathrm{P}(A \rightarrow \beta|u)$.

Možných klasifikačních metod, které podporují klasifikaci do více tříd s odhadem aposteriorní pravděpodobnosti jednotlivých tříd je celá řada. V této práci byly použity SVM implementující klasifikaci do více cílových tříd a odhad aposteriorních pravděpodobností (kapitoly 5.1.5, 5.1.6) [76], nicméně je možné použít i jiné typy klasifikátorů, například dopředné perceptronové neuronové sítě s výstupní aktivační funkcí softmax [113] nebo lesy znáhodněných rozhodovacích stromů [114].

Klasifikátor poskytující odhad aposteriorní pravděpodobnosti jednotlivých derivací uzlu A má na vstupu příznakový vektor, který je získán ze vstupní mřížky U. Je možné opět použít lexiko-syntaktické příznaky, obdobně jako u modelu STC, nicméně experimenty prokázaly vhodnost použít hierarchickou architekturu, kde skrytá vrstva je realizována jako STC, nicméně neprovádí přímo klasifikaci jednotlivých sémantických n-tic, ale pouze určuje vzdálenost k jednotlivým rozhodovacím nadrovinám $H^{\mathbf{t}}$. Pro promluvu u reprezentovanou mřížkou U je pak příznakový vektor složen z prvků $d_{\mathbf{t}}(U)\ \forall \mathbf{t} \in \mathcal{S}_N$. Předpokládejme existenci libovolného uspořádání nad množinou $\mathcal{S}_N = (\mathbf{t}_i)_{i=1}^{|}\mathcal{S}_N|$, potom příznakový vektor $\mathbf{d}(U)$ získaný z promluvy u je definován jako:

$$\mathbf{d}(U) = \left[d_{\mathbf{t}_1}(U), d_{\mathbf{t}_2}(U), \ldots, d_{\mathbf{t}_{|\mathcal{S}_N|}}(U)\right] \tag{7.23}$$

kde $d_{\mathbf{t}_i}$ je dáno rovnicí (5.9). Příznakový vektor $\mathbf{d}(U)$ je v modelu HDM výstupem skryté vrstvy – jedná se o transformaci vstupního příznakového prostoru, který v případě užití n-gramové racionální jádrové funkce odpovídá prostoru možných n-gramů nad slovními nebo fonémovými mřížkami, do prostoru jehož dimenze odpovídají sémantickým n-ticím z množiny $\mathcal{S}_N$.

Poznamenejme, že příznakový vektor $\mathbf{d}(U)$ může být rozšířen o další příznaky získané ze vstupní promluvy u, popřípadě z mřížky U. Toto rozšíření můžeme s výhodou použít při modelování podmíněné pravděpodobnosti $\mathrm{P}(\mathsf{C}|\mathsf{E},\mathsf{U})$, která reprezentuje konceptový model z kapitoly 6. Zde poznamenejme, že pravděpodobnost definovaná pomocí rovnice (7.20) vyjadřuje $\mathrm{P}(\mathsf{C}|\mathsf{U})$. Podmínění této pravděpodobnosti náhodnou proměnnou E lze realizovat pomocí vektoru příznaků $\mathbf{d}(E)$ získaného z mřížky sémantických entit E. Tyto příznaky se následně přidají k příznakům na výstupu skryté vrstvy a sestaví se nový příznakový vektor $\mathbf{d}'(U) = [\mathbf{d}(U), \mathbf{d}(E)]$, který se použije při trénování a při dekódování pomocí výstupní vrstvy. Mřížka sémantických entit E je generována z mřížky U a potažmo z promluvy u pomocí algoritmu popsaného v kapitole 8.2. Přínos modelování $\mathrm{P}(\mathsf{C}|\mathsf{E},\mathsf{U})$ namísto $\mathrm{P}(\mathsf{C}|\mathsf{U})$ je pak popsán v kapitole 10.5.

Označme klasifikátor klasifikující příznakový vektor $\mathbf{d}(U)$, popř. rozšířený příznakový vektor $\mathbf{d}'(U)$, do jedné ze tříd z množiny $\mathcal{B}_A$ jako

$$g_A(\mathbf{d}(U), \alpha(A)) : \mathbb{R}^{|\mathcal{S}_N|} \rightarrow \mathcal{B}_A, \quad A \in \Theta \tag{7.24}$$

kde $\alpha(A)$ je vektor parametrů klasifikátoru trénovaného pro koncept A. Zde je nutné poznamenat, že použitá implementace [80, 115] používá strategii one-against-one (kapitola 5.1.5) a tudíž se klasifikátor g_A se skládá z $\frac{|\mathcal{B}_A|\cdot(|\mathcal{B}_A|-1)}{2}$ binárních klasifikátorů a dimenze vektorů parametrů je v tomto případě závislá na početním zastoupení jednotlivých tříd v trénovací množině.

Jako $cls(s_k, A)$ označme funkci, která pro sémantický strom s_k a sémantický koncept A vrátí množinu sémantických konceptů $\beta \in \mathcal{B}_A$, která je tvořena následovníky sémantického konceptu A ve stromu s_k. Je-li sémantický koncept A obsažen v listovém uzlu, je vrácena prázdná množina, není-li A obsažen v s_k, pak vrátí $\{\nu\}$:

$$\mathrm{cls}(s_k, A) = \begin{cases} \beta & \text{pokud } A \to \beta \in s_k \\ \emptyset & \text{pokud } A \text{ je list } s_k \\ \{\nu\} & \text{jinak} \end{cases} \tag{7.25}$$

Z důvodu omezení kladených na tvar sémantických stromů generovaných HDM je nutné speciální ošetření případu 7.3.2 bod 4, tj. případu, kdy trénovací abstraktní sémantická anotace obsahuje více uzlů označených stejným sémantickým konceptem A, ale s různou množinou následovníků β. V takovém případě funkce $cls(s_k, A)$ není definována a v aktuální implementaci HDM jsou tyto příklady z trénovací množiny vypuštěny. Tento jev nastává velice zřídka a neovlivňuje přesnost klasifikace.

Obdobně jako u binárních klasifikátorů sémantických n-tic $f_{\mathbf{t}}$ je vhodné omezit množinu možných cílových tříd pro každý sémantický koncept A. Označíme-li $\mathrm{cnt}(\mathcal{T}, A \to \beta)$ počet výskytů uzlu se sémantickým konceptem A a následníky β napříč trénovací množinou $\mathcal{T}$, pak můžeme definovat modifikovanou funkci $\mathrm{cls}_M(s_k, A)$ následujícím způsobem:

$$\mathrm{cls}_M(s_k, A) = \begin{cases} \beta & \text{pokud } A \to \beta \in s_k \ \wedge \ \mathrm{cnt}(\mathcal{T}, A \to \beta) \geq M \\ \emptyset & \text{pokud } A \text{ je list } s_k \\ \{\nu\} & \text{jinak} \end{cases} \tag{7.26}$$

kde M je volitelný práh. Každý klasifikátor $g_A(\mathbf{d}(U), \alpha(A))$ je pak trénován z trénovací množiny $\mathcal{T}_A$, která je vygenerována z množiny sémanticky anotovaných promluv $\mathcal{T} = (U_k, s_k)_{k=1}^{l}$ jako:

$$\mathcal{T}_A = \{(\mathbf{d}(U_k), \mathrm{cls}_M(s_k, A)) : (U_k, s_k) \in \mathcal{T}, \ k = 1, 2, \ldots, l\} \tag{7.27}$$

Natrénování klasifikátoru je pak úlohou z oblasti strojového učení. V případě SVM použitých v této práci byla na základě doporučení v [73] použita jádrová funkce ve tvaru radiální bázové funkce (rovnice (5.27)). Trénování každého klasifikátoru je řízeno metaparametrem C^A a zároveň jádrová funkce každého z klasifikátorů g_A obsahuje volitelný parametr γ_A. Tyto parametry jsou nastavovány tak, aby byla maximalizována přesnost jednotlivých dílčích klasifikátorů vyhodnocovaná pomocí křížové validace nad trénovacími daty.

7.4 Shrnutí

Výše popisovaný hierarchický diskriminativní model se snaží eliminovat nedostatky modelu STC zmíněné na začátku této kapitoly, především umožňuje přiřazení aposteriorní pravděpodobnosti dané výstupní hypotéze. Této vlastnosti je použito dále při propojení HDM jako konceptového modelu predikujícího abstraktní sémantické stromy a modelu detekce sémantických entit, který detekuje konkrétní sémantické entity obsažené v promluvě spolu s jejich sémantickou interpretací. Navíc, z Bayesovské sítě na obrázku 6.1 vyplývá, že detekce sémantických entit předchází přiřazení abstraktního sémantického stromu a tedy, že informace o sémantických entitách obsažených v promluvě může být použita pro rozšíření příznakového vektoru na výstupu skryté vrstvy (použití $\mathbf{d}'(U)$ namísto $\mathbf{d}(U)$)). Na obrázku 7.9 je uvedeno ilustrativní schema hierarchického diskriminativního modelu a za ním následuje stručná rekapitulace jak trénovacího, tak dekódovacího algoritmu.

Obrázek 7.9: Schéma hierarchického diskriminativního modelu.

7.4.1 Trénování HDM

Vstup:

- Trénovací množina $\mathcal{T} = \{(U_k, s_k)\},\ k = 1 \dots l$
- U_k je vážený konečný automat reprezentující mřížku k-té promluvu
- s_k je k-tý sémantický strom

Výstup:

- Konečné automaty T a R definující racionální jádrovou funkci
- Vektor pro normalizaci jádrových funkcí $\mathbf{k}_\mathcal{T} = [K(U_k, U_k)]_{k=1}^{l}$
- Klasifikátory sémantických n-tic $f_\mathbf{t},\ \mathbf{t} \in \mathcal{S}_N$
- Množina sémantických pravidel $\mathcal{R}$
- Klasifikátory $g_A,\ A \in \Theta$

Algoritmus trénování modelu HDM:

1. Vytvoř vážený konečný transducer $\bar{R}$:

$$\bar{R} = \min\left[\det\left[\text{rmeps}\bigoplus_{j=1}^{l}\Pi_1(T^{-1}\circ U_j)\otimes I(j)\right]\right] \tag{7.28}$$

2. Aplikací postupu z kapitoly 7.1.1 vytvoř z transduceru $\bar{R}$ transducer R.
3. Pro každý prvek U_j trénovací množiny vytvoř transducer L_j.
4. Pomocí kompozice $L_j \circ R$ vypočítej hodnoty jádrových funkcí $K(U_j, U_k),\ k = 1, 2, \dots l$.
5. Sestav normalizační vektor $\mathbf{k}_\mathcal{T} = [K(U_k, U_k)]_{k=1}^{l}$.
6. Proveď normalizaci:

$$\tilde{K}(U_j, U_k) = \frac{K(U_j, U_k)}{\sqrt{K(U_j, U_j)\cdot K(U_k, U_k)}} \tag{7.29}$$

7. Získej množinu sémantických n-tic $\mathcal{S}_N$ parametrizované prahem N.
8. Sestav trénovací množiny $\mathcal{T}_\mathbf{t} = \mathcal{T}_\mathbf{t}^+ \cup \mathcal{T}_\mathbf{t}^-$ pro všechna $\mathbf{t} \in \mathcal{S}_N$.
9. Natrénuj $|\mathcal{S}_N|$ binárních klasifikátorů $f_\mathbf{t}(U, \alpha(\mathbf{t}))$ nad trénovacími množinami $\mathcal{T}_\mathbf{t}$.
10. Pro všechna $s_k,\ k = 1, \dots, l$ vyčísli $\mathbf{d}(U_k) = [d_{\mathbf{t}_i}(U_k)]\,,\ i = 1, 2, \dots, |\mathcal{S}_N|$.
11. Sestav množiny sémantických pravidel $\mathcal{R}$.
12. Pro dané M vytvoř trénovací množiny $\mathcal{T}_A = \{(\mathbf{d}(U_k), \text{cls}_M(s_k, A) : (U_k, s_k) \in \mathcal{T})\}$ pro všechny sémantické koncepty $A \in \Theta$.
13. Pro každý sémantický koncept A natrénuj klasifikátor $g_A(\mathbf{d}(U), \alpha(A))$ klasifikujícího do $|\mathcal{B}_A|$ tříd.

7.4.2 Dekódování pomocí HDM

Vstup:

- Mřížka reprezentující vstupní promluvu U
- Konečné automaty T a R definující racionální jádrovou funkci
- Vektor pro normalizaci jádrové funkce $\mathbf{k}_{\mathcal{T}} = [K(U_k, U_k)]_{k=1}^{l}$
- Klasifikátory sémantických n-tic $f_{\mathbf{t}},\ \mathbf{t} \in \mathcal{S}_N$
- Množina sémantických pravidel $\mathcal{R}$
- Klasifikátory $g_A,\ A \in \Theta$

Výstup:

- Nejpravděpodobnější sémantický strom r_U
- (Volitelně) pravděpodobnostní rozdělení nad hypotézami $P(\mathsf{C} = r_i|\mathsf{U} = U)$

Algoritmus dekódování pomocí modelu HDM:

1. Sestav transduceru L pro promluvu U:

$$L = \det\left[\operatorname{rmeps} \Pi_2(U \circ T)\right] \tag{7.30}$$

2. Proveď kompozici $L \circ R$ a vypočítej hodnotu jádrové funkce:

$$K(U, U_j) = \psi\left(\bigoplus_{x \in \mathcal{A}^*} w\left[(L \circ R)(x, j)\right]\right) \tag{7.31}$$

3. Vypočítej normalizační konstantu $K(U, U)$.

4. Proveď normalizaci pomocí prvků $\mathbf{k}_{\mathcal{T}}$:

$$\tilde{K}(U, U_j) = \frac{K(U, U_j)}{\sqrt{K(U, U) \cdot K(U_j, U_j)}} \tag{7.32}$$

5. Pro každou sémantickou n-tici $\mathbf{t}$ pomocí klasifikátoru $f_{\mathbf{t}}$ vyčísli vzdáleností k rozhodovací nadrovině $d_{\mathbf{t}}(U)\ \forall t \in \mathcal{S}_N$ a sestav vektor $\mathbf{d}(U) = [d_{\mathbf{t}_i}(U)]_{i=1}^{|\mathcal{S}_N|}$.

6. Predikuj pravděpodobností $\mathrm{P}(A \to \beta \in \mathcal{B}_A|U)$ s využitím klasifikátorů g_A.

7. Vytvoř množiny $\mathcal{R}_U$ z množiny obecných sémantických pravidel $\mathcal{R}$ doplněním pravděpodobností expanze $\mathrm{P}(A \to \beta \in \mathcal{B}_A|U)$. Sestav sémantickou gramatiku promluvy $G_U = (\Theta, \mathcal{R}_U, S)$.

8. Získej nejpravděpodobnější sémantický strom r_U z gramatiky G_U pomocí algoritmu z kapitoly 7.3.1.

9. Volitelně vygeneruj n-nejlepších sémantických stromů $r_1, r_2, \ldots, r_n$ s přiřazenými pravděpodobnostmi $\mathrm{P}(\mathsf{C} = r_i|\mathsf{U} = U)$ podle rovnice (7.22).

Kapitola 8

Detekce sémantických entit

V této kapitole se zaměříme na popis algoritmu vyvinutého pro hierarchický diskriminativní model, který umožňuje získání pravděpodobnostního rozdělení $\mathrm{P}(\mathsf{E} = e|U = u)$ z promluvy u, popř. z mřížky U. Poznamenejme, že popisovaný algoritmus předpokládá použití slovní mřížky, popř. první nejlepší hypotézy. Problematika detekce sémantických entit z fonémových mřížek je nad rámec této práce, nicméně velice blízce souvisí s problematikou detekce klíčových slov ve fonémových mřížkách [69, 116].

Nejprve definujme několik pojmů týkajících se sémantických entit. *Sémantickou entitou* myslíme konkrétní objekt zmíněný v dané promluvě a významný z pohledu sémantické analýzy. Sémantické entity mohou být různých *typů*, například časové údaje, datum, položky z rozsáhlých databází (seznamy stanic, osob). Sémantické entity mohou mít svoji vnitřní strukturu – *sémantickou interpretaci*. Pro příklad jmenujme sémantické entity typu čas, které se mohou skládat z údaje o hodinách a minutách.

Název *sémantická entita* nebyl zvolen náhodně. Cílem je evokovat podobnost se pojmenovanými entitami a detekcí pojmenovaných entit, která probíhá nad textovými daty a jejím cílem je označit odpovídající části textu typem pojmenované entity, např. [88, 117, 118]. Detekce sémantických entit má obdobný cíl, označit určitou část vstupní informace jako příslušející sémantické entitě daného typu, nicméně zásadní kvalitativní rozdíl je v reprezentaci vstupní informace – nejedná se o text, ale o slovní mřížku ze systému automatického rozpoznávání řeči definující pravděpodobnostní rozdělení nad množinou možných hypotéz a reprezentovanou pomocí váženého konečného automatu. Navíc detekce sémantických entit neprovádí pouze označení sémantické entity, ale i analýzu její vnitřní struktury – přiřazení sémantické interpretace závislé na typu sémantické entity.

Idea detekce sémantických entit není nová, například v práci [119] autoři Béchet a kol. prezentují metodu pro detekci pojmenovaných entit v mluvené řeči. Využívají hierarchickou architekturu, kde na základě první nejlepší hypotézy z mřížky je nejprve rozhodnuto, v jakých segmentech mřížky se entity nacházejí a následně je tato část mřížky zpracována v parseru pro získání typu sémantické entity a pomocí vážených konečných transducerů pro získání interpretace entity. V práci [120] autoři používají přímo algoritmy pro práci s váženými konečnými transducery pro získání sémantických entit, nicméně navíc používají statistický výplňový model (filler model) modelující segmenty promluvy, které nenáleží žádné sémantické entitě. V práci [121] pak byl použit 3-gramový konceptový tagger predikující ke každému slovu w_t odpovídající sémantickou značku c_t. Jako implementace tohoto přístupu jsou opět použity vážené konečné automaty.

V práci [122] jsou pro detekci sémantických entit použity namísto slovních mřížek slovní grafy vyjadřující alternativní slovní hypotézy pro slova v určitých časech (angl. word confusion networks, WCN). Autoři používají WCN namísto slovních mřížek z důvodu efektivnější implementace algoritmů pro detekci sémantických entit v porovnání s detekcí ze slovních mřížek. Dosahované přesnosti detekce sémantických entit z WCN a ze slovních mřížek jsou však v této práci prakticky totožné.

Obecně můžeme mluvit o dvou různých přístupech k modelování sémantických entit:

- *Statistický přístup* – v tomto přístupu jsou pravidla pro přiřazení sémantických entit dané posloupnosti slov trénována statistickým přístupem z trénovacích dat. Tento přístup má však svá omezení – snadno lze nahlédnout, že v úloze poskytování informací o odjezdech a příjezdech vlaků se počet různých stanic a zastávek v České republice pohybuje řádově v tisících a toto číslo je srovnatelné s celkovým počtem trénovacích příkladů. Nicméně pravděpodobnostní rozdělení počtu výskytů jednotlivých vlakových stanic je silně nerovnoměrné a většina vlakových stanic a zastávek se tak v trénovacích datech ani nebude vyskytovat.
- *Znalostní přístup* – zde jsou využity pro přiřazení sémantických entit použity expertní znalosti, zpravidla návrháře hlasového dialogového systému, případně získané z databáze řešeného problému. Tyto expertní znalosti umožňují vhodným způsobem pokrýt neviděná data a zvýšit robustnost detekce sémantických entit. Znalosti mohou být reprezentovány mnoha způsoby. Zpravidla se jedná o výčet možných hodnot nebo o jejich popis vhodnou gramatikou. Použití znalostního přístupu pro detekci sémantických entit vhodně doplňuje statistické metody používané v hierarchickém diskriminativním modelu a zvyšuje robustnost výsledného systému. Je důležité zdůraznit, že expertní znalosti mohou být často automaticky generovány z vhodné databáze (seznam stanic, seznam osob) nebo dokonce mohou být relativně jednoduše přenositelné mezi různými doménami (gramatiku sémantických entit typu čas lze použít bez úprav ve více různých úlohách).

Soustřeďme se pouze na přístup založený na znalostech. Důvodem je především snaha pomocí expertních znalostí posílit statisticky založený model porozumění a zvýšit tak jeho robustnost a přesnost. Budeme se však velice specificky zajímat o detekci sémantických entit ze vstupních slovních mřížek, neboť právě takový typ vstupních dat je použit v hierarchickém diskriminativním modelu popsaném v kapitole 7. Přístup popsaný v této kapitole lze chápat rovněž jako zobecnění metody náhrady posloupnosti slov za identifikátory lexikálních tříd používané v modelu STC – zde je doplněn o možnost reprezentovat jazyk sémantických entit pomocí bezkontextových gramatik a navíc lze pomocí zde prezentovaného přístupu generovat ze vstupní slovní mřížky výstupní mřížku obsahující sémantické entity.

Předpokládejme, že sémantické entity daného typu mají svoji vnitřní strukturu popsatelnou bezkontextovou gramatikou. Přestože lze uvažovat i stochastické bezkontextové gramatiky, je v praxi velmi obtížné expertním způsobem určit pravděpodobnosti expanzí jednotlivých pravidel. Bezkontextové gramatiky jsou intuitivní, standardizovaný [51] způsob zápisu znalostí návrháře hlasového dialogového systému, navíc jejich užití umožňuje znovupoužít existující bázi znalostí z existujících hlasových dialogových systémů, kde rozpoznávání nebo porozumění řeči je založeno na těchto gramatikách.

Ve vstupní promluvě je nutné označit (a tudíž gramatikami modelovat) pouze takové podposloupnosti terminálních symbolů, které odpovídají nějaké gramatice. Tím se řádově zjednodušuje úloha návrhu gramatiky reprezentující sémantické entity daného typu, neboť není zpravidla nutné uvažovat různá výplňová slova, která nenáleží žádné sémantické entitě. Při použití jediné globální stochastické bezkontextové gramatiky by bylo nutné výplňová slova modelovat a zahrnout je mezi terminální symboly gramatiky.

Poněvadž jsou mřížky na výstupu systému automatického rozpoznávání řeči reprezentovány pomocí vážených konečných automatů, je vhodné realizovat detekci sémantických entit rovněž v rámci těchto struktur. Proto je nutné nejprve provést kompilaci bezkontextových gramatik do podoby váženého konečného transduceru. Poznamenejme, že vážené konečné automaty umožňují exaktní reprezentaci pouze (stochastických) regulárních gramatik a jazyků [3]. Nicméně v oblasti rozpoznávání a porozumění řeči jsou gramatiky reprezentovány jako bezkontextové, především z důvodu výrazně vyšší čitelnosti výsledné gramatiky [123].

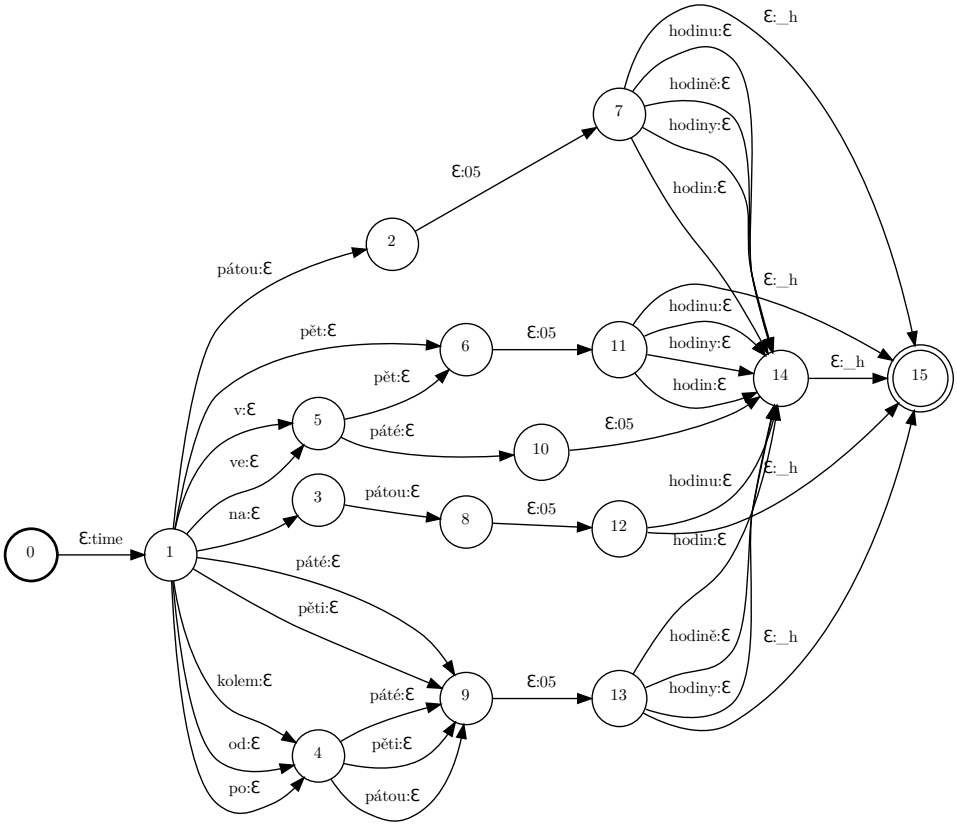
Předpokládejme, že bezkontextová gramatika není rekurzivní, tj. pomocí pravidel gramatiky nelze z neterminálního symbolu A odvodit derivační podstrom obsahující A v jiném uzlu než v kořeni. Potom lze tuto bezkontextovou gramatiku převést na regulární [124]. V případě, že bezkontextová gramatika je rekurzivní, je možné v průběhu kompilace hlídat hloubku rekurze a od určité hloubky znoření aplikovat omezení, např. dále neexpandovat neterminál způsobující rekurzi a nahradit jej prázdným symbolem. Výsledný vážený konečný automat však poté bude přijímat pouze aproximaci původní bezkontextové gramatiky [124, 125]. V oblasti zpracování mluvené řeči však tato omezení nejsou limitující, neboť možné promluvy uživatele hlasového dialogového systému jsou svojí délkou v čase a tudíž i v počtu slov omezené a aproximace bezkontextové gramatiky konečným automatem je pro tyto omezené posloupnosti slov dostačující. Problematice kompilace bezkontextových gramatik na vážené konečné automaty se věnují například publikace [123, 126].

Výstupem kompilace bezkontextové gramatiky odpovídající sémantickým entitám určitého typu je konečný transducer převádějící posloupnost slov na posloupnost značek reprezentujících sémantickou entitu, tj. typ a interpretaci. Budeme předpokládat, že první symbol posloupnosti značek identifikuje typ sémantické entity, zbylé symboly jsou závislé na typu sémantické entity a reprezentují interpretaci. Pro přehlednost zápisu budou v příkladech jednotlivé symboly sémantických entit odděleny znakem dvojtečky. Příklad takové gramatiky je zobrazen na obrázku 8.1. Sémantické gramatiky mohou záměrně obsahovat i množství negramatických cest, např. čas *ve pět hodin* ve výše uvedeném příkladě. Tímto přístupem lze účinně podchytit např. chybně rozpoznaná slova na výstupu systému automatického rozpoznávání řeči a zvýšit tak robustnost detekce sémantických entit.

Nyní předpokládejme, že každé sémantické entitě z odpovídá gramatika G_z a z ní kompilací získaný konečný transducer T_z. Gramatika G_z je konstruována tak, aby transducer T_z při akceptaci posloupnosti vstupních symbolů jako první symbol vrátil identifikátor z. Pak lze všechny sémantické entity reprezentovat pomocí transduceru Z získaného jako sjednocení dílčích T_z:

$$Z = \bigoplus_z T_z \tag{8.1}$$

Transducer Z nemusí nutně být funkcionální, tj. pro jeden akceptovaný vstupní řetězec může vrátit obecně více výstupních řetězců. Proto tento transducer nelze přímo optimalizovat pomocí algoritmů determinizace a minimalizace. Nicméně lze použít postup po-



Obrázek 8.1: Konečný automat získaný kompilací gramatiky sémantické entity typu *čas*. Pro názornost byla vybrána pouze ta část automatu, která odpovídá výstupním symbolům *time:05:_h* (pět hodin).

psaný např. v [69] – zde jsou přeznačeny vstupní a výstupní symboly přechodů tak, že nově nesou jediný symbol vzniklý zakódováním původní dvojice vstupní-výstupní symbol. Takto vznikne (vážený) konečný akceptor, na který již lze aplikovat algoritmus determinizace a minimalizace popsaný v kapitole 5.2.3. Po optimalizaci zakódovaného automatu je nutné převést zakódované symboly na přechodech zpět do původní abecedy vstupních a výstupních symbolů. Tímto krokem se může porušit vlastnost determinismu a minimality výsledného automatu, nicméně míra nedeterminismu vyjádřená počtem přechodů z daného stavu označených stejným symbolem je menší nebo rovná původnímu automatu před optimalizací.

Nyní uvažujme vstupní promluvu u a odpovídající slovní mřížku U. Cílem je nalézt pravděpodobnostní rozložení $\mathrm{P}(\mathsf{E} = e|\mathsf{U} = U)$ posloupností sémantických entit $e = (e_1, e_2, \ldots, e_n)$, přičemž každé posloupnosti e odpovídá posloupnost typů sémantických entit $(z_1, z_2, \ldots, z_n)$ a pro každou sémantickou entitu e_i platí, že kompozice $T_{z_i} \circ e_i = T$ je neprázdný automat. Jinými slovy, pro každou sémantickou entitu e_i existuje cesta automatem T_{z_i} taková, že posloupnost výstupních symbolů odpovídá e_i. Automat U se předpokládá v takové podobě, kdy jeho váhové ohodnocení tvoří pravděpodobnostní distribuci nad množinou cest přijímaných tímto automatem.

Pro řešení úlohy detekce sémantických entit použijeme přístup faktorového transduceru, který umožňuje efektivně reprezentovat všechny možné faktory vstupní mřížky U a jejich

aposteriorní pravděpodobnosti. Navíc díky tomuto přístupu není nutné modelovat výplňová slova mezi jednotlivými sémantickými entitami, neboť ty faktory, která tato výplňová slova obsahují neodpovídají žádné cestě transducerem Z a tudíž neovlivní ostatní detekované sémantické entity. Nicméně, jednotlivé faktory různých délek v rámci faktorového automatu $F(U)$ se překrývají a tudíž jsou generovány nadbytečné výskyty sémantických entit. Proto je pro nalezení množiny jednoznačně přiřazených sémantických entit použito celočíselného programování (kapitola 8.1) a následně je popsán algoritmus, který umožňuje z množiny jednoznačně přiřazených sémantických entit rekonstruovat mřížku sémantických entit (kapitola 8.2).

8.1 Nalezení jednoznačně přiřazených sémantických entit

Před samotným popisem procesu hledání sémantických entit pomocí vážených konečných transducerů uveďme výčet případů, které mohou pro danou cestu mřížkou U nastat:

1. Jedné cestě v mřížce U odpovídá prázdná posloupnost sémantických entit. Váhu této cesty je však nutné promítnout do pravděpodobnosti $\mathrm{P}(\mathsf{E} = \{\}|\mathsf{U} = U)$.

2. Jedné cestě v mřížce U odpovídá právě jedna posloupnost sémantických entit.

3. Jedné cestě v mřížce U může odpovídat více různých posloupností sémantických entit.

   Příklad: Promluva *chci jet v jedenáct dvacet pět* může vést na následující sémantické entity: *time:11:_h*, *time:11:_h:20 _m* a *time:11:_h:20:05:_m* reprezentující časové údaje 11:00, 11:20, 11:25. Entity *time:11:_h*, *time:11:_h:20 _m* jsou nadbytečné, odpovídají faktorům *v jedenáct* a *v jedenáct dvacet*. Očekávaná sémantická entita pro tento příklad je *time:11:_h:20:05:_m* a časový údaj 11:25.

4. Více různých cest v mřížce U může mít přiřazeno stejnou posloupnost sémantických entit.

   Příklad: hypotézy *kolem páté hodiny* a *kolem pátý hodiny* vedou na stejnou sémantickou entitu *time:05:_h*

Na proces nalezení posloupnosti sémantických entit lze aplikovat obdobný postup jako při indexaci a vyhledávání informací v archívech mluvené řeči [68, 69] stručně popsaný v kapitole 3.6. Přitom je však nutné brát v úvahu zmíněné případy a stanovit způsob, jakým je řešit. V případě 3, kdy jedna cesta vede na více interpretací, se nabízí použít pro reprezentaci stochastické bezkontextové gramatiky vedoucí na vážené konečné transducery. Nicméně jak expertní, tak i statistický odhad pravděpodobností expanzí jednotlivých pravidel je netriviální úlohou. Proto byly v této práci použity prosté bezkontextové gramatiky a následující heuristika *jednoznačného maximálního přiřazení*:

- Každé slovo dané cesty π_U mřížkou U může náležet pouze jedné sémantické entitě.
- A zároveň, z možných posloupností sémantických entit pro danou cestu π_U je vybrána taková, která maximalizuje počet slov dané cesty, která jsou součástí některé sémantické entity.

V dalším textu budeme používat logaritmický a pravděpodobnostní polookruh. Tyto polookruhy jsou izomorfní, pro váhu přechodu v logaritmické polookruhu $w_{\log}$ a pravděpodobnostním polookruhu w_{pr} platí:

$$\begin{aligned} w_{\log} &= -\log(w_{\mathrm{pr}}) \\ w_{\mathrm{pr}} &= \exp(-w_{\log}) \end{aligned} \tag{8.2}$$

Platí-li pro daný akceptor A nad pravděpodobnostním nebo logaritmickým polookruhem, že $\bigoplus_{\pi \in \mathcal{P}(I_A, \mathcal{F}_A)} = \bar{1}$, pak akceptor definuje pravděpodobnostní rozdělení nad množinou řetězců přijímaných tímto akceptorem a pravděpodobnost cesty je dána její vahou:

$$\begin{aligned} \mathrm{P}(\pi \in A_{\mathrm{pr}}) &= \mathrm{P}(\pi \in \mathcal{P}(I_A, \mathcal{F}_A)) = w_{\mathrm{pr}}[\pi] \\ \mathrm{P}(\pi \in A_{\log}) &= \mathrm{P}(\pi \in \mathcal{P}(I_A, \mathcal{F}_A)) = \exp(-w_{\log}[\pi]) \end{aligned} \tag{8.3}$$

Uveďme nyní algoritmus hledání pravděpodobnostního rozdělení $\mathrm{P}(\mathsf{E} = e|\mathsf{U} = u)$ posloupností sémantických entit pro danou mřížku:

Vstup:

- Mřížka U nad logaritmickým polookruhem
- Transducer reprezentující expertní znalost Z

Výstup:

- Množina nepřekrývajících se sémantických entit P_Z^*
- Mřížka sémantických entit E

Algoritmus pro nalezení pravděpodobnostního rozdělení $\mathrm{P}(\mathsf{E} = e|\mathsf{U} = U)$**:**

1. Mřížka (vážený akceptor) U je převedena na vážený transducer U_T tak, že vstupní symboly přechodů jsou nahrazeny v rámci mřížky jednoznačnými identifikátory, výstupní symboly jsou zachovány.

2. Vážený transducer U_T je převeden na vážený faktorový automat $F(U_T)$ pomocí postupu popsaného v odstavci 5.2.2.

3. Kompozicí $F(U_T) \circ Z$ jsou vybrány ty faktory, které odpovídají nějaké sémantické entitě ze Z.

4. Je aplikována heuristika *jednoznačného přiřazení*. Pro její aplikaci jsou použity jednoznačné identifikátory přechodů mezi stavy zavedené v bodu 1. Výsledkem je množina nepřekrývajících se sémantických entit P_Z^*.

5. Z množiny sémantických entit je rekonstruován vážený konečný akceptor přijímající všechny posloupnosti sémantických entit e odpovídajících mřížce U.

6. Nad výsledným akceptorem je provedeno odstranění ϵ-přechodů, determinizace a minimalizace. Akceptor E získaný optimalizací odpovídá mřížce sémantických entit a jeho ohodnocení vahami odpovídá pravděpodobnostní distribuci $\mathrm{P}(\mathsf{E} = e|\mathsf{U} = U)$.

Po provedení kompozice $F_Z = F(U_T) \circ Z$ výsledný transducer reprezentuje množinu faktorů cest transducerem U_T označených sémantickými entitami. Vstupní symboly libovolné cesty $\pi(F_Z)$ reprezentují část cesty (faktor) v původní mřížce U_T, výstupní symboly pak již samotné sémantické entity definované transducerem Z.

Pro každý faktor $\pi^i(F_Z) = (u^i, y^i)$ sestavme pětici:

$$(u^i, y^i, p[u^i], n[u^i], \mathrm{P}(u^i \in U))$$

kde:

- u^i je faktor $\Pi_1(U_T)$ – posloupnost unikátních identifikátorů přechodů o délce k^i symbolů,
- y^i je posloupnost značek, jako celek jednoznačně definuje typ sémantické entity a její hodnotu,
- $p[u^i]$ je počáteční stav přechodu s identifikátorem u^i_1 v automatu U_T,
- $n[u^i]$ je koncový stav přechodu s identifikátorem $u^i_{k^i}$ v automatu U_T,
- $\mathrm{P}(u^i \in U_T) = \exp(-w_{F_Z}[u^i, y^i])$ je aposteriorní pravděpodobnost výskytu faktoru (u^i, y^i) v mřížce U.

Předpokládejme, že těchto cest (faktorů) je celkem n a definujme jejich libovolné uspořádání s indexem $i = 1, 2, \ldots, n$ do posloupnosti P_Z:

$$P_Z = \{(u^i, y^i, p[u^i], n[u^i], \mathrm{P}(u^i \in U_T))\}_{i=1}^{n} \tag{8.4}$$

Z posloupnosti P_Z je nyní nutné vybrat takovou podposloupnost P_Z^*, která splňuje požadavky heuristiky jednoznačného maximálního přiřazení. Pro rozhodnutí o tom, zda konkrétní dvojice (u^i, y^i) odpovídá heuristice, formulujme optimalizační úlohu binárního celočíselného programování. Její omezení zajišťují splnění prvního bodu heuristiky (pro každou cestu mřížkou je libovolné slovo přiřazeno nejvýše jedné sémantické entitě), optimalizační kriterium pak splnění druhého bodu (z možných řešení je vybráno takové, které pokrývá maximální počet slov dané cesty). Úlohu formulujme následovně:

$$\begin{aligned} &\mathbf{G} \cdot \mathbf{x} \leq \mathbf{h} \\ &\mathbf{c}^\intercal \cdot \mathbf{x} \to \max \end{aligned} \tag{8.5}$$

kde optimalizace probíhá vzhledem k prvkům n-rozměrného vektoru $\mathbf{x} = [x_i]_{i=1}^n$, kde prvky $x_i \in \{0, 1\}$. Pro optimální řešení platí, že je-li $x_i = 1$, pak (u^i, y^i) splňuje heuristiku jednoznačného maximálního pokrytí a y^i patří do některé posloupnosti sémantických entit e přiřazených mřížce U.

Pro definici matice $\mathbf{G}$ nejprve zkonkretizujme požadavek prvního bodu heuristiky jednoznačného maximálního pokrytí. Poněvadž nad mřížkou U_T může obecně existovat velké množství cest, je nutné pro každý pár (u^i, u^j), $i \neq j$ stanovit, zda v automatu U_T neexistuje cesta π_{U_T} taková, že faktory u^i a u^j se překrývají. Budeme říkat, že faktory u^i a u^j $(i \neq j)$ *se překrývají*, pokud je splněna alespoň jedna z následujících podmínek:

- Existuje neprázdná posloupnost u' a posloupnosti a, b takové, že $u^i = au'$ a zároveň $u^j = u'b$.
- Existuje neprázdná posloupnost u' a posloupnosti a, b takové, že $u^i = u'a$ a zároveň $u^j = bu'$.

- Existují posloupnosti a, b takové, že $u^i = au^jb$.
- Existují posloupnosti a, b takové, že $u^j = au^ib$.

Podotkněme, že tato definice je symetrická, pokud se u^i překrývá s u^j, pak se i u^j překrývá s u^i. Pokud se faktory u^i a u^j překrývají, pak pro splnění heuristiky jednoznačného maximálního pokrytí je nutné, aby ve výsledné posloupnosti P_Z^* byl nejvýše jeden z nich a tedy:

$$x_i + x_j \leq 1 \tag{8.6}$$

Předpokládejme, že v posloupnosti P_Z existuje m párů (u^i, u^j), které se překrývají. Platí $0 \leq m \leq \frac{n^2-n}{2}$. Pro případ, kdy neexistují žádné překrývající se faktory ($m = 0$), definujme $x_i = 1,\ i = 1, 2, \ldots, n$.

Pro případy $m > 0$ sestavíme matici $\mathbf{G} = [g_{kl}]$ o $m \times n$ prvcích, pro kterou platí, že pokud se u^i a u^j překrývají, pak existuje řádek matice k takový, že $g_{ki} = g_{kj} = 1$ a $g_{kl} = 0$ pro $l \neq i, j$. Vektor $\mathbf{h}$ je m-rozměrný sloupcový vektor samých jedniček.

Kriteriální funkce má za cíl vybrat ze všech možných řešení takové, které maximalizuje počet slov v mřížce s přiřazenou sémantickou entitou. Kritérium je dáno n-rozměrným sloupcovým vektorem $\mathbf{c}$ s prvky c_i, pro jejichž výpočet byl použit následující vzorec:

$$c_i = (k^i)^2 \cdot \mathrm{P}(u^i \in U) \tag{8.7}$$

Tento tvar kriteria má za cíl preferovat ta řešení, která zahrnují delší faktory u^i (k^i je délka faktoru u^i). Vážení aposteriorní pravděpodobností pak zajistí, že jsou do posloupnosti P_Z^* prioritně vybírány ty faktory, které mají vyšší aposteriorní pravděpodobnost. Druhá mocnina délky faktoru pak má za cíl při optimalizaci prioritizovat ta řešení, která mají menší počet faktorů. Zabrání se tím rozdělení sémantických entit na jednotlivé části – například oddělení hodin a minut u časového údaje do samostatných sémantických entit.

Příklad: Pokud $u^1 = \{1\}$, $u^2 = \{2\}$ a $u^3 = \{1, 2\}$ a $\mathrm{P}(u^i \in U) = 1,\ i = 1, 2, 3$. Pak by bez použití druhé mocniny v kriteriu existovala dvě ekvivalentní řešení $\mathbf{x}_1 = [1, 1, 0]^\intercal$ a $\mathbf{x}_2 = [0, 0, 1]^\intercal$ s hodnotou kriteriální funkce 2. Zavedením druhé mocniny získáme jediné řešení $\mathbf{x} = [0, 0, 1]^\intercal$ s hodnotou kriteriální funkce 4. □

Po aplikaci algoritmu binárního celočíselného programování a získání optimálního vektoru $\mathbf{x}$ již lze provést omezení posloupnosti P_Z na podposloupnost P_Z^* splňující heuristiku jednoznačného maximálního pokrytí:

$$P_Z^* = \{(u^i, y^i, p[u^i], n[u^i], \mathrm{P}(u^i \in U) \in P_Z) : x_i = 1\} \tag{8.8}$$

8.2 Sestavení mřížky sémantických entit

Cílem detekce sémantických entit však není nalezení množiny faktorů splňujících výše uvedenou heuristiku, nýbrž získání pravděpodobnostního rozdělení $\mathrm{P}(\mathsf{E} = e|\mathsf{U} = U)$, přičemž e je posloupnost sémantických entit tvořená prvky y^i z P_Z^*.

V rámci této práce byl vyvinut exaktní algoritmus, který pro dané faktory z P_Z^* sestaví minimální deterministický acyklický vážený konečný akceptor E, jehož cesty odpovídají

různým posloupnostem e a váhy pak pravděpodobnosti $\mathrm{P}(\mathsf{E} = e)$. Pokud byly faktory P_Z^* generovány z mřížky U s použitím transduceru Z, pak E reprezentuje mřížku sémantických entit a přímo modeluje podmíněnou pravděpodobnost $\mathrm{P}(\mathsf{E} = e|\mathsf{U} = U)$.

Pro sestavení takové mřížky použijeme hodnoty $\alpha[q]$ odpovídající nejkratší vzdálenosti z počátečních stavů E do stavu q a $\beta[q]$ nejkratší vzdálenosti ze stavu q do koncových stavů E. Pro přehlednost znovu uveďme jejich rekurzivní výpočet popsaný na straně 35 v rovnicích (5.63) a (5.64):

$$\alpha[q] = \bigoplus_{e \in \mathcal{E}: n[e]=q} \alpha[p[e]] \otimes w[e] \tag{8.9}$$

$$\beta[q] = \bigoplus_{e \in \mathcal{E}: p[e]=q} w[e] \otimes \beta[n[e]] \tag{8.10}$$

Uvažujme nyní, že automat E je definován nad pravděpodobnostním polookruhem, pak $\alpha[q]$ odpovídá dopředné pravděpodobnosti dosažení uzlu q z množiny počátečních stavů, resp. $\beta[q]$ zpětné pravděpodobnosti dosažení koncových stavů automatu z uzlu q. Aposteriorní pravděpodobnost výskytu faktoru se vstupními symboly u a výstupními symboly y v automatu E nad pravděpodobnostním polookruhem lze pak vyjádřit jako:

$$\mathrm{P}\left((u, y) \in E\right) = \sum_{\substack{\pi \\ i[\pi]=u \\ o[\pi]=y}} \alpha\left[p[\pi]\right] \cdot w[\pi] \cdot \beta\left[n[\pi]\right] \tag{8.11}$$

S využitím výše popsaných pravděpodobností již můžeme popsat algoritmus, který na základě posloupnosti P_Z^* vytvoří akceptor E takový, že:

$$\mathrm{P}(y \in E) = \mathrm{P}(\mathsf{E} = e|\mathsf{U} = U) = \mathrm{P}\left((u, y) \in F_Z\right) \tag{8.12}$$

tj. aposteriorní pravděpodobnost výskytu symbolů y v mřížce E je rovna aposteriorní pravděpodobnosti výskytu dvojice (posloupnost slov u, sémantická entita y) v transduceru $F_Z = F(U_T) \circ Z$ a potažmo i v mřížce U_T a tedy i U, neboť Z je transducer bez vah a U_T je vzájemně jednoznačným způsobem převoditelné na U (a naopak).

Vstup:

- Posloupnost $P_Z^* = \{(u^i, y^i, p[u^i], n[u^i], \mathrm{P}(u^i \in U))\}_{i=1}^{n^*}$
- Časy $t[p[u^i]]$ a $t[n[u^i]]$ přiřazené stavům $p[u^i]$ a $n[u^i]$

Výstup:

- Akceptor E

Algoritmus pro sestavení mřížky sémantických entit:

1. Vytvoř prázdný akceptor E nad pravděpodobnostním polookruhem.
2. Vytvoř prázdnou posloupnost paralelních stavů S.
3. Vytvoř přechody označené sémantickými entitami, pro $i = 1, 2, \ldots n^*$:
   - Vytvoř nový přechod e^i: $i[e^i] = y^i$, $w[e^i] = 1$.
   - Označ počáteční stav jako $p^i = p[e^i]$, koncový stav jako $n^i = n[e^i]$.
   - Nastav $\alpha[p^i] = \alpha[n^i] = \mathrm{P}(u^i \in U)$.
4. Vytvoř červené a modré stavy, pro $i = 1, 2, \ldots n^*$:
   - Pro stav p^i vytvoř paralelní červený stav R^i.
   - Nastav $t[R^i] = t[p[u^i]]$.
   - Pro stav n^i vytvoř paralelní modrý stav B^i.
   - Nastav $t[B^i] = t[n[u^i]]$.
   - Přidej R^i a B^i do S.
5. Uspořádej S vzhledem k relaci $\prec$ (seřazení podle času).
6. Označ S_1 jako počáteční stav E, S_{2n^*} jako koncový stav s váhou 1.
7. Nastav dopřednou pravděpodobnost počátečního stavu $\alpha[S_1] = 1$.
8. Iteruj přes paralelní stavy, $j = 1, 2, \ldots 2n^*$:
   - Je-li S_j modrý stav:
     - Najdi stav n^i odpovídající modrému stavu S_j.
     - Vytvoř nový přechod a: $i[a] = \epsilon$, $p[a] = n^i$, $n[a] = S_j$, $w[a] = 1$.
     - Vytvoř nový přechod b: $i[b] = \epsilon$, $p[b] = S_j$, $n[b] = S_{j+1}$, $w[b] = 1$.
     - Nastav: $\alpha[S_j] = \alpha[S_{j-1}] \cdot w[\pi(S_{j-1}, S_j)] + \alpha[n^i]$.
   - Je-li S_j červený stav:
     - Je-li $S_j \neq S_1$ nastav: $\alpha[S_j] = \alpha[S_{j-1}] \cdot w[\pi(S_{j-1}, S_j)]$.
     - Najdi stav p^i odpovídající modrému stavu S_j.
     - Vytvoř nový přechod a: $i[a] = \epsilon$, $p[a] = S_j$, $n[a] = p^i$, $w[a] = \frac{\alpha[p^i]}{\alpha[S_j]}$.
     - Vytvoř nový přechod b: $i[b] = \epsilon$, $p[b] = S_j$, $n[b] = S_{j+1}$, $w[b] = 1 - w[a]$.
9. Proveď optimalizaci $E \leftarrow \min\left[\det\left[\mathrm{rmeps}\left[E\right]\right]\right]$.

K popsanému algoritmu uveďme několik poznámek:

- Algoritmus provádí přímo projekci dvojice (u^i, y^i) na výstupní symboly, zároveň celá posloupnost y^i je považována za jediný symbol. Posloupnosti u^i slouží pouze k rozlišení různých faktorů z P_Z^* a k přiřazení časové informace k přechodům v neoptimalizovaném akceptoru E.
- Výsledný akceptor E je acyklický akceptor.
- Relaci pro uspořádání paralelních stavů definujme tak, že $S_i \prec S_j$ právě tehdy, když:
  - $t[S_i] < t[S_j]$, nebo
  - $t[S_i] = t[S_j]$ a S_i je modrý stav a S_j je červený stav
- Suma vah přechodů z každého stavu je 1. Důkaz je triviální – pro stavy p^i, n^i a B^i existuje právě jeden přechod s vahou 1, pro stavy R^i pak existují dva přechody a, b, ale platí, že $w[b] = 1 - w[a]$ a tudíž i suma vah přechodů ze stavu R^i je 1.
- Pro zpětné pravděpodobnosti $\beta[q]$ v akceptoru E platí, že $\beta[q] = 1$.
  Důkaz: Uvažujme v S poslední červený stav R^i. Tento červený stav je následován minimálně jedním modrým stavem, neboť $R^i \prec B^i$. Poněvadž $\beta[S_{2n^*}] = 1$ a modré stavy opouští právě jeden přechod vahou 1, pak pro všechny modré stavy B^k takové, že $R^i \prec B^k$ platí, že $\beta[B^k] = 1$. Poněvadž stav n^k opouští právě jeden přechod s vahou 1, pak i $\beta[n^k] = 1$. A obdobně i stav p^k opouští právě jeden přechod s vahou 1 a tedy i $\beta[p^k] = 1$.

  Vyčísleme nyní zpětnou pravděpodobnost v červeném stavu R^i:

  $$\beta[R^i] = w[\pi(R^i, p^i)] \cdot \beta[p^i] + w[\pi(R^i, S_{j+1})] \cdot \beta[S_{j+1}] \tag{8.13}$$

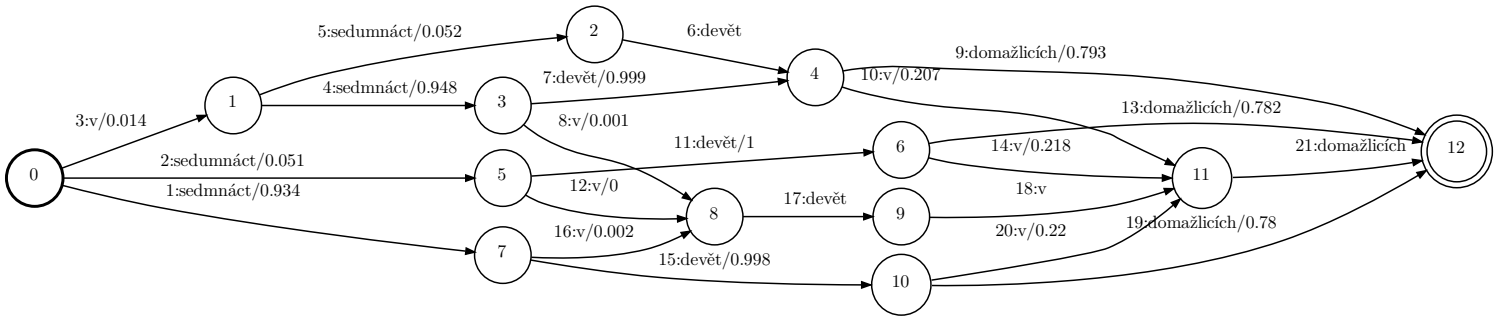
  kde S_{j+1} je stav následující v S za červeným stavem R^i. Poněvadž $\beta[S_k] = 1$ pro všechna $R^i \prec S_k$, pak musí na základě výše uvedené úvahy být i $\beta[p^i] = 1$. Dostáváme:

  $$\beta[R^i] = w[\pi(R^i, p^i)] + w[\pi(R^i, S_{j+1})] \tag{8.14}$$

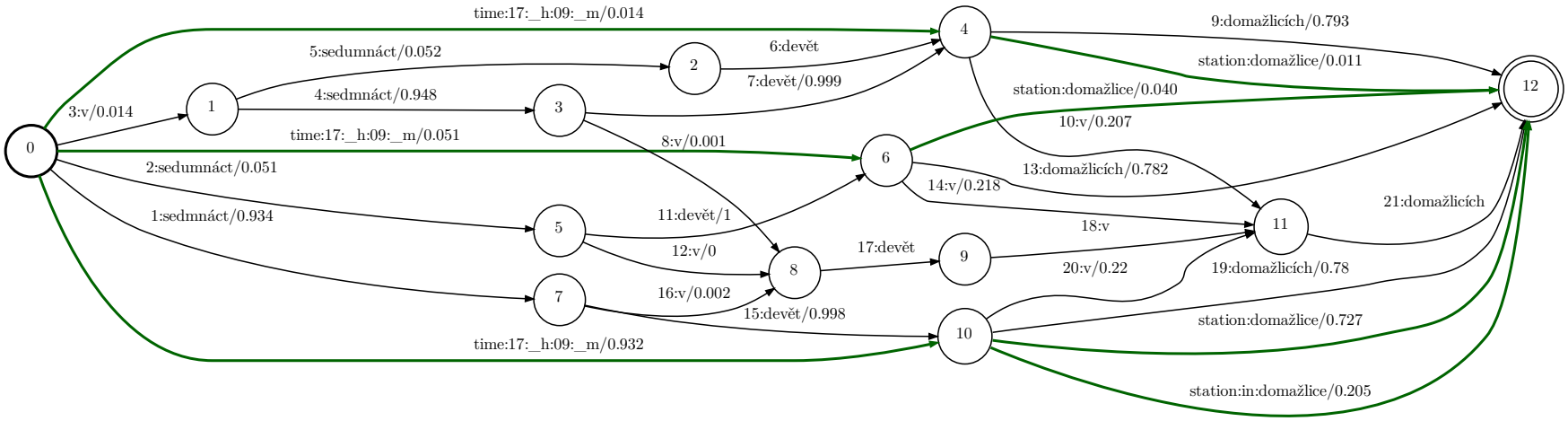
  Poněvadž suma přechodů z každého stavu E je 1, pak i $\beta[R^i] = 1$. Pro účely důkazu je nyní možné R^i považovat za modrý stav se zpětnou pravděpodobností rovnou 1 a indukcí pokračovat, tj. opět najít poslední červený stav $R^{(i-1)}$ a opět dokázat, že $\beta[R^{(i-1)}] = 1$. □
- Pro přechody e^i vytvářené v kroku 3. algoritmu platí, že před optimalizací E je $\mathrm{P}(e^i \in E) = \mathrm{P}\left((u, y) \in U\right)$, neboť:

  $$\mathrm{P}(e^i \in E) = \alpha[p^i] \cdot w[e^i] \cdot \beta[n^i] = \alpha[p^i] \tag{8.15}$$

  a platí $w[e^i] = \beta[n^i] = 1$.



Obrázek 8.2: Vstupní mřížka U odpovídající promluvě *v sedmnáct devět v domažlicích.*



Obrázek 8.3: Jednoznačně přiřazené sémantické entity jsou pro názornost zaneseny do původní mřížky (zelené přechody). Pro vyšší přehlednost je dále pracováno pouze se sémantickými entitami s pravděpodobností vyšší než 0,001.

i	u^i	y^i	$p[u^i]$	$n[u^i]$	$\mathrm{P}\left((u^i, y^i) \in U_T\right)$
1	1	time:17:_h	0	7	0,934
2	1, 15	**time:17:_h:09:_m**	0	10	0,932
3	2	time:17:_h	0	5	0,051
4	2, 11	**time:17:_h:09:_m**	0	6	0,051
5	3, 4	time:17:_h	0	3	0,014
6	3, 4, 7	**time:17:_h:09:_m**	0	4	0,014
7	4	time:17:_h	1	3	0,014
8	4, 7	time:17:_h:09:_m	1	4	0,014
9	7	time:09:_h	3	4	0,014
10	9	**station:domažlice**	4	12	0,011
11	11	time:09:_h	5	6	0,051
12	13	**station:domažlice**	6	12	0,040
13	14, 21	**station:in:domažlice**	6	12	0,011
14	15	time:09:_h	7	10	0,932
15	19	**station:domažlice**	10	12	0,727
16	20, 21	**station:in:domažlice**	10	12	0,205
17	21	station:domažlice	11	12	0,221

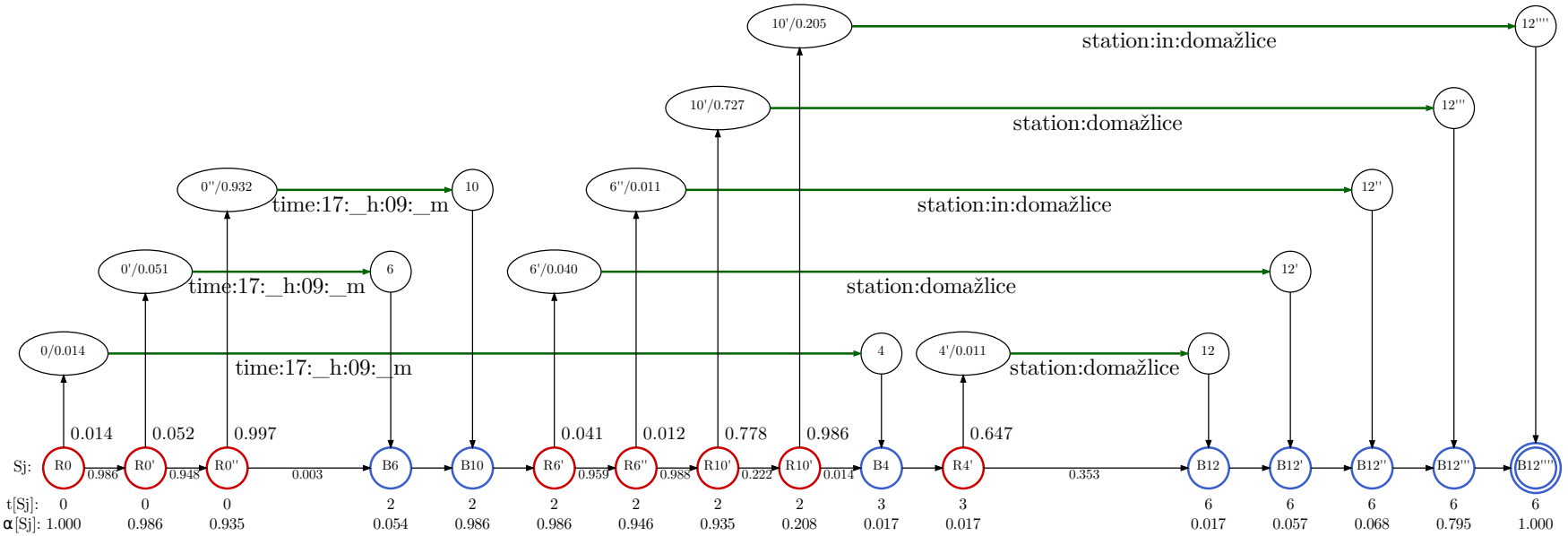
Tabulka 8.1: Prvky posloupnosti P_Z získané z U, tučně jsou zvýrazněny prvky P_Z^*, tj. jednoznačně přiřazené sémantické entity.

$$
\begin{aligned}
x_1 + x_2 &\leq 1 \\
x_2 + x_{14} &\leq 1 \\
x_3 + x_4 &\leq 1 \\
x_4 + x_{11} &\leq 1 \\
x_5 + x_6 &\leq 1 \\
x_5 + x_7 &\leq 1 \\
x_5 + x_8 &\leq 1 \\
x_6 + x_7 &\leq 1 \\
x_6 + x_8 &\leq 1 \\
x_6 + x_9 &\leq 1 \\
x_7 + x_8 &\leq 1 \\
x_8 + x_9 &\leq 1 \\
x_{13} + x_{17} &\leq 1 \\
x_{16} + x_{17} &\leq 1
\end{aligned}
$$

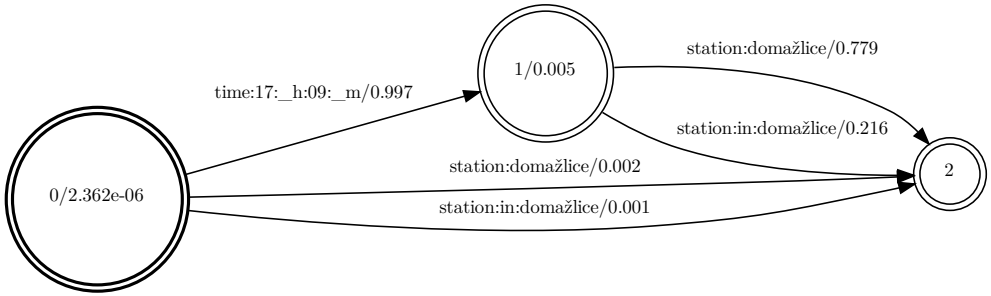
Tabulka 8.2: Množina omezovacích podmínek úlohy binárního celočíselného programování.

$s \in Q_U$	$t[s]$
0	0
1	1
2	2
3	2
4	3
5	1
6	2
7	1
8	3
9	4
10	2
11	5
12	6

Tabulka 8.3: Přiřazení času stavům automatu U, použito při sestavení akceptoru E.



Obrázek 8.4: Ilustrace rekonstrukce akceptoru sémantických entit z jejich seznamu, výpočtu dopředných pravděpodobností a vah přechodů, byl použit pravděpodobnostní polookruh, neuvedené symboly přechodů odpovídají ϵ, neuvedené váhy pak 1, počáteční stav akceptoru je R0, koncový B12''''.



Obrázek 8.5: Výsledný optimalizovaný vážený konečný akceptor reprezentující pravděpodobnostní rozdělení $\mathrm{P}(\mathsf{E} = e|\mathsf{U} = U)$.

$\mathrm{P}(\mathsf{E} = e\|\mathsf{U} = U)$	Posloupnost e
0,77652	time:17:_h:09:_m, station:domažlice
0,21542	time:17:_h:09:_m, station:in:domažlice
0,00489	time:17:_h:09:_m
0,00225	station:domažlice
0,00063	station:in:domažlice

Tabulka 8.4: Výsledné posloupnosti e a pravděpodobnosti $\mathrm{P}(\mathsf{E} = e|\mathsf{U} = U)$.

Kapitola 9

Definice úlohy

V této kapitole je popsána konfigurace experimentů pro vyhodnocení hierarchického diskriminativního modelu, modelu detekce sémantických entit a jejich kombinace. Byly použity dva sémanticky anotované korpusy obsahující zvukové nahrávky, jejich referenční textový přepis a a přiřazené abstraktní sémantické stromy – korpus *HHTT* (Human-Human Train Timetable, kapitola 9.1) a korpus *TIA* (Telefonní Inteligentní Asistentka, kapitola 9.2). Tato kapitola rovněž obsahuje popis způsobu vyhodnocení a to jak pro hierarchický diskriminativní model (kapitola 9.3), tak pro model detekce sémantických entit (kapitola 9.3.2). Dále jsou stručně popsány akustické a jazykové modely použité při rozpoznávání promluv ze sémantických korpusů HHTT a TIA (kapitola 9.4).

9.1 Korpus HHTT

První z úloh, na niž byly prezentované metody ověřovány, byla úloha *Nádraží* řešená na Katedře kybernetiky Západočeské univerzity v Plzni jako modelová úloha pro vývoj hlasových dialogových systémů nové generace v rámci projektu Centra aplikované kybernetiky (CAK). Předmětem této úlohy je vývoj hlasového dialogového systému pro podávání informací o odjezdech a příjezdech vlaků. Pro vývoj tohoto hlasového dialogového systému byl sestaven korpus *HHTT* (Human-Human Train Timetable) [57]. Tento sémanticky anotovaný korpus byl zvolen záměrně, neboť výsledky lze porovnat s výsledky předcházejících prací vyhodnocovaných na tomtéž korpusu, především se jedná o práce [46, 58].

Korpus HHTT obsahuje záznamy dialogů probíhajících v rámci provozu informačního centra o odjezdech a příjezdech vlaků. Tyto dialogy probíhaly vždy mezi dvěma lidmi. Tato skutečnost je významná především z pohledu variability a obsahu různých promluv vyskytujících se v dialozích – v rámci dialogů člověk-člověk dochází mnohem méně často k nedorozuměním v porovnání s dialogy člověk-stroj. Navíc lidé při komunikaci používají i různé neřečové komunikační prostředky, do proudu zvuku vkládají neřečové události, např. ehm-hmm, ehm-mm apod.

Data byla sbírána v době od dubna 2000 do srpna 2000. Volající byli především Češi mluvící spontánní češtinou. Audio signál byl získán z analogové telefonní linky vzorkovaný frekvencí 8kHz a komprimovaný A-Law kompresí, přičemž oba kanály (operátor a uživatel) byly smíchány do jediného monofonního kanálu. Každá promluva byla rozdělena

na segmenty, přičemž každému segmentu byl následně přiřazen právě jeden dialogový akt [57].

Anotační schéma korpusu HHTT je založeno na anotačním schématu *DATE* (Dialogue Act Tagging for Evaluation, [127]). Dialogový akt se ve schématu DATE skládá ze tří dimenzí: Domain, Speech-Act, Task-Subtask. Schéma DATE se zaměřuje především na dialogy typu člověk-počítač, proto pro účely anotace korpusu HHTT bylo toto schéma modifikováno tak, aby pokrylo oblast dialogů typu člověk-člověk. Toho bylo dosaženo nahrazením dimenze Task-Subtask novou dimenzí nazvanou Semantics. Tato nová dimenze obsahuje abstraktní sémantickou anotaci dialogového aktu. Anotace dialogových aktů je tedy provedena ve třech dimenzích nazvaných Conversational-Domain, Speech-Act, Semantics.

Dimenze Conversational-Domain vymezuje oblast promluvy obsažené v dialogovém aktu. Tato dimenze může nabývat třech hodnot: *task*, *communication* a *frame*. Hodnota *task* označuje promluvu vedoucí ke splnění určitého cíle (např. dotaz na čas odjezdu). Hodnota *communication* určuje promluvy umožňující vzájemnou komunikaci uživatele a operátora (např. potvrzení přijatého dotazu). Hodnota *frame* popisuje promluvy týkající se stavu dialogu (např. zahájení nebo ukončení dialogu).

Dimenze Speech-Act určuje cíl dialogového aktu nezávisle na významu dialogového aktu. Speech-Act může nabývat následujících hodnot: *acknowledgment*, *apology*, *closing*, *explicit_confirmation*, *implicit_confirmation*, *instruction*, *offer*, *opening*, *present_info*, *request_info*, *speech_repair*, *status_report*, *thanking*, *verify*, *verify_neg*. Bližší popis těchto dimenzí lze najít v [46, 57].

Samotný význam dialogového aktu je vyjádřen pomocí abstraktní sémantické anotace. Tato abstraktní sémantická anotace je uložena v dimenzi Semantics. Tato dimenze zahrnuje pouze sémantickou informaci relevantní z hlediska dané úlohy. V problémové oblasti korpusu HHTT jsou nejčastější požadavky obsahující především jména stanic, časy odjezdů a příjezdů, typy vlaků a další entity shrnuté v tabulce 9.1. Ukázkový dialog z korpusu HHTT je zachycen v tabulce 9.2.

Pro trénování subsystému porozumění je využita pouze dimenze Semantics, která charakterizuje význam dané promluvy. Navíc byly pro trénování použity jak promluvy operátora, tak uživatele telefonní linky. Toto sloučení obou stran dialogu je odůvodnitelné, neboť dialogy probíhají v rámci totožné domény (vlaková spojení) a operátor i uživatel sdílí jak slovník, tak i množinu sémantických konceptů. Sloučením dojde ke zvýšení počtu trénovacích vět, což vede obecně k robustnějšímu modelu porozumění.

V průběhu zpracování dat anotátoři nejprve celou větu přepsali do textové reprezentace, následně ji rozdělili do jednotlivých dialogových aktů a těmto aktům určili abstraktní sémantickou anotaci ve tvaru stromu, nemuseli ovšem určovat zarovnaný sémantický strom, tj. přiřazení konceptů sémantické anotace jednotlivým slovům dialogového aktu. Abstraktní sémantické anotace nejsou ovlivněny modalitou promluvy, tj. otázka i odpověď mohou mít tutéž sémantickou anotaci. Modalita promluvy je následně rozlišována hodnotou dimenze Speech-Act.

Pro účely automatického rozpoznávání řeči bohužel nemohl být použit celý korpus HHTT, neboť tento korpus byl nahráván pomocí analogové telefonní linky a veškerý zvukový signál dialogu je tedy v jediném mono signálu. Proto je velice problematické separovat jednotlivé strany dialogu (operátor a uživatel) do samostatných promluv obsahujících vždy pouze

koncept	*popis*
ACCEPT	Souhlas
AMOUNT	Peněžní obnos k zaplacení
AREA	Místo, odkud uživatel telefonuje
ARRIVAL	Otázka/odpověď na příjezd
BACK	Otázka/odpověď na zpáteční spoj
DELAY	Informace o zpoždění
DEPARTURE	Otázka/odpověď na odjezd
DISCONNECT	Informace o odpojení vagónů
DISTANCE	Otázka/odpověď na vzdálenost
DURATION	Doba jízdy
FROM	Odjezdová stanice
GREETING	Pozdrav
LENGTH	Vzdálenost stanic
MAYBE	Nejistá odpověď
NEXT	Otázka/odpověď na následující spoj
NUMBER	Číslo nereprezentující čas a cenu
OTHER_INFO	Jiná informace
PERSON	Jméno volající osoby
PLATFORM	Informace o nástupišti
PREVIOUS	Otázka/odpověď na předchozí spoj
PRICE	Otázka/odpověď na cenu
REF	Odkaz na již zmíněné
REJECT	Nesouhlas, odmítnutí
REPEAT	Požadavek na zopakování
STATION	Informace o stanici
SYSTEM_FEATURE	Dotaz na služby operátora
THROUGH	Průjezdní stanice
TIME	Čas nebo datum
TO	Cílová stanice
TRAIN_TYPE	Typ vlaku
TRANSFER	Informace o přestupu
WAIT	Informace o čekání
WHAT_TIME	Dotaz na čas nebo datum

Tabulka 9.1: Sémantické koncepty korpusu HHTT

č.	*ml.*	*promluva*	*abstraktní sémantický strom*
1	O	informace prosím	GREETING
2	U	dobrý den	GREETING
		já bych potřebovala zítra ráno kolem osmé nebo sedmé nějaký vlak do prahy	DEPARTURE(TIME, TO(STATION))
3	O	takže buď vám jede šest třicet šest rychlík	DEPARTURE(TIME, TRAIN_TYPE)
		ten je v praze osm deset	ARRIVAL(TO(STATION), TIME)
		a nebo rychlík osm nula dva	DEPARTURE(TRAIN_TYPE, TIME)
		a praha devět čtyřicet šest	ARRIVAL(TO(STATION), TIME)
4	U	devět čtyřicet šest	TIME
5	O	ano	ACCEPT
6	U	a všechno jsou to rychlíky	TRAIN_TYPE
7	O	oba dva ano	ACCEPT
8	U	děkuji	—
		nashledanou	—
9	O	není zač	—
		nashledanou	—

Tabulka 9.2: Ukázka anotovaného dialogu (dimenze SEMANTICS) z korpusu HHTT. Sloupec *ml.* určuje mluvčího (O – operátor, U – uživatel).

jediného mluvčího. Byl tedy použit automatický postup, kdy nejprve podle anotovaných časů změn řečníka byly dialogy rozděleny na jednotlivé promluvy. Tyto promluvy však stále mohly obsahovat překrývající se řečníky. Byla proto použita automatická kontrola správnosti slovní anotace [128], kde byl použit slovní rozpoznávač řeči v módu forced-alignment (kapitola 8, strana 8). V tomto módu slovní rozpoznávač pro danou slovní hypotézu určuje pouze časové zarovnání stavů HMM modelu se vstupní promluvou. Následně automatickým zpracováním tohoto časování byly na základě natrénovaných rozhodovacích pravidel zamítnuty ty promluvy, které neodpovídají slovní anotaci. Při rozhodování se berou v úvahu především časy trvání jednotlivých stavů HMM modelu, neboť při nesprávné anotaci zpravidla některá slova přebývají (řečníci se úplně nepřekrývají a zvukový signál obsahuje slova obou časově oddělená) nebo naopak některá slova chybí (řečníci se úplně překrývají, zvukový signál neodpovídá ani jednomu ze slov) – v těchto případech pak doba trvání jednotlivých stavů HMM je příliš krátká nebo dlouhá proti běžným statistikám a tudíž je možné anotaci označit za chybnou.

Následně bylo provedena redukce promluv tak, aby do dalšího zpracování postoupily pouze promluvy obsahující jediný dialogový akt. Rozdělení promluvy na jednotlivý dialogové akty na úrovni zvukového signálu je sice možné dle informací z automatického zarovnání jednotlivých slov, nicméně vzhledem k existenci koartikulace by toto postihlo především slova na hranici dialogového aktu. Opačný postup – sloučení dialogových aktů do jednoho u těch promluv, které jsou tvořeny více dialogovými akty – je v konfliktu s anotačním schématem. Ilustrujme tento jev na následující promluvě operátora z korpusu HHTT:

jinak tam není žádný rozdíl | jede to stejně v osm dvacet tři každý den
paní deset padesát sedm každý den ať je pátek nebo svátek

Zde symbol | označuje místo, ve kterém je původní promluva rozdělena na dva dialogové akty s následující sémantickou anotací v podobě abstraktního sémantického stromu:

promluva	SEMANTICS
jinak tam není žádný rozdíl	REJECT(OTHER_INFO)
jede to stejně v osm dvacet tři každý den paní deset padesát sedm každý den ať je pátek nebo svátek	TIME, TIME

Pokud bychom však celou tuto promluvu považovali za jediný sloučený dialogový akt, pak by zcela jistě byla postačující sémantická anotace TIME, TIME. Tímto bychom však do experimentu zavedli chybu, neboť referenční sémantická anotace by odpovídala REJECT(OTHER_INFO), TIME, TIME.

V tabulce 9.3 jsou uvedeny vlastnosti korpusu HHTT po odstranění promluv, které neodpovídají slovní transkripci, a také promluv, které obsahují více jak jeden dialogový akt.

Dále vhledem k omezením na tvar sémantického stromu popsaným v kapitole 7.3.2 (strana 77), konkrétně generování pouze neuspořádaných stromů pomocí HDM (omezení 1), je vhodné tomu uzpůsobit způsob vyhodnocení. V tomto případě tak, že všechny shodné sémantické koncepty, které jsou na stejné úrovni sémantického stromu a mají společného rodiče, nahradíme pouze jediným sémantickým konceptem. Násobné sémantické koncepty na stejné úrovni sémantického stromu jsou téměř výhradně listové sémantické koncepty. V celé uvažované podmnožině korpusu HHTT použité pro trénování HDM jsou řádově pouze jednotky promluv, ve kterých se vyskytují násobné sémantické koncepty, které nejsou listy abstraktního sémantického stromu a liší se množinou potomků, tj. porušující omezení 3 z kapitoly 7.3.2.

Příklad: Pro ilustraci výše zmíněného předzpracování uvažujme promluvu, která má abstraktní sémantický strom ve tvaru:

DEPARTURE(TIME, TO(STATION), TIME), ARRIVAL(TIME)

Pro účely vyhodnocení je pak abstraktní sémantický strom převeden na strom:

DEPARTURE(TIME, TO(STATION)), ARRIVAL(TIME).

	train	*devel*	*test*
Počet vět	5240	570	1439
Celková délka h:m:s	2:40:25	0:17:22	0:44:59
ø doba 1 věty ($\pm\sigma$)	1,84±1,44	1,83±1,25	1,88±1,31
Počet tokenů	21517	2301	5838
ø počet tokenů 1 věty ($\pm\sigma$)	4,11±3,47	4,04±3,21	4,06±3,22
Velikost slovníku	1656	476	731
Četnost OOV	–	4,00%	7,45%
Počet konceptů v sémantických stromech	8967	997	2584
Počet unikátních konceptů	32	28	28
ø počet konceptů 1 věty ($\pm\sigma$)	1,71±1,24	1,75±1,24	1,80±1,30
Počet vět s 1 konceptem	3439	360	896
ø počet konceptů 1 věty ($\pm\sigma$), má-li věta více než 1 koncept	3,07±1,29	3,03±1,26	3,11±1,32

Tabulka 9.3: Vlastnosti korpusu HHTT.

9.2 Korpus TIA

Korpus telefonní inteligentní asistentky vznikl v rámci výzkumného projektu MPO TIP FR-TI1/518 Inteligentní telefonní asistentka. Tento projekt řešený firmou Speech-Tech s.r.o. a Katedrou kybernetiky Západočeské univerzity v Plzni si klade za cíl výzkum a vývoj hlasového dialogového systému Telefonní inteligentní asistentka (TIA). Tento systém byl měl poskytovat skupinám v rámci malých a středních podniků unifikované hlasové rozhraní k nástrojům pro organizaci času – především k osobním a sdíleným kalendářům, k plánování sdílených prostředků jako jsou automobily, projektory, zasedací místnosti apod. Další z funkcionalit by měla umožňovat napojení na telefonní seznam organizace a spojování přímých i konferenčních hovorů.

Pro účely vývoje tohoto hlasového dialogového systému a výzkumu metod pro porozumění řeči byl nahrán a anotován řečový korpus. Tento korpus obsahuje dvě části. První z nich byla nahrávána pomocí hlasového dialogového systému, který simuloval chování budoucího systému pomocí posloupnosti jednotlivých poddialogů. Tato část je složena ze 187 dialogů a 2469 vět. Druhá část korpusu byla zaměřena na cílený sběr promluv obsahujících vybrané sémantické entity, dotazovaní odpovídali na předpřipravené otázky typu *Kdy jste se narodil?* (datum), *Kdy je neděle vzhledem k dnešku?* (relativní datum), *Pršelo včera?* (souhlas/nesouhlas). Na závěr dialogu byl ponechán prostor pro vyslovení několika simulovaných požadavků.

Promluvy mnohdy obsahovaly odpovědi a požadavky, které nebyly konzistentní s danou úlohou (např. *Pršelo včera? – Myslím, že pršelo*). To vedlo k zavedení sémantického konceptu označujícího promluvy mimo téma dialogu (OOT). Dále byly definovány sémantické koncepty definující jednotlivé *sémantické entity* – např. datum, časy, jména lidí a prostředků. Tabulka 9.5 přináší seznam těchto sémantických entit. Další množina sémantických konceptů popisuje *sémantické akce*, tj. příkazy, které celá promluva nese a které má hlasový dialogový systém vykonávat. Výčet možných sémantických akcí obsahuje tabulka 9.6. Posledním podmnožinou sémantických konceptů jsou *sémantické cíle* definující funkcionalitu dialogového systému, např. skupinové plánování schůzek, osobní kalendář,

spojovatelka. Sémantické cíle jsou shrnuty v tabulce 9.7. Takto získaná druhá část korpusu obsahuje 210 dialogů a 3698 vět.

Data byla nahrána prostřednictvím digitálního telefonního rozhraní ISDN. Byl vytvořen automatický nástroj [129] umožňující import nahrávek z telefonního rozhraní do nástroje WebTransc tak, aby bylo možné tyto nahrávky dále anotovat. Anotace na slovní a významové úrovni proběhla najednou, tj. anotátor každou promluvu nejprve přepsal na úrovni slov a následně do ní vložil popis významu pomocí sémantických konceptů. Tento přístup urychluje anotaci, neboť anotátor má přehled o aktuální větě a jejím kontextu bez opakovaného přehrávání nahrávky.

Ukázkový dialog z korpusu TIA je zachycen v tabulce 9.4. V porovnání s korpusem HHTT je sémantická anotace korpusu TIA jednodušší, je anotována pouze jediná dimenze. Modalita promluvy je vyjádřena sémantickými akcemi, podoblast dialogu sémantickými cíli a informace parametrizující uživatelův požadavek pak sémantickými entitami.

promluva	*abstraktní sémantický strom*
potřebovala bych rezervovat zasedací místnost na čtvrtek ve čtyři hodiny	VYTVOR(REZERVACE(VEC, DATUM, T))
je tento termín volný	ZJISTI(KALENDAR)
potom potřebuju zasedací místnost na úterý to je zítra o+	VYTVOR(REZERVACE(VEC, DATUM))
na zítra v deset hodin zasedací místnost číslo tři	VYTVOR(REZERVACE(DATUM, T, VEC))
ne to stačí	NE
všechny problémy mám vyřešené	OOT
již nemám další dotaz ukončím tuto anketu	NE
jaké schůzky mám sjednaný na p+ sjednané na příští týden	ZJISTI(KALENDAR(RELATIVNI))
lze zjistit jaké schůzky mám zjiš+ zji+ sjednané na příští týden	ZJISTI(KALENDAR(RELATIVNI))
ne již nemám žádné přání děkuji	NE, DIKY
ne všechny problémy mám vyřešené	NE

Tabulka 9.4: Ukázkové promluvy z korpusu TIA, slova ukončená symbolem + jsou nedořeky.

entita	*popis*
HELLO	Pozdrav, včetně představení se
DIKY	Poděkování
BYE	Rozloučení
DATUM	Vyjádření data nezávislé na aktuálním dni
T	Vyjádření času, včetně předložek
MISTO	Vyjádření místa
JMENO	Jméno osoby
VEC	Jméno prostředku
SOUHLAS	Explicitní souhlas
NE	Explicitní nesouhlas
SUBJECT	Název události
INTERVAL	Časový interval
RELATIVNI	Relativní čas/datum

Tabulka 9.5: Sémantické entity korpusu TIA

akce	*popis*
ZJISTI	Dotaz na objekt
VYTVOR	Vytvoření objektu
ZRUS	Zrušení objektu
UPRAV	Změna objektu

Tabulka 9.6: Sémantické akce korpusu TIA

cíl	*popis*
OOT	Out-of-topic, mimo řešenou doménu
KALENDAR	Osobní kalendář, osobní události
SCHUZKY	Skupinové plánování
REZERVACE	Rezervace prostředků
SPOJ	Spojení hovoru
KONFERENCE	Konferečního hovory
POZNAMKA	Čtení a uložení poznámek

Tabulka 9.7: Sémantické cíle korpusu TIA

	train	*devel*	*test*
Počet vět	4166	452	1054
Celková délka h:m:s	6:45:57	0:40:30	1:39:38
ø doba 1 věty ($\pm\sigma$)	3,80±1,34	3,81±1,30	3,83±1,29
Počet tokenů	33562	3501	8387
ø počet tokenů 1 věty ($\pm\sigma$)	7,74±7,63	7,46±7,28	7,82±7,34
Velikost slovníku	2655	703	1181
Četnost OOV	–	4,14%	8,62%
Počet konceptů v sémantických stromech	9027	1017	2305
Počet unikátních konceptů	25	24	24
ø počet konceptů 1 věty ($\pm\sigma$)	2,08±1,60	2,17±1,62	2,15±1,61
Počet vět s 1 konceptem	2662	274	637
Počet vět označených OOT	915	74	210
ø počet konceptů 1 věty ($\pm\sigma$), má-li věta více než 1 koncept	3,80±1,34	3,81±1,30	3,83±1,29

Tabulka 9.8: Vlastnosti korpusu TIA.

9.3 Metriky použité pro vyhodnocení

Pro objektivní vyhodnocení výstupů modelu porozumění je nutné definovat vhodné míry, kterými je možné vyčíslit shodu predikovaného výstupu sémantického klasifikátoru s referenční sémantickou anotací poskytnutou anotátorem.

V práci [55] a dalších byly pro vyhodnocení přesnosti používány míry často používané v oblasti systémů pro získávání informací (information retrieval). Jedná se o míry úplnost (recall) R, přesnost (precision) P a F-skóre [130]:

$$R = \frac{TP}{TP + FN} \tag{9.1}$$

$$P = \frac{TP}{TP + FP} \tag{9.2}$$

$$F = 2\frac{P \cdot R}{R + P} \tag{9.3}$$

kde TP je počet správně predikovaných výskytů daného jevu; FP je počet chyb, kdy jev byl predikován, ale referenční data jej neobsahují; FN je počet chyb, kdy jev nebyl predikován, ale referenční data jej obsahují (více v kapitole 9.3.2). F-skóre je pak definováno jako harmonický průměre měr P a R.

V práci [55] bylo na úlohu porozumění nahlíženo jako na úlohu získávání informací, ve které jsou ze vstupní promluvy získávány dvojice *atribut-hodnota*, kde *atribut* byla n-tice sémantických konceptů s přiřazenou lexikální *hodnotou*. V tomto případě je však nutné mít i testovací množinu ve stejném tvaru. V případě korpusu ATIS [56] byly tyto hodnoty přímo jeho součástí. Stejný způsob vyhodnocení byl použit i v případě dalších modelů aplikovaných nad korpusem ATIS [59, 65].

V případech, kdy výše zmíněné přiřazení *atribut-hodnota* není v testovacích datech zahrnuto, lze použít metriky vyhodnocující podobnost predikovaného abstraktního sémantického stromu a anotovaného referenčního stromu.

Nejjednodušší z těchto metrik je *větná přesnost sAcc* (sentence accuracy) definovaná jako:

$$sAcc = \frac{E}{N} \tag{9.4}$$

kde E je počet vět testovací množiny, pro které se predikovaný a referenční strom shodují a N je celkový počet vět testovací množiny. Přesná shoda predikovaného a referenčního stromu však neodráží chyby na úrovni jednotlivých uzlů sémantického stromu.

Pokud jsou referenční sémantické stromy zarovnané, pak je možné použít pro vyhodnocení přesnosti sémantického modelu metodiku PARSEVAL [131, 132] spočívající ve vyčíslení metrik přesnosti, úplnosti a F-míry nad uzly predikovaného a referenčním stromu. Uzel v predikovaném stromu je vyhodnocen jako správně označený vzhledem k referenčnímu stromu, pokud odpovídající uzel v referenčním stromu je označen stejným konceptem a zároveň pokud oba uzly mají stejnou lexikální realizaci. Je nutné podotknout, že metodika PARSEVAL nutně vyžaduje sémantické stromy plně zarovnané s lexikální realizací – a to jak pro predikované, tak pro referenční stromy. Z tohoto důvodu je tato metodika nevhodná pro vyhodnocení přesnosti navrženého modelu, neboť jeho výhoda leží právě v možnosti trénování s použitím abstraktních sémantických stromů. Tyto stromy jsou také výstupem modelu pro neznámé promluvy.

Z tohoto důvodu byla navržena metodika vyhodnocení využívající algoritmus pro výpočet editační vzdálenosti mezi dvěma abstraktními sémantickými stromy. Mějme seznam L, tvořený editačními operacemi l_i ve tvaru:

- $a \to a$ (shoda)
- $a \to b$ (substituce konceptu a konceptem b)
- $a \to \lambda$ (odstranění konceptu a)
- $\lambda \to b$ (vložení konceptu b)

Cenu operace l_i definujme jako nezápornou reálnou funkci $\gamma(l_i)$. Funkce $\gamma(l_i)$ musí být metrikou, tj. musí platit nezápornost ($\gamma(a \to b) \geq 0$), totožnost ($\gamma(a \to a) = 0$), symetrie ($\gamma(a \to b) = \gamma(b \to a)$) a trojúhelníková nerovnost ($\gamma(a \to c) \leq \gamma(a \to b) + \gamma(b \to c)$).

Definujme nyní cenu seznamu editačních operací $\gamma(L)$:

$$\gamma(L) = \sum_{l_i \in L} \gamma(l_i) \tag{9.5}$$

Editační vzdálenost $D(T_1, T_2)$ mezi stromy T_1 a T_2 je pak definována jako:

$$D(T_1, T_2) = \min_{L} \{\gamma(L) | L \text{ je posloupnost editačních operací převádějících } T_1 \text{ na } T_2\}$$

Vzhledem k definici funkce γ je editační vzdálenost opět metrikou [133].

Pro nalezení editační vzdálenosti mezi uspořádanými sémantickými stromy, tj. stromy, kde mezi následovníky libovolného uzlu vždy existuje relace uspořádání, lze použít algoritmus popsaný v [134] vedoucí na úlohu dynamického programování. Algoritmus uvedený v práci [135] převádí problém editační vzdálenosti mezi dvěma uspořádanými stromy na

problém editační vzdálenosti mezi dvěma řetězci [43], které reprezentují uzavřenou Eulerovskou cestu stromem počínající v jeho kořenovém uzlu. Tyto algoritmy je možné použít nejen k efektivnímu vyčíslení editační vzdálenosti, ale i k získání posloupnosti editačních operací, jejichž postupná aplikace na predikovaný a referenční strom vede k získání dvou identických stromů.

Hierarchický diskriminativní model popisovaný v této práci však generuje neuspořádané sémantické stromy, neboť potomci libovolného konceptu tvoří množinu, nikoli uspořádanou posloupnost. Obecný algoritmus pro výpočet editační vzdálenosti dvou neuspořádaných stromů je NP-úplná úloha [133]. Proto je vhodné omezit možná přiřazení uzlů stromu T_1 uzlům stromu T_2 tak, aby dva nepřekrývající se podstromy v T_1 byly přiřazeny opět nepřekrývající se podstromům v T_2 [136, 137]. Takto definované přiřazení uzlů zachovává strukturu jednotlivých podstromů, jak je naznačeno na obrázku 9.1. Oproti tomu přiřazení uzlů na obrázku 9.2 strukturu podstromů nezachovává.

V dostupných pramenech lze nalézt jistou kritiku směřující k používání editační vzdálenosti pro účely vyhodnocení přesnosti syntaktických a sémantických parserů, např. [54, str. 81] nebo [138], zdůvodňovanou především výraznou penalizací výsledného skóre za každou chybu vložení. Nicméně tuto kritiku je nutné uvažovat v širším kontextu – zmiňované práce byly věnovány především evaluaci systémů v úlohách zpracování přirozeného jazyka. V oblasti výzkumu hlasových dialogových systémů, kam je i tato práce příspěvkem, jsou chyby vložení (tj. vygenerování významu, který ve vstupní promluvě obsažen není) zásadním problémem, neboť subsystém řízení dialogu následně musí pomocí několika dialogových obrátek provést odstranění tohoto chybně vygenerovaného významu ze svého stavu.

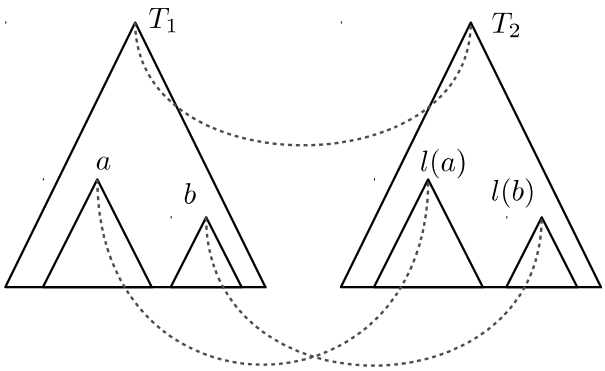
Vyhodnocení přesnosti porozumění řeči založené na editační vzdálenosti stromů lze interpretovat podobně jako vyhodnocení přesnosti rozpoznávání řeči založené na editační vzdálenosti posloupností slov. Díky existenci zarovnání dvou (sémantických) stromů lze využít i další metriky, například pro vyhodnocení statistické významnosti [139] nebo pro výpočet intervalů spolehlivosti [140].

Předpokládejme testovací množinu $\mathcal{T}_e = \{(u_i, s_i)\}_{i=1}^{n}$, kde u_i je vstupní promluva a s_i je odpovídající referenční sémantický strom. Při vyhodnocení přesnosti daného modelu je pro vstupní promluvu u_i vygenerován predikovaný, hypotetický sémantický strom $\hat{s}_i$. Označme seznam editační operací převádějících s_i na $\hat{s}_i$ jako L_i^*:

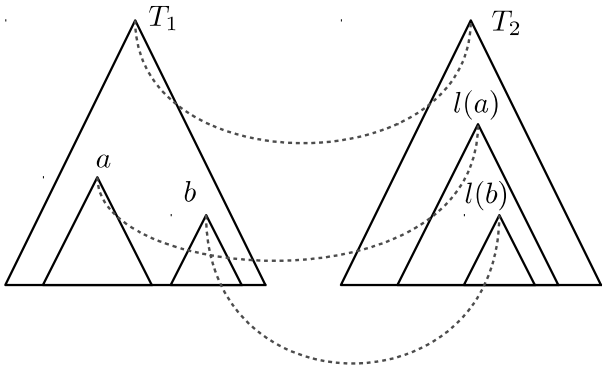
$$L_i^* = \text{argmin}_{L_i} \{\gamma(L_i) | L_i \text{ je posloupnost editačních operací převádějících } s_i \text{ na } \hat{s}_i\}$$

Definujme pomocnou funkci $\delta_l(\cdot, \cdot)$ jako:

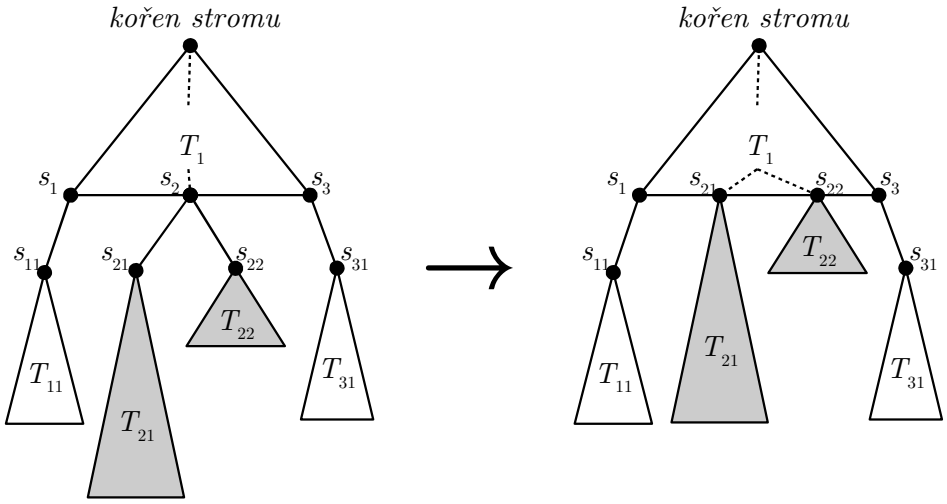
$$\delta_l(a, b) = \begin{cases} 1 & \text{pokud } l \text{ je ve tvaru } a \to b \\ 0 & \text{jinak} \end{cases} \tag{9.6}$$



Obrázek 9.1: Zarovnání uzlů stromů T_1 a T_2 zachovávající strukturu.



Obrázek 9.2: Zarovnání uzlů stromů T_1 a T_2 nezachovávající strukturu.



Obrázek 9.3: Vypuštění uzlu s_2 ze stromu, jeho následovníci s_{21} a s_{22} jsou vloženi na úroveň původního uzlu.

Potom můžeme pro danou testovací množinu $\mathcal{T}_e$ zadefinovat čísla:

- Počet správných konceptů: $H = \sum_{i=1}^{n} \sum_{l \in L_i^*} \delta_l(a, a)$
- Počet referenčních konceptů: $N = \sum_{i=1}^{n} |s_i|$
- Počet chyb vynechání: $D = \sum_{i=1}^{n} \sum_{l \in L_i^*} \delta_l(a, \lambda)$
- Počet chyb vložení: $I = \sum_{i=1}^{n} \sum_{l \in L_i^*} \delta_l(\lambda, a)$
- Počet chyb substituce: $S = \sum_{i=1}^{n} \sum_{l \in L_i^*} \delta_l(a, b), \quad a \neq b$
- Počet správných predikcí konceptu C : $H_C = \sum_{i=1}^{n} \sum_{l \in L_i^*} \delta_l(C, C)$
- Počet chyb vynechání konceptu C : $D_C = \sum_{i=1}^{n} \sum_{l \in L_i^*} \delta_l(C, a), \quad a \neq C$
- Počet chyb vložení konceptu C : $I_C = \sum_{i=1}^{n} \sum_{l \in L_i^*} \delta_l(a, C), \quad a \neq C$

Analogicky k mírám slovní přesnosti *Acc* a správnosti *Corr* (viz kapitola Rozpoznávání řeči, str. 10) definujme *konceptovou přesnost cAcc* a *konceptovou správnost cCorr*. Tyto míry slouží k vyhodnocení přesnosti predikce sémantických stromů tvořených sémantickými koncepty, přičemž jsou schopny detailně podchytit i míru shody mezi dvěma různými stromy (oproti *sAcc*, která zachycuje pouze rovnost/nerovnost dvou stromů):

$$cAcc = \frac{N - D - S - I}{N} = \frac{H - I}{N} \tag{9.7}$$

$$cCorr = \frac{H}{N} \tag{9.8}$$

Nebude-li řečeno jinak, je přesnost modelu porozumění vyhodnocována vždy pomocí konceptové přesnosti *cAcc*. Dále pro vyhodnocení přesnosti P_C a úplnosti R_C a jejich harmonického průměru F_C na úrovni jednotlivých sémantických konceptů lze zavést míry vyčíslované pro každý sémantický koncept C:

$$R_C = \frac{H_C}{H_C + D_C} \tag{9.9}$$

$$P_C = \frac{H_C}{H_C + I_C} \tag{9.10}$$

$$F_C = 2\frac{R_C \cdot P_C}{R_C + P_C} \tag{9.11}$$

9.3.1 Intervaly spolehlivosti

Pro vyhodnocení intervalů spolehlivosti pro konceptovou přesnost byl použit přístup z práce [140], ve které byly odvozeny vztahy pro intervaly spolehlivosti pro četnost slovních chyb (Word Error Rate, *WER*, 10). Toto odvození vychází z práce [141], kde byl pro získání intervalů spolehlivosti použit tzv. bootstrapping (opakované vzorkování množiny dat). Práce [140] pak nabízí efektivní způsob výpočtu, který vychází z předpokladu, že bootstrapping je validní, nicméně používá přímý výpočet, nikoli opakované vzorkování. Pro úplnost zde uveďme odvození výsledných vztahů pro (konceptovou) přesnost.

Předpokládejme n testovacích příkladů z testovací množiny $\mathcal{T}_e = \{(u_i, s_i)\}_{i=1}^n$, pak konceptová přesnost je vyjádřena jako:

$$cAcc = \frac{N - D - S - I}{N} = \frac{\sum_{i=1}^n N_i - E_i}{\sum_{i=1}^n N_i} \tag{9.12}$$

kde $N_i = |s_i|$ je počet sémantických konceptů i-tého sémantického stromu a $E_i = \sum_{l \in L_i^*} \delta_l(a, b) + \delta_l(a, \lambda) + \delta_l(\lambda, a), \quad a \neq b$ je počet chyb při predikci i-tého sémantického stromu.

Předmětem řešení je pak nalezení intervalů spolehlivosti pro náhodnou proměnnou A definovanou jako:

$$\mathsf{A} = \frac{\sum_{i=1}^n \mathsf{N}_i - \mathsf{E}_i}{\sum_{i=1}^n \mathsf{N}_i} \tag{9.13}$$

přičemž se předpokládá, že hodnoty N_i a E_i vznikly vzorkováním náhodných proměnných N_i, resp. E_i, kde dvojice $(\mathsf{E}_i, \mathsf{N}_i)$ jsou i.i.d. (independent and identically distributed, nezávislé a identicky rozdělená náhodné proměnné). Vyjádřeme nyní distribuční funkci náhodné proměnné A:

$$\mathrm{P}(\mathsf{A} < x) = \mathrm{P}\left(\frac{\sum_{i=1}^n \mathsf{N}_i - \mathsf{E}_i}{\sum_{i=1}^n \mathsf{N}_i} < x\right) \tag{9.14}$$

Protože $N_i > 0$, pak lze psát:

$$\begin{aligned}
\mathrm{P}(\mathsf{A} < x) &= \mathrm{P}\left(\sum_{i=1}^n \mathsf{N}_i - \mathsf{E}_i < x \cdot \sum_{i=1}^n \mathsf{N}_i\right) \\
&= \mathrm{P}\left(\sum_{i=1}^n \mathsf{N}_i - \mathsf{E}_i - x\mathsf{N}_i < 0\right) \\
&= \mathrm{P}\left(\sum_{i=1}^n (1-x)\mathsf{N}_i - \mathsf{E}_i < 0\right)
\end{aligned} \tag{9.15}$$

Zadefinujme náhodné proměnné $\mathsf{Z}_i^x = (1-x)\mathsf{N}_i - \mathsf{E}_i$, které jsou opět i.i.d. Potom:

$$\begin{aligned}
\mathrm{P}(\mathsf{A} < x) &= \mathrm{P}\left(\sum_{i=1}^n \mathsf{Z}_i^x < 0\right) \\
&= \mathrm{P}\left(\sum_{i=1}^n \mathsf{Z}_i^x - n\mathrm{E}(\mathsf{Z}^x) < -n\mathrm{E}(\mathsf{Z}^x)\right) \\
&= \mathrm{P}\left(\frac{\sum_{i=1}^n \mathsf{Z}_i^x - n\mathrm{E}(\mathsf{Z}^x)}{\sqrt{n}\sigma(\mathsf{Z}^x)} < \frac{-\sqrt{n}\mathrm{E}(\mathsf{Z}^x)}{\sigma(\mathsf{Z}^x)}\right)
\end{aligned} \tag{9.16}$$

kde $\mathrm{E}(\mathsf{Z}^x)$ je střední hodnota veličiny Z^x a $\sigma(\mathsf{Z}^x)$ její směrodatná odchylka. Podle centrální limitní věty se pak levá strana nerovnosti pro dostatečně velké n blíží normálnímu normovanému rozdělení:

$$\lim_{n\to\infty} \mathrm{P}\left(\frac{\sum_{i=1}^n \mathsf{Z}_i^x - n\mathrm{E}(\mathsf{Z}^x)}{\sqrt{n}\sigma(\mathsf{Z}^x)} < \frac{-\sqrt{n}\mathrm{E}(\mathsf{Z}^x)}{\sigma(\mathsf{Z}^x)}\right) = \Phi\left(\frac{-\sqrt{n}\mathrm{E}(\mathsf{Z}^x)}{\sigma(\mathsf{Z}^x)}\right) = \Phi\left(f(x)\right) \tag{9.17}$$

$$\frac{-\sqrt{n}\mathrm{E}(\mathsf{Z}^x)}{\sigma(\mathsf{Z}^x)} = f(x) \tag{9.18}$$

Nalezení intervalu spolehlivosti poté spočívá ve vyřešení rovnice (9.18) pro neznámou x tak, aby $\Phi(f(x))$ nabývalo mezních hodnot intervalu spolehlivosti.

Pro $\alpha \cdot 100\%$ oboustranný interval spolehlivosti pak požadujeme, aby $\Phi(f(x)) \in [\frac{1-\alpha}{2}, \frac{1+\alpha}{2}]$. Hledejme tedy krajní body x_1 a x_2, pro které nastává rovnost $\Phi(f(x_{1,2})) = \frac{1\pm\alpha}{2}$. Odtud:

$$\begin{aligned} x_{1,2} &= \Phi^{-1}\left(\frac{1 \pm \alpha}{2}\right) \\ &= \pm\Phi^{-1}\left(\frac{1-\alpha}{2}\right) \end{aligned} \tag{9.19}$$

Vyjádříme $\mathrm{E}(\mathsf{Z}^x)$ a $\sigma(\mathsf{Z}^x)$ jako:

$$\mathrm{E}(\mathsf{Z}^x) = (1-x)\mathrm{E}(\mathsf{N}) + \mathrm{E}(\mathsf{E}) \tag{9.20}$$

$$\sigma(\mathsf{Z}^x) = \sqrt{(1-x)^2\sigma^2(\mathsf{N}) + \sigma^2(\mathsf{E}) + 2(1-x)\operatorname{cov}(\mathsf{N},\mathsf{E})} \tag{9.21}$$

kde $\sigma^2(\cdot)$ značí rozptyl a $\operatorname{cov}(\cdot,\cdot)$ kovarianci a dosadíme do (9.18) a (9.19), získáme:

$$\frac{-\sqrt{n}\left[(1-x)\mathrm{E}(\mathsf{N}) + \mathrm{E}(\mathsf{E})\right]}{\sqrt{(1-x)^2\sigma^2(\mathsf{N}) + \sigma^2(\mathsf{E}) + 2(1-x)\operatorname{cov}(\mathsf{N},\mathsf{E})}} = \pm\Phi^{-1}\left(\frac{1-\alpha}{2}\right) \tag{9.22}$$

Po zavedení substitucí $y = (1-x)$ a $l = \Phi^{-1}\left(\frac{1-\alpha}{2}\right)$, dosazení a přeuspořádání získáme následující kvadratickou rovnici, jejímiž řešeními jsou y_1 a y_2, resp. x_1 a x_2 určující hranici intervalu spolehlivosti, uvnitř kterého se s pravděpodobností α nachází hodnota $cAcc$, která vznikla vzorkováním náhodné proměnné A:

$$\begin{aligned} &y^2\left(\sigma^2(\mathsf{N}) - n\mathrm{E}^2(\mathsf{N})\right) \\ &+y\left(2n\mathrm{E}(\mathsf{N})\mathrm{E}(\mathsf{E}) - 2l^2\operatorname{cov}(\mathsf{N},\mathsf{E})\right) \\ &+1\left(l^2\sigma^2 E - n\mathrm{E}^2(\mathsf{E})\right) = 0 \end{aligned} \tag{9.23}$$

Samotná implementace výpočtu intervalu spolehlivosti spočívá v následujícím postupu:

- Určení $l = \Phi^{-1}\left(\frac{1-\alpha}{2}\right)$ pro zadanou šířku intervalu spolehlivosti α
- Odhad $\mathrm{E}(\mathsf{E})$, $\mathrm{E}(\mathsf{N})$, $\mathrm{E}(\mathsf{E}^2)$, $\mathrm{E}(\mathsf{N}^2)$ a $\mathrm{E}(\mathsf{NE})$ z testovací sady
- Výpočet $\sigma^2(\mathsf{E}) = \mathrm{E}(\mathsf{E}^2) - \mathrm{E}^2(\mathsf{E})$ a $\sigma^2(\mathsf{N}) = \mathrm{E}(\mathsf{N}^2) - \mathrm{E}^2(\mathsf{N})$.
- Výpočet $\operatorname{cov}(\mathsf{N},\mathsf{E}) = \mathrm{E}(\mathsf{NE}) - \mathrm{E}(\mathsf{N})\mathrm{E}(\mathsf{E})$.
- Výpočet hranic intervalu spolehlivosti $x_1 = 1 - y_1$, $x_2 = 1 - y_2$ z řešení y_1, y_2 kvadratické rovnice (9.23) parametrizované hodnotou l.

9.3.2 Vyhodnocení přesnosti detekce sémantických entit

Pro vyhodnocení přesnosti modulu detekce sémantických entit byla použita modifikovaná ROC křivka (Receiver Operating Characteristic). Ta vychází z klasické ROC křivky pro binární klasifikátor. Nejprve pro konkrétní binární klasifikátor klasifikující do tříd 0 nebo 1 zadefinujme následující čísla:

	reference	
predikce	1	0
1	*TP*	*FP*
0	*FN*	*TN*

Číslo *TP* (True Positives) znamená, kolikrát klasifikátor správně predikoval třídu 1; *FP* (False Positives) kolikrát klasifikátor chybně predikoval třídu 1, pokud v referenčních datech byla třída 0; *FN* (False Negatives) kolikrát klasifikátor chybně predikoval třídu 0, pokud v referenčních datech byla třídu 1; *TN* (True Negatives) kolikrát klasifikátor správně predikoval třídu 0. Následně můžeme definovat veličiny *TPR* (True Positives Rate) a *FPR* (False Positives Rate):

$$TPR = \frac{TP}{TP + FN}, \quad FPR = \frac{FP}{FP + TN} \tag{9.24}$$

Binární klasifikátor často pro neznámý příznakový vektor $\mathbf{x}$ nevrací pouze predikci cílové třídy, ale i určité skóre $d(\mathbf{x})$. Toto skóre může odpovídat například aposteriorní pravděpodobnosti nebo vzdálenosti k oddělovací nadrovině. Jako rozhodovací pravidlo je pak použito porovnání s daným prahem θ:

$$\hat{y} = \begin{cases} 1 & \text{pokud } d(\mathbf{x}) \geq \theta \\ 0 & \text{pokud } d(\mathbf{x}) < \theta \end{cases} \tag{9.25}$$

ROC křivka je pak definována jako křivka vyjadřující závislost *TPR* na *FPR* pro spojitě se měnící hodnotu prahu θ. ROC křivka začíná v bodě $[0; 0]$ a končí v bodě $[1; 1]$. Její průběh pomáhá určit vhodný pracovní bod klasifikátoru pomocí prahu θ, kdy poměr správně detekovaných příkladů (*TPR*) a četností falešných poplachů (*FPR*) odpovídá cílové úloze.

Při vyhodnocení systému detekce sémantických entit se však nejedná o binární klasifikaci – jedné vstupní promluvě je přiřazeno více různých hypotetických sémantických entit s odpovídajícími aposteriorními pravděpodobnostmi a rovněž referenční sémantické entity jsou tvořeny nikoli jedinou cílovou třídou, ale celou posloupností sémantických entit.

Veličinu *FPR* tak není možné počítat jako poměr počtu falešných poplachů (*FP*) a celkového počtu negativních příkladů (třída 0), neboť počet různých sémantických entit může být nekonečný a tudíž i počet negativních příkladů může být nekonečný. Proto zavádíme modifikovanou ROC křivku, která vyjadřuje závislost veličiny *TPR* na veličině FPR_{norm} při proměnném prahu θ. Normalizovaná četnost falešných poplachů FPR_{norm} je pak definována jako:

$$FPR_{norm} = \frac{FP}{n} \tag{9.26}$$

kde n je počet jednotek, ke kterému je vztažen počet falešných poplachů. V této práci bude jako n použit počet promluv v testovací množině. Veličina FPR_{norm} pak vyjadřuje relativní četnost falešných poplachů na jednu promluvu.

Při detekci sémantických entit je nutné vyřešit problém, jak získat referenční soubor sémantických entit k daným promluvám. Jednou z možností je pracovat se sémantickým korpusem obsahujícím zarovnané sémantické stromy, kde jsou jednotlivé sémantické entity označeny včetně jejich sémantického typu a sémantické interpretace. V tomto případě je možné validovat i pokrytí množiny sémantických entit bezkontextovými gramatikami použitými pro jejich detekci.

Použité sémantické korpusy však obsahují abstraktní sémantické anotace. Referenční sémantické entity byly vygenerovány ze slovních přepisů použitím téže množiny bezkontextových gramatik, jaká byla následně použita pro detekci sémantických entit z rozpoznaných promluv. Při tomto přístupu není validováno pokrytí množiny použitých sémantických entit, neboť ty sémantické entity, které neodpovídají použité množině gramatik, nejsou zahrnuty mezi referenční. Dochází tak k nadhodnocení poměru správně detekovaných příkladů (TPR). Při použití tohoto přístup je však stále možné vyhodnotit vliv systému automatického rozpoznávání řeči na detekci sémantických entit. Nás pak bude zajímat především vliv použití slovních mřížek pro detekci sémantických entit v porovnání s použitím pouze první nejlepší hypotézy. Nadhodnocení veličiny (TPR) je systematickou chybou a ovlivní pouze škálování svislé osy, nikoli tvar ROC křivky. Pokud všechny experimenty nad stejnými daty sdílí i množinu bezkontextových gramatik, pak i různé ROC křivky vygenerované při použití stejných referenčních dat jsou vzájemně porovnatelné.

Pro porovnání různých modifikovaných ROC křivek je možné použít hodnotu získanou jako velikost plochy pod modifikovanou ROC křivkou – AUC (Area Under the Curve). V této práci budeme tuto plochu počítat z hodnot TPR pro FPR_{norm} z intervalu $< 0; 1 >$:

$$AUC = \int_0^1 TPR(FPR_{norm}) dFPR_{norm} \tag{9.27}$$

Hodnotu AUC lze interpretovat jako střední počet správně detekovaných sémantických entit při nejvýše jednom falešném poplachu na jednu promluvu.

9.4 Použité modely

Shrňme nyní vlastnosti akustických a jazykových modelů použitých při experimentálním ověření. Tyto modely byly použity v systému automatického rozpoznávání řeči pro rozpoznání promluv ze sémantických korpusů HHTT a TIA. Navíc jsou uvedeny dosahované přesnosti rozpoznávání řeči na slovní a fonémové úrovni stejně jako vybrané vlastnosti použitých mřížek.

9.4.1 Akustické modely

Pro parametrizaci nahrávek byla využita perceptivní lineární prediktivní analýza (PLP, [34]) se 12 koeficienty a delta- a delta-delta- koeficienty. Dále byla použita on-line verze kepstrální normalizace na střední hodnotu (cepstral mean normalization, CMN) počítaná

na základě výsledku detektoru řeč/neřeč (voice activity detector, VAD) z průběžného okénka o velikosti 1 sekundy.

Jako akustické modely byly použity standardní třístavové levopravé HMM modely trifónů s 2000 stavy. Výstupní hustoty pravděpodobností byly modelovány jako směs Gaussovských rozdělení (Gaussian Mixture Model, GMM) se 16 složkami na stav. Modely byly trénovány pomocí HTK [142], nejprve byl natrénován monofónový model s jednou Gaussovskou složkou na stav, který byl následně převeden na trifónový, bylo provedeno zarovnání a postupné přidávání složek, přičemž po přidání jedné složky následovalo šest reestimací.

Pro rozpoznání korpusu HHTT byl zvolen postup, kdy byl akustický model po počátečním natrénování monofónového jednosložkového modelu přetrénován z trénovací sady HHTT, neboť je nahrán prostřednictvím analogových telefonních linek a univerzální telefonní akustický model byl pro tato akustická data nevyhovující. Navíc byl použit akustický model zvlášť pro data odpovídající operátorovi linky a zvlášť zákazníkovi linky.

Pro korpus TIA byl použit univerzální telefonní akustický model TEL03 natrénovaný z obecných akustických dat v následujícím složení – proprietární telefonní korpus spontánní promluvy dvou převážně blízkých lidí (Bezplatné hovory, použito ≈200 hodin), telefonní řečový korpus SpeechDat-East[1] (použito ≈20 hodin), telefonní řečový korpus Siemens[2] (použito ≈10 hodin) a čtený telefonní korpus analogové linky (2000 řečníků, z nichž každý namluvil 40 vět, použito ≈34 hodin).

9.4.2 Slovní jazykové modely

Pro rozpoznání korpusu HHTT dostupná databáze všech vlakových zastávek a stanic umožnila vytvoření trigramového jazykového modelu s třídami, kde jednotlivé třídy reprezentovaly stanice v různých gramatických pádech. Pro automatické vyskloňování názvů stanic byla použita databáze vyskloňovaných tvarů jmen všech obcí v ČR spolu s pravidly pro skloňování víceslovných názvů stanic (např. *Praha hlavní nádraží*). Navíc samotný korpus HHTT obsahuje množství dat anotovaných na slovní úrovni, která ale nejsou použita v experimentech s modely porozumění, neboť například obsahují více než jeden dialogový akt nebo nejsou vůbec anotovány sémantickou informací. Proto se počet slov a vět použitých pro trénování jazykového modelu pro HHTT liší od tabulky 9.3, kde je zachycena pouze ta část korpusu, která je použita pro experimenty s modely porozumění. Přitom bylo dbáno na to, aby jako trénovací data použitá pro získání jazykového modelu nebyla použita data obsažená v development nebo testovací sadě určené pro experimenty s modely porozumění. Při trénování jazykových modelů opět byly vytvořeny zvláštní modely pro data odpovídající operátorovi a pro data odpovídající zákazníkovi telefonní linky.

Trigramový jazykový model pro úlohu TIA byl získán z trénovacích dat totožných s daty popsanými v tabulce 9.8.

Pro jazykové modelování byla použita sada nástrojů SRI Language Modeling Toolkit (SRILM) [107]. Jazykové modely byly estimovány jako trigramové jazykové modely s Witten-Bellovým vyhlazováním [106] a ústupovým vyhlazovacím schématem. Vlastnosti jazykových modelů jsou shrnuty v tabulce 9.9. Pro porovnání jazykové složitosti jednotlivých částí korpusů jsou uvedeny hodnoty perplexity (PPL) [143].

[1] http://www.fee.vutbr.cz/SPEECHDAT-E/sample/czech.html
[2] http://catalog.elra.info/product_info.php?products_id=562

	HHTT		TIA
	operátor	*zákazník*	
Počet vět	30802	27862	4166
Počet tokenů	191360	210945	33562
Velikost slovníku	11282	12638	2655
PPL (devel)	66,9	67,0	36,1
PPL (test)	50,5	62,4	38,0

Tabulka 9.9: Vlastnosti jazykových modelů použitých pro rozpoznávání korpusů.

9.4.3 Fonémové jazykové modely

Jedním z cílů této disertační práce je umožnit porozumění mluvené řeči bez nutnosti vytváření jazykového modelu dané úlohy, poněvadž pro získání robustního slovního jazykového modelu je nutné ručně přepsat dostatečné množství trénovacích dat. Toto množství dat závisí na složitosti cílové domény a zpravidla je nutné přepsat několik desítek hodin řeči. Možnost získání trénovacích dat pro jazykový model z jiných zdrojů je velmi omezená, neboť v dialozích se velmi často objevuje spontánní mluva, která v běžných korpusech obsahujících často psaný nebo čtený projev, není obsažena. Malé množství existujících korpusů obsahujících spontánní řeč zase nevyhovuje z hlediska jazykové podobnosti s danou dialogovou doménou.

Proto jsou zkoumány možnosti porozumění řeči pomocí rozpoznávání fonémů. Použitý fonémový rozpoznávač řeči stále používá akustický a jazykový model, nicméně jazykový model je nyní na úrovni jednotlivých fonémů. Vzhledem k tomu, že v akustických modelech je modelováno 40 fonémů a 9 neřečových událostí, je množství možných n-gramů, které lze z těchto jednotek sestavit, řádově menší. Je proto možné použít n-gramové jazykové modely s delší historií (vyššími hodnotami řádu n). A protože množina možných fonémů je fixovaná, lze použít i adaptaci fonémových jazykových modelů pomocí učení bez učitele. Při použití adaptovaných fonémových jazykových modelů není nutné přepisovat trénovací data pro slovní jazykový model nebo pro fonémový jazykový model získaný ze zarovnaných slovních přepisů – je možné použít obecný fonémový jazykový model a pouze jej adaptovat na datech z vybrané domény.[3]

Pro experimenty s porozuměním řeči založeným na fonémových jazykových modelech byly pro úlohu HHTT a TIA vytvořeny fonémové jazykové modely. Na základě experimentů s rozpoznáváním založeným na fonémech byly použity 5-gramové fonémové jazykové modely, neboť tyto modely poskytovaly nejvyšší fonémovou přesnost rozpoznávání (*Acc*), přičemž byla zachována schopnost dekódovat promluvy v reálném čase.

Byly použity tři různé typy fonémových jazykových modelů – jazykový model trénovaný z fonémově zarovnané přepisu, obecný jazykový model z korpusu *Bezplatné hovory* (BH) a jazykový model z korpusu BH adaptovaný na cílovou doménu (více v následující kapitole 9.4.4). Korpus BH je korpus spontánní češtiny obsahující dialogy dvou osob, které se

[3]Poznamenejme, že přestože není nutné sbírat data pro trénování slovního jazykového modelu, stále jsou nutná sémanticky anotovaná data pro trénování modelu porozumění. Diskuze této problematiky je obsažena v závěru disertační práce (kapitola 11).

zpravidla znají, realizovaný prostřednictvím telefonu. Korpus byl pořízen pro komerční účely ve společnosti SpeechTech s.r.o.

Pro získání fonémově zarovnaného přepisu byla nejprve ke všem slovům ze slovníku slovního přepisu vygenerována fonetická transkripce, tj. množina možných výslovností – posloupností fonémů, které mohou danému slovu odpovídat. Následně byl použit rozpoznávač řeči v módu forced-alignment (kapitola 2.2, str. 8 pro získání časového zarovnání jednotlivých fonémů se vstupní promluvou. Kromě časového zarovnání je v tomto módu vybrána i vhodná výslovnostní varianta u těch slov, která mají více než jednu fonémovou realizaci. Následně byl z fonémové úrovně těchto zarovnaných přepisů natrénován 5-gramový jazykový model, opět byla použita sada nástrojů SRILM a Witten-Bellovo vyhlazování a ústupové vyhlazovací schéma. Tímto způsobem byly získány fonémové jazykové modely pro datové korpusy HHTT, TIA a BH.

9.4.4 Adaptace fonémových jazykových modelů

Poněvadž 5-gramové jazykové modely modelují již celé fragmenty slov a často historie o délce 4 fonémy postihne i více než jedno slovo, jsou tyto jazykové modely nutně doménově závislé. Dobře to ilustruje tabulka 9.10, kde obecný fonémový jazykový model *ph-bh* má perplexitu více než dvakrát vyšší než fonémový jazykový model *ph-fa* trénovaný ze zarovnaných dat a to pro obě úlohy – HHTT i TIA.

Proto byly provedeny i experimenty s adaptovanými fonémovými jazykovými modely. Tato adaptace byla uvažována jako učení bez učitele. Algoritmus adaptace byl následující:

1. Natrénování obecného fonémového jazykového modelu (např. z dat BH)
2. Rozpoznání adaptačních dat pomocí obecného fonémového jazykového modelu
3. Z výsledného hypotetického fonémového přepisu natrénování adaptovaného jazykového modelu
4. Použití adaptovaného jazykového modelu pro rozpoznávání

Při trénování adaptovaného jazykového modelu v kroku 3 lze obecně použít počty n-gramů získané buď z první nejlepší hypotézy nebo střední počty výskytů n-gramů získané z celé fonémové mřížky. V této práci byl použit první přístup trénování z první nejlepší hypotézy. Perplexity takto adaptovaného jazykového modelu pro úlohy HHTT a TIA jsou shrnuty v tabulce 9.10 (řádky označené *ph-ad*). Je vidět, že i s pomocí n-gramových četností získaných z první nejlepší hypotézy lze natrénovat adaptovaný fonémový jazykový model, který výrazně redukuje perplexitu původního obecného fonémového jazykového modelu (*ph-bh*). Perplexita adaptovaného fonémového jazykového modelu se tak blíží perplexitě fonémového jazykového modelu získaného z fonémově zarovnaných přepisů. Jako adaptační data byly v obou případech použity trénovací množiny daných sémantických korpusů.

9.4.5 Pseudofonémové mřížky

Bylo rovněž zkoumáno použití tzv. *pseudofonémových mřížek*, které byly automaticky vygenerovány ze slovních mřížek. Pro jejich vytvoření byl použit výslovnostní slovník specifický pro danou úlohu (HHTT nebo TIA). Výslovnostní slovník každému slovu z rozpoznávacího slovníku $\mathcal{V}$ přiřazuje možné posloupnosti fonémů – výslovnostní varianty. Pro

Model	*HHTT*		*TIA*	
	devel	*test*	*devel*	*test*
ph-fa	7,02	6,59	8,08	8,26
ph-bh	15,63	15,76	20,82	20,36
ph-ad	8,25	7,79	11,54	11,65

Tabulka 9.10: Perplexity fonémových jazykových modelů, *ph-fa* – fonémový jazykový model získaný ze zarovnaných slovních přepisů, *ph-bh* – obecný jazykový model získaný z korpusu *ph-bh*, *ph-ad* – fonémový jazykový model *ph-bh* adaptovaný na cílovou úlohu (HHTT nebo TIA).

Korpus	*Model*	*2-gramů*	*3-gramů*	*4-gramů*	*5-gramů*
BH	*ph-bh*	2204	30812	173326	441758
HHTT	*ph-fa*	1761	16616	58019	105199
	ph-ad	1839	17609	63359	101995
TIA	*ph-fa*	1705	8199	13517	14141
	ph-ad	1746	9195	16923	14919

Tabulka 9.11: Počty unikátních n-gramů v použitých fonémových jazykových modelech. Počty n-gramů jsou ovlivněny standardním nastavením nástrojů SRILM, kde pro $n \geq 3$ jsou do jazykového modelu zahrnuty pouze ty n-gramy, které se vyskytly více jako jedenkrát.

snížení neurčitosti byla pro každé slovo ve výslovnostním slovníku ponechána pouze jediná výslovnostní varianta (ta s nejvyšším počtem fonémů).

Převod na pseudofonémové mřížky byl realizován pomocí operací s váženými konečnými transducery. Zjednodušený výslovnostní slovník byl převeden do konečného transduceru bez vah V. Vstupní abeceda tohoto transduceru je rozpoznávací slovník $\mathcal{V}$, výstupní abeceda pak fonémová sada tohoto výslovnostního slovníku. Každá cesta transducerem V je označena právě jedním vstupním symbolem – slovem z množiny $\mathcal{V}$. Transducer V je deterministický, neboť pro každé slovo z rozpoznávacího slovníku existuje právě jedna výslovnostní varianta. Následně je ze slovní mřížky U_w vytvořena pseudofonémová mřížka U_{map} pomocí tohoto algoritmu:

$$U_{map} = \min \det \Pi_2 \left(U_w \circ V^* \right) \tag{9.28}$$

Takto vygenerované pseudofonémové mřížky budeme v následujícím textu označovat pomocí identifikátoru *ph-map*.

9.4.6 Systém automatického rozpoznávání řeči

Systém automatického rozpoznávání řeči je implementován jako rozpoznávač spojité řeči s rozsáhlým slovníkem (Large Vocabulary Continuous Speech Recognizer, LVCSR). Tento rozpoznávač byl použit jako modul s akustickými a jazykovými modely popsanými výše. Tento dekodér byl použit například v pracech [116, 144] vzniklých na Katedře kybernetiky Západočeské univerzity v Plzni. Dekodér využívá strukturu trifónového lexikálního stromu, ve kterém jsou sloučeny cesty slov se shodnými počátečními trifóny. V průběhu dekódování jsou dynamicky vytvářeny historií podmíněné kopie lexikálního stromu (maximálně dvě slova historie), přičemž jsou použity techniky faktorizující jazykovou pravděpodobnost slov do struktury stromu. Tento proces zabírá významnou část dekódovacího času, proto je prováděn na základě optimalizované struktury uloženého jazykového modelu s využitím předpočítaných pravděpodobností a jejich cache. V průběhu dekódování jsou dynamicky vytvářeny trifóny pravého kontextu slov jen pro hypotézy, které nebyly prořezány na základě pravděpodobnostního odstupu od nejlepší hypotézy. Pro následné generování slovních grafů jsou v průběhu dekódování zaznamenávány cesty několika nejlepších tokenů. Celý proces dekódování je po částech paralelizován pro optimální využití systémů s více procesory.

Fonémový rozpoznávač pracuje se sítí všech stavů akustického modelu a jejich přechodů vytvořených na základě všech možných trifónů. Jazykový model je do procesu dekódování začleněn dynamicky, přičemž ale nejsou vyhodnocovány všechny n-gramové historie, ale pouze ty, jejichž hypotézy jsou aktivní v rozpoznávací síti. Díky velkému množství stavů akustického modelu a jejich přechodů tento přístup dosahuje jen nepatrně vyšší chybovosti rozpoznávání, přičemž ale umožňuje běh ve zlomcích reálného času minimálně do řádu n=5 jazykového modelu (toto omezení je dáno velikostí použitelné paměti). Pro následné generování fonémových mřížek jsou v průběhu dekódování zaznamenávány cesty několika nejlepších tokenů.

Pro běh systému automatického rozpoznávání řeči je nutné navíc určit hodnoty dvou parametrů – váhy jazykového modelu a penalty vložení slova (resp. fonému). Obě tyto metody hodnoty byly určeny pomocí vyčerpávajícího prohledávání mřížky hodnot a byla vybrána ta kombinace parametrů, která na development sadě daného korpusu dosáhla nejvyšší přesnosti Acc.

Uvěďme ještě způsob, jakým byly generovány mřížky ze systému automatického rozpoznávání řeči. Veškeré použité mřížky byly generovány nad logaritmický polookruhem. Použitý modul rozpoznávání řeči však neumožňuje přímé získání těchto mřížek, namísto toho jím vygenerované mřížky jsou ohodnoceny pomocí záporného logaritmu akustické pravděpodobnosti. Před jejich použitím je tedy bylo nutné doplnit o odpovídající jazykovou pravděpodobnost přiřazenou jazykovým modelem. Vstupem algoritmu je transducer A definovaný nad tropickým polookruhem. Tento transducer je ohodnocen vahami, které odpovídají záporným logaritmům akustické pravděpodobnosti. Symboly přiřazené přechodům v této mřížce odpovídají slovům, popřípadě fonémům promluvy. Dále je nutné použít stejný jazykový model, jako byl použit pro generování mřížky A, převedený do podoby transduceru L nad tropickým polookruhem. Přestože automaty A a L jsou de facto akceptory, budeme zde používat konvenci popsanou v kapitole 5.2, tj. že akceptor lze považovat za transducer, pro který pro každý přechod $e \in \mathcal{E}$ platí $i[e] = o[e]$.

Vstup:

- Transducer nad tropickým polookruhem A
- Jazykový model ve tvaru transduceru nad tropickým polookruhem L

Výstup:

- Transducer U nad logaritmickým polookruhem

Algoritmus generování mřížek:

1. Prořezej mřížku A prahem t_1, výstup označ $\text{prune}(A, t_1)$.
2. Proveď kompozici $A' = \text{prune}(A, t_1) \circ L$.
3. Opětovně prořezej mřížku prahem t_2: $U' = \text{prune}\,(A', t_2)$.
4. Změň polookruh transduceru U' z tropického na logaritmický.
5. Proveď optimalizaci $U'' = \min \det \text{rmeps}\, U'$.
6. Aplikuj algoritmus weight-pushing: $U = \text{push}(U'')$.

Algoritmus prořezávání mřížky [96] s prahem t zajistí, že ve výsledném automatu se nebudou nacházet cesty π, jejichž váhové ohodnocení $w[\pi]$ je $t\otimes$-krát větší, než váha nejlepší cesty $w[\hat{\pi}]$, tj. jsou odstraněny cesty π, pro které platí:

$$w[\pi] > t \otimes w[\hat{\pi}] \tag{9.29}$$

Algoritmus weight-pushing (v doslovném překladu stlačení vah) [8] nejprve upraví váhové ohodnocení vstupního transduceru tak, že $\oplus$-součet vah z libovolného stavu s výjimkou počátečního je roven $\bar{1}$. Tato změna váhového ohodnocení vstupního transduceru je prováděna takovým způsobem, aby výsledný automat byl ekvivalentní vstupnímu. Následně je aplikována normalizace vah přechodů z počátečního stavu automatu, aby i zde $\oplus$-součet těchto vah byl roven $\bar{1}$. Tímto krokem se již poruší ekvivalence vstupního a výstupního automatu. Tento krok odpovídá normalizaci pravděpodobnosti $\mathrm{P}(\mathsf{U})\mathrm{P}(\mathsf{O}|\mathsf{U})$ podělením pravděpodobností $\mathrm{P}(\mathsf{O})$ za účelem získání aposteriorní pravděpodobnosti $\mathrm{P}(\mathsf{U}|\mathsf{O})$ podle rovnice (2.2).

Transducer U nad logaritmickým polookruhem získaný tímto postupem již definuje pravděpodobnostní rozložení nad množinou cest tímto transducerem, platí:

$$\sum_{\pi \in \mathcal{P}_U(I,\mathcal{F})} w[\pi] = \bar{1} \tag{9.30}$$

Váha libovolné cesty $u \in U$ pak odpovídá zápornému logaritmu aposteriorní pravděpodobnosti $\mathrm{P}(\mathsf{U} = u|\mathsf{O})$.

U výše uvedeného postupu byly použity prahy $t_1 = 50$ a $t_2 = 5$. Přestože nastavováním těchto prahů lze velmi ovlivnit strukturu a především složitost vygenerovaných mřížek, experimenty neprokázaly žádné zvýšení konceptové přesnosti modelu porozumění při použití mřížek prořezaných za použití vyšších hodnot prahů t_1 a t_2. Proto byla vygenerována pouze jediná sada mřížek pro daný jazykový model a zvolený korpus.

Vyhodnocení přesnosti rozpoznávání je možné najít v tabulce 9.12. Přesnost rozpoznávání je vyhodnocena pomoc slovní/fonémové přesnosti *Acc*. V tabulce je rovněž uvedena

míra oracle accuracy (hodnoty v závorkách). Tato míra vyjadřuje slovní/fonémovou přesnost *Acc* vyhodnocenou nad hypotézou, která minimalizuje Levenshteinovu vzdálenost k referenčnímu přepisu (bez ohledu na váhu přiřazenou hypotéze). Oracle accuracy jistým způsobem vyjadřuje o kolik více informace je uloženo v mřížce v porovnání s první nejlepší hypotézou.

Na závěr poznamenejme, že novější verze použitého modulu automatického rozpoznávání řeči již podporuje přímo generování přechodových pravděpodobností mezi jednotlivými stavy mřížky. Není tedy nutné provádět kompozici mřížky a jazykového modelu. Ve výše zmíněném algoritmu pak výpočet začíná až krokem 3. Tímto dochází k velmi významnému urychlení generování mřížek vhodných pro další zpracování pomocí popsaných metod. Bohužel, tato verze rozpoznávače byla vyvinuta až po provedení naprosté většiny experimentů a tudíž je v experimentálním vyhodnocení použit výše uvedený postup.

9.5 Shrnutí

V této kapitole byly popsány dva různé sémanticky anotované korpusy HHTT a TIA. Rovněž byl uveden i způsob vyhodnocení jednotlivých částí popsaného modelu. Hierarchický diskriminativní model bude vyhodnocován pomocí *konceptové přesnosti cAcc*, detekce sémantických entit pak bude validována za použití modifikovaných ROC křivek a míry *AUC* vyjadřující plochu pod touto křivkou. V poslední části pak byly popsány použité akustické a jazykové modely spolu s rozpoznávacím dekodérem a dalšími postupy nutnými pro získání sad mřížek použitých v experimentální části disertační práce.

Model		*HHTT*		*TIA*	
		devel	*test*	*devel*	*test*
Slovní		70,5 (82,7)	72,9 (84,8)	72,4 (78,6)	62,5 (71,1)
Fonémový	*ph-fa*	74,7 (79,5)	76,2 (81,1)	77,2 (81,7)	68,5 (74,2)
	ph-bh	65,5 (75,4)	67,6 (76,7)	58,6 (70,8)	51,4 (64,0)
	ph-ad	72,5 (78,5)	74,4 (80,0)	69,4 (76,4)	61,8 (69,6)
Pseudofon.	*ph-map*	74,8 (79,3)	76,1 (80,3)	79,3 (82,0)	72,5 (76,7)

Tabulka 9.12: Přesnost rozpoznávání (*Acc* v procentech) pro slovní a fonémové jazykové modely použité v experimentální části. Hodnoty v závorkách odpovídají míře oracle accuracy, která vyjadřuje přesnost (*Acc*) té hypotézy z mřížky, která je nejbližší referenci.

	Slovní mřížky	*ph-map*	*ph-fa*	*ph-ad*	*ph-bh*
Počet stavů	7,47±7,75 (5,00)	30,35±27,92 (22,00)	24,07±22,86 (17,00)	26,49±26,35 (18,00)	43,68±61,07 (24,00)
Počet přechodů	11,40±19,61 (6,00)	30,86±30,36 (22,00)	26,66±29,16 (18,00)	30,25±34,50 (20,00)	56,68±88,97 (27,00)
Počet cest	64,21±1038,21 (2,00)	4,65±25,30 (2,00)	70,20±790,57 (2,00)	88,57±769,67 (3,00)	342,07±1962,81 (8,00)
Počet přechodů ze stavu	1,53±1,46 (1,00)	1,02±0,34 (1,00)	1,11±0,52 (1,00)	1,14±0,56 (1,00)	1,30±0,72 (1,00)
Maximální počet přechodů ze stavu	2,69±2,22 (2,00)	1,71±0,93 (1,00)	2,06±1,09 (2,00)	2,23±1,17 (2,00)	2,89±1,37 (3,00)
Počet symbolů nejlepší cesty	4,06±3,39 (3,00)	23,07±18,20 (18,00)	17,94±14,22 (15,00)	18,11±14,28 (15,00)	17,82±14,03 (15,00)
Pst. nejlepší cesty	0,87±0,20 (0,99)	0,87±0,20 (0,99)	0,77±0,26 (0,89)	0,72±0,29 (0,81)	0,58±0,32 (0,57)

Tabulka 9.13: Vlastnosti použitých rozpoznaných mřížek pro korpus HHTT

	Slovní mřížky	*ph-map*	*ph-fa*	*ph-ad*	*ph-bh*
Počet stavů	8,26±8,33 (5,00)	41,86±45,40 (25,00)	45,96±44,96 (31,00)	55,08±57,34 (36,00)	142,68±191,88 (69,00)
Počet přechodů	9,81±13,71 (5,00)	42,62±48,17 (25,00)	54,85±55,96 (37,00)	68,85±74,75 (45,00)	199,92±279,23 (94,00)
Počet cest	69,16±1440,83 (2,00)	14,08±185,70 (1,00)	454,98±2068,69 (12,00)	655,47±2524,83 (16,00)	1114,72±3800,96 (14,00)
Počet přechodů ze stavu	1,19±0,94 (1,00)	1,02±0,30 (1,00)	1,19±0,59 (1,00)	1,25±0,63 (1,00)	1,40±0,75 (1,00)
Maximální počet přechodů ze stavu	1,99±1,54 (2,00)	1,65±1,05 (1,00)	3,06±1,26 (3,00)	3,24±1,23 (3,00)	4,03±1,43 (4,00)
Počet symbolů nejlepší cesty	6,00±6,23 (4,00)	34,84±36,88 (21,00)	32,31±31,51 (21,00)	31,88±30,81 (20,00)	31,14±30,60 (20,00)
Pst. nejlepší cesty	0,80±0,30 (0,99)	0,80±0,29 (0,99)	0,53±0,30 (0,50)	0,47±0,30 (0,43)	0,38±0,30 (0,30)

Tabulka 9.14: Vlastnosti použitých rozpoznaných mřížek pro korpus TIA

Vysvětlivky k tabulkám 9.13 a 9.14: V každé z buněk tabulek jsou zapsány tři hodnoty ve tvaru 8,26±8,33 (5,00). Tento zápis vyjadřuje, že průměr dané veličiny je 8,26 s rozptylem 8,33 a mediánem 5,00.

Kapitola 10

Experimentální ověření

V této kapitole bude popsáno experimentální ověření popsaných metod nad sémantickými korpusy HHTT a TIA. Nejprve uveďme popis postupu realizovaného při tomto ověření.

V kapitole 10.1 bude nejprve vyhodnocen přínos inovativního výpočtu racionální jádrové funkce pomocí optimalizovaného váženého transduceru R reprezentujícího prvky celé trénovací množiny. Tato metoda bude porovnána se dvěma základními metodami pro výpočet racionální jádrové funkce mezi dvěma automaty. S ohledem na nasazení v hlasových dialogových systémech nás bude zajímat především rychlost vyhodnocení racionální jádrové funkce.

V následující kapitole 10.2 je popsán postup, jakým byly určeny parametry modelu HDM. Za účelem nastavení těchto parametrů byla použita trénovací sada korpusu HHTT, která byla rozdělena v poměru 72:8:20 na sady train_t, train_d a train_e. Ty byly následně použity jako trénovací, development a testovací sada pro nastavení parametrů modelu HDM. Tento krok byl proveden, aby původní development a testovací sada korpusu HHTT nebyla při nastavování parametrů použita, čímž se zabrání vychýlení (přetrénování) nastavovaných parametrů na konkrétní data nebo dokonce rozdělení dat. V rámci nastavování parametrů modelu HDM jsou postupně určovány parametry vstupní vrstvy (kapitola 10.2.1), skryté vrstvy (kapitola 10.2.2) a výstupní vrstvy (kapitola 10.2.3).

Po nastavení parametrů s využitím výše popsaných sad train_t, train_d a train_e byl natrénován HDM model nad celým korpusem HHTT a nad korpusem TIA. Tyto výsledky jsou shrnuty v kapitole 10.3. Celý postup dokládá, že dosahovaná přesnost modelu HDM není příliš citlivá na přesné nastavení parametrů modelu pro daná data, především co se týče jejich domény a množství.

V kapitole 10.4 následuje vyhodnocení modelu pro detekci sémantických entit. Detailněji jsou popsány vlastnosti použitých gramatik a vyhodnocení pomocí modifikovaných ROC křivek a hodnot metriky AUC.

Model detekce sémantických entit a model zarovnání následně umožňují kombinaci s hierarchickým diskriminativním modelem pomocí pravděpodobnostních vztahů popsaných v kapitole 6. Tento výsledný diskriminativní model pro porozumění řeči je vyhodnocen z pohledu konceptové přesnosti opět na korpusy HHTT a TIA, výsledky tohoto vyhodnocení jsou popsány v kapitole 10.5.

Poslední kapitola 10.6 prezentuje křivky učení modelu HDM. Tyto křivky vyjadřují závislost konceptové přesnosti dosahované modelem HDM při použití různého množství trénovacích dat.

Vzhledem k tomu, že model HDM byl navržen tak, aby jej bylo možné trénovat a následně provozovat nad velmi různorodými strukturami vstupních dat, setkáme se s následujícím označením:

- *Slovní přepis* – jedná se o přepis přiřazený dané promluvě člověkem anotátorem. Přepis není ovlivněn chybami automatického rozpoznávání řeči, data mají podobu řetězce slov.
- *Slovní 1. hypotéza* – nejlepší slovní hypotéza ze systému automatického rozpoznávání řeči, data opět mají podobu řetězce slov.
- *Slovní mřížka* – výstupní mřížka ze systému automatického rozpoznávání řeči, data jsou ve formě acyklického váženého konečného akceptoru.
- *Fonémová 1. hypotéza* – nejlepší fonémová hypotéza z fonémového rozpoznávače, data mají podobu řetězce fonémů.
- *Fonémová mřížka* – výstupní mřížka z fonémového rozpoznávače, data jsou ve formě acyklického váženého konečného akceptoru.
- *Pseudofonémová mřížka* – mřížka vzniklá převedením slovních mřížek na fonémové pomocí postupu popsaného v kapitole 9.4.5, data jsou ve formě acyklického váženého konečného akceptoru.

U dat, která reprezentují fonémy, je vždy ještě uveden fonémový jazykový model použitý k jejich vygenerování. Jsou použity zkratky *ph-fa*, *ph-bh*, *ph-ad* a *ph-map* popsané v kapitole 9.4.4.

10.1 Výpočet racionální jádrové funkce

Výpočet racionální jádrové funkce pomocí vážených konečných automatů má perspektivní využití v hlasových dialogových systémech. Pro vyhodnocení porovnáme algoritmus výpočtu pomocí algoritmu prezentovaného v kapitole 7.1 (dále nazývaný *HDM*) s *naivním výpočtem* (popis níže) a s klasickým algoritmem popsaným v sekci 5.3.4 [10], tj. výpočtem hodnot racionální jádrové funkce *po párech*.

Naivní výpočet nejprve explicitně vyčíslí příznakový vektor odpovídající vstupní mřížce a následně spočítá skalární součin tohoto příznakového vektoru vzhledem k příznakovým vektorům pro všechny mřížky v trénovací množině. Vzhledem k velkému počtu možných příznaků, z nichž však pouze malá část je nenulová, je použita řídká reprezentace tohoto příznakového vektoru pomocí asociativního pole. Příznakový vektor je vypočítán jako střední počet výskytů daného n-gramu ve vstupní mřížce podle rovnice (5.74). Předpokládá se, že $\oplus$-součet vah všech cest v dané mřížce je $\bar{1}$. Algoritmus výpočtu iteruje přes všechny cesty danou mřížkou a sčítá četnosti výskytů každého n-gramu v dané cestě vážené pravděpodobností cesty. Tento algoritmus má obecně exponenciální časovou složitost, neboť počet cest akceptorem roste exponenciálně s počtem jeho stavů a přechodů.

Porovnejme nyní časy výpočtu racionální jádrové funkce pro různé velikosti trénovací sady $|T|$. Abychom v experimentech nebyli omezeni řádově nejvýše 7249 větami z korpusu HHTT (viz tabulka 9.3, součet vět trénovací, development a testovací sady), byla použita nadmnožina korpusu HHTT sestávající se i z vět, které nemají sémantickou anotaci, popřípadě obsahují více než jeden dialogový akt (více v kapitole 9.1). Takto bylo možné experimenty týkající se výpočetní náročnosti vstupní vrstvy provést nad množinou přibližně 56 000 mřížek. Výpočetní složitost byla testována pro dva extrémní případy:

1. *Slovní 1. hypotéza*, kde existuje vždy právě jedna cesta vstupním akceptorem U a složitost akceptoru je závislá pouze na počtu přechodů mezi stavy. Pro tato data byl použit transducer $T_{1,3}$ definující racionální jádrovou funkci, tj. jako příznaky byly uvažovány střední četnosti všech n-gramů o délce $n = 1$ až $n = 3$.

2. *Fonémová mřížka, ph-fa*, která je tvořena velkým počtem uzlů a přechodů, složitost akceptoru je závislá nejen na jejich počtu, ale i na způsobu jejich propojení, tj. na počtu cest akceptorem U. Počet stavů na jednotku času je také vyšší oproti slovním mřížkám, přibližně třikrát až šestkrát (více v tabulkách 9.13 a 9.14). Pro fonémové mřížky byl použit transducer $T_{1,5}$ a tudíž jako příznaky byly použity střední četnosti všech n-gramů o délce $n = 1$ až $n = 5$.

Pro porovnání časů výpočtu byly použity tzv. krabicové diagramy (angl. box plot), které zobrazují hlavní statistické charakteristiky datového souboru více prvků [145]. V těchto diagramech modrý obdélník znázorňuje svým rozsahem na Y-ose grafu 25% a 75% kvartily, červená úsečka pak medián, černé úsečky znázorňují rozsah dat, modré křížky pak odlehlé hodnoty (angl. outliers) [146]. Z celého rozsahu dostupných slovních hypotéz a fonémových mřížek (cca 56 000 promluv) bylo vždy vybráno 100 různých příkladů a tyto příklady byly po jednom předkládány k výpočtu racionální n-gramové jádrové funkce, přičemž velikost hypotetické trénovací množiny byla měněna v rozsahu několika řádů (1-20 000). Tento postup byl vždy pětkrát opakován, tak aby se množina vstupních příkladů a trénovací množina nepřekrývaly. Takto byly získány časy pro 500 různých vstupních vektorů. Z důvodu asymptoticky exponenciální časové složitosti naivního algoritmu byl maximální uvažovaný počet různých hypotéz pro fonémovou mřížku omezen hodnotou 100 000.

Pro danou trénovací množinu $\mathcal{T}$ byly nejprve předpočítány následující struktury tak, aby doba jejich výpočtu následně nebyla zahrnuta do doby výpočtu hodnot jádrových funkcí $K(U, U_j),\ j = 1, 2, \ldots l$:

- Pro algoritmus použitý ve vstupní vrstvě modelu HDM byl předpočítán optimalizovaný transducer R.
- Pro naivní výpočet byl pro každou mřížku $U_j \in \mathcal{T}$ předpočítán příznakový vektor $\mathbf{x}(U_j)$ a uložen jako řídký vektor reprezentovaný asociativním polem.
- Pro výpočet po párech byl pro každou mřížku $U_j \in \mathcal{T}$ předpočítán transducer $\det(\mathrm{rmeps}(T^{-1} \circ U_j))$.

Obrázky 10.1, 10.2 a 10.3 zobrazují závislost doby výpočtu jádrových funkcí na velikosti množiny $\mathcal{T}$ tvořené nejlepšími slovními hypotézami. Obdobně obrázky 10.6, 10.7 a 10.8 zobrazují závislost doby výpočtu jádrových funkcí na velikosti množiny $\mathcal{T}$ tvořené fonémovými mřížkami.

Ze třech vyhodnocovaných algoritmů má naivní výpočet nejvyšší rozptyl naměřených časů, neboť doba získání příznakového vektoru pro automat U je extrémně závislá na počtu cest automatem a rovněž na délkách těchto cest.

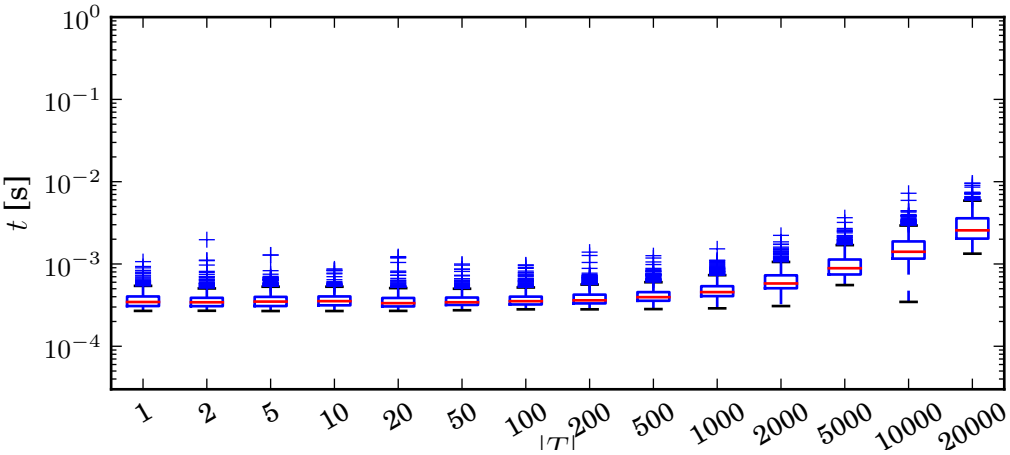
Algoritmus výpočtu po párech vykazuje lineární růst doby výpočtu s velikostí trénovací množiny a zároveň doba výpočtu vykazuje velmi malý rozptyl při konkrétní hodnotě $|\mathcal{T}|$.

Obrázky 10.4 a 10.9 ukazují závislosti mediánu doby výpočtu jádrových funkcí $K(U, U_j)$ na velikosti trénovací množiny $|\mathcal{T}|$. V grafech je možné vzájemně porovnat mediány doby výpočtu pro jednotlivé metody a pro různé velikosti trénovací množiny. Pro běžné velikosti trénovací množiny (řádově více než 10^3 prvků) je z prezentovaných metod výpočtu racionální jádrové funkce nejrychlejší metoda použitá ve vstupní vrstvě HDM.

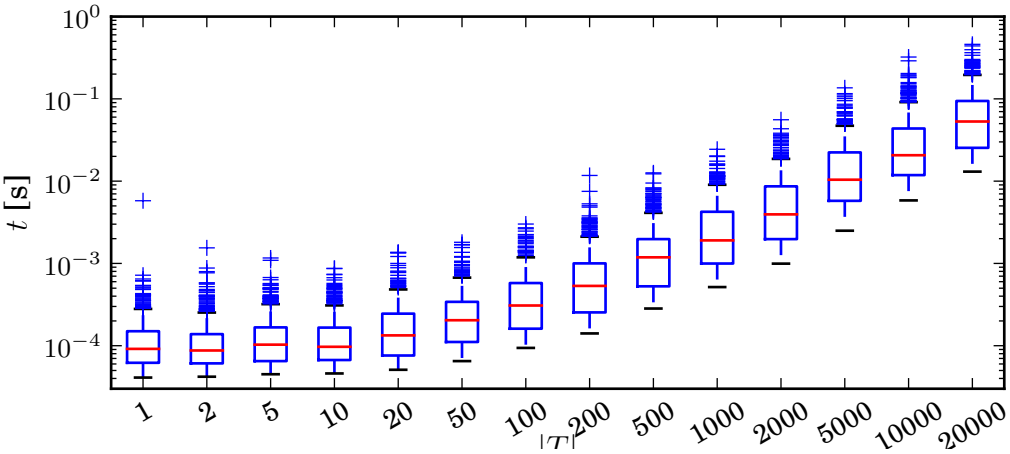
Konečně u algoritmu použitého ve vstupní vrstvě HDM doba výpočtu s velikostí trénovací množiny roste nejpomaleji – naplno se zde projevuje minimální deterministická reprezentace transduceru R použitého pro výpočet racionální jádrové funkce. Vyšší rozptyl doby výpočtu pro mřížku U je způsoben rozdílnou velikostí transduceru $L \circ R$, ve které se projevuje počet shodných n-gramů mezi U a prvky trénovací množiny U_j. Shodují-li n-gramy z mřížky U s n-gramy velkého množství prvků trénovací množiny, pak bude i počet přechodů $L \circ R$ vysoký a doba získání $K(U, U_j)$ se úměrně prodlouží.

Obrázky 10.5 a 10.10 zobrazují závislosti doby výpočtu jádrových funkcí $K(U, U_j)$ na počtu přechodů $|\mathcal{E}|$ automatu U při velikosti trénovací množiny $|\mathcal{T}| = 20\,000$. Doba výpočtu po párech je téměř nezávislá na délce slovní hypotézy. Doba naivního výpočtu stejně jako algoritmu HDM je téměř lineární s počtem slov hypotézy, nicméně směrnice myšlené lineární aproximace v případě HDM má výrazně menší sklon, neboť se zde projevuje výpočet jádrových funkcí globálně nad celou trénovací množnou.

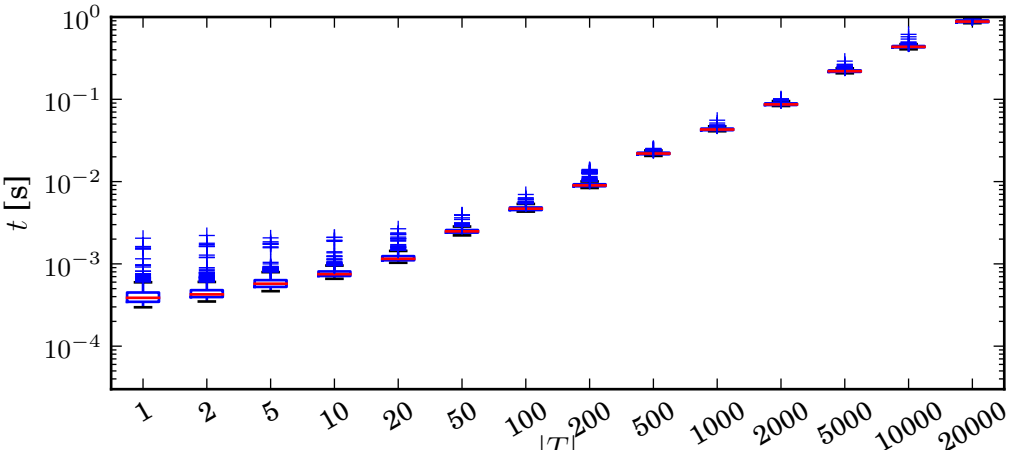
Z pohledu výpočetní složitosti jsou dále zajímavé vlastnosti transduceru R pro danou trénovací množinu. Tyto vlastnosti jsou uvedeny na straně 140 v tabulce 10.8, neboť pro jejich určení je nutné nejprve vhodně zvolit parametry n a m racionální jádrové funkce definované pomocí transduceru $T_{n,m}$.



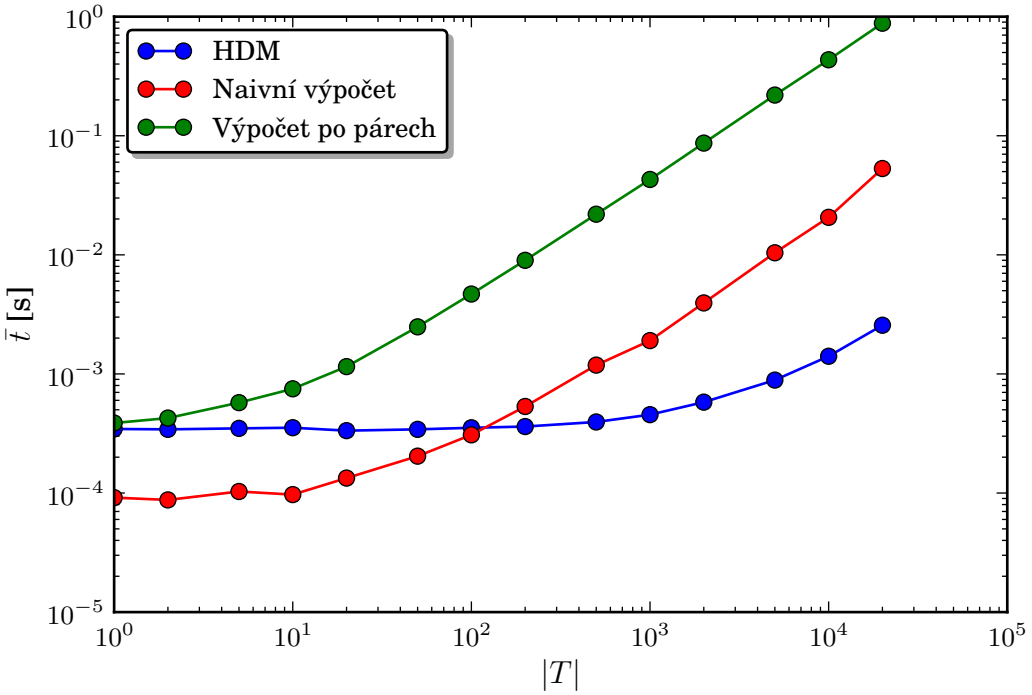
Obrázek 10.1: Časová náročnost algoritmu použitého ve vstupní vrstvě HDM, slovní 1. hypotéza.



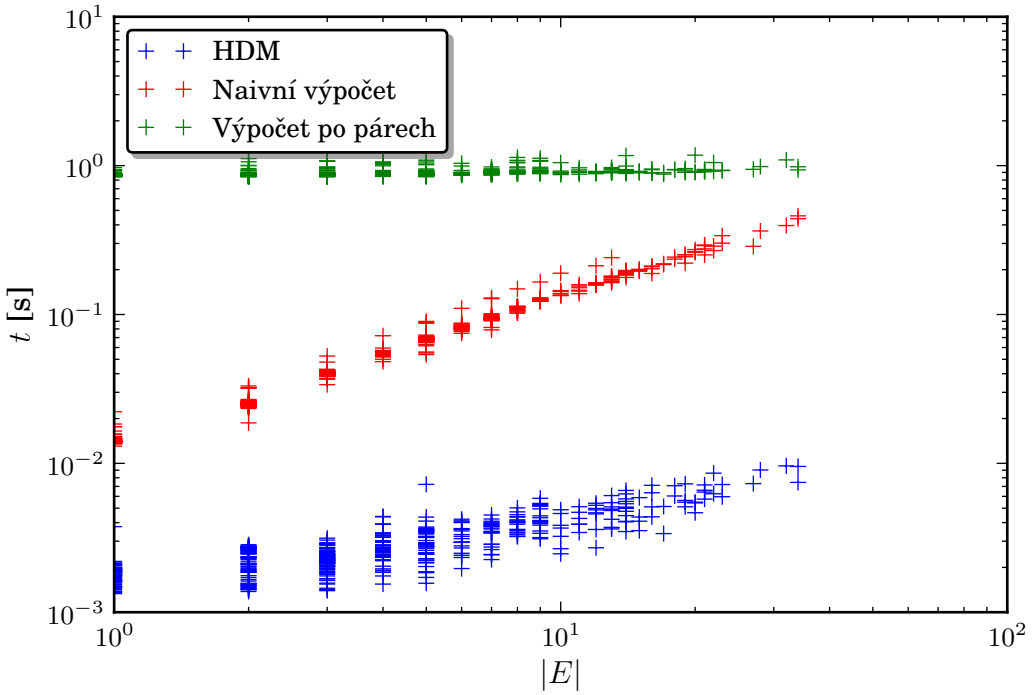
Obrázek 10.2: Časová náročnost naivního výpočtu, slovní 1. hypotéza.



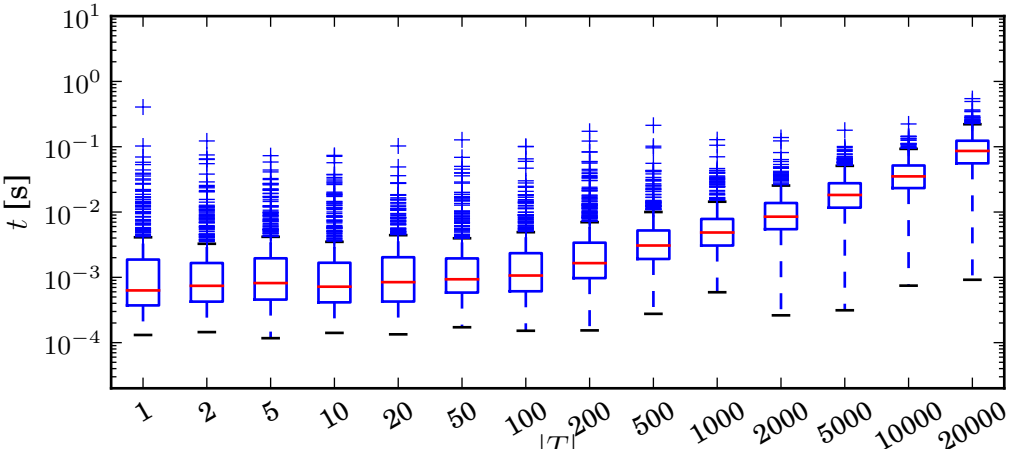
Obrázek 10.3: Časová náročnost výpočtu po párech, slovní 1. hypotéza.



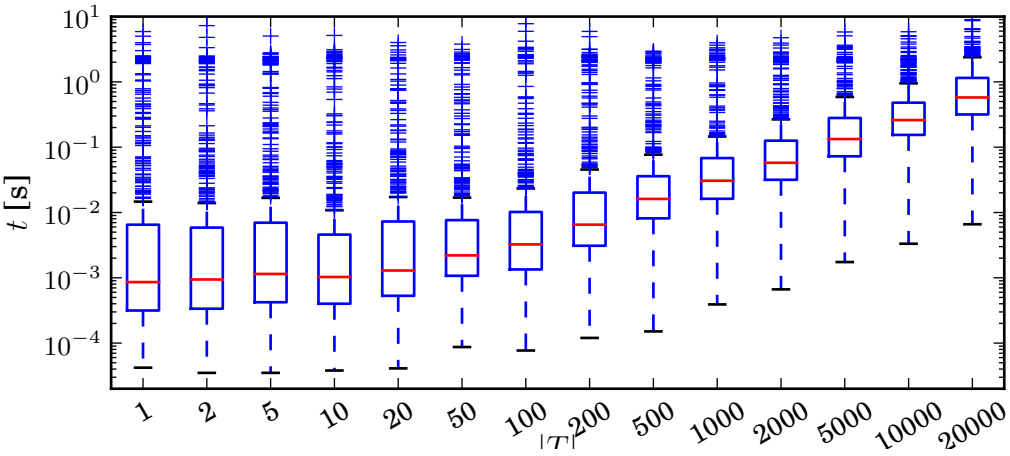
Obrázek 10.4: Závislost mediánu doby výpočtu racionální jádrové funkce v závislosti na rozsahu trénovací množiny $\mathcal{T}$, slovní 1. hypotéza.



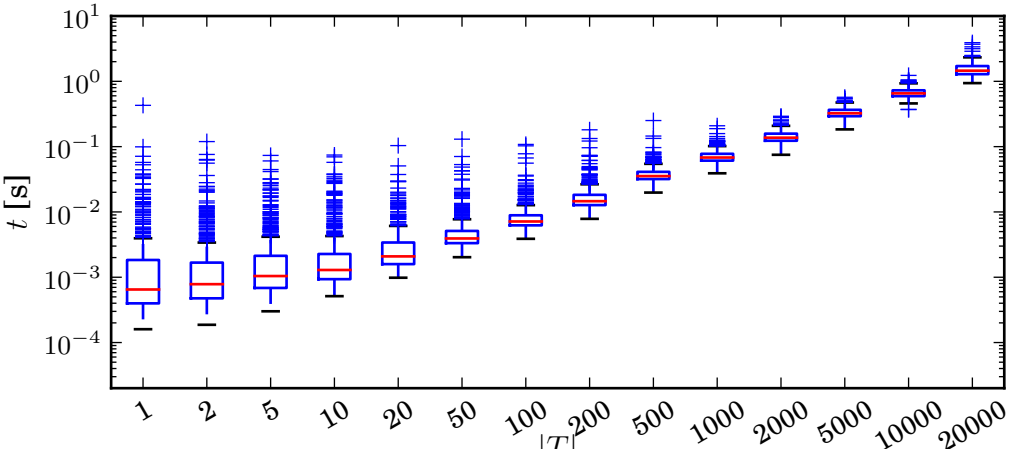
Obrázek 10.5: Závislost doby výpočtu racionální jádrové funkce na počtu přechodů vstupní mříky U, velikost trénovací sady fixována na hodnotu $|\mathcal{T}| = 20\,000$, slovní 1. hypotéza.



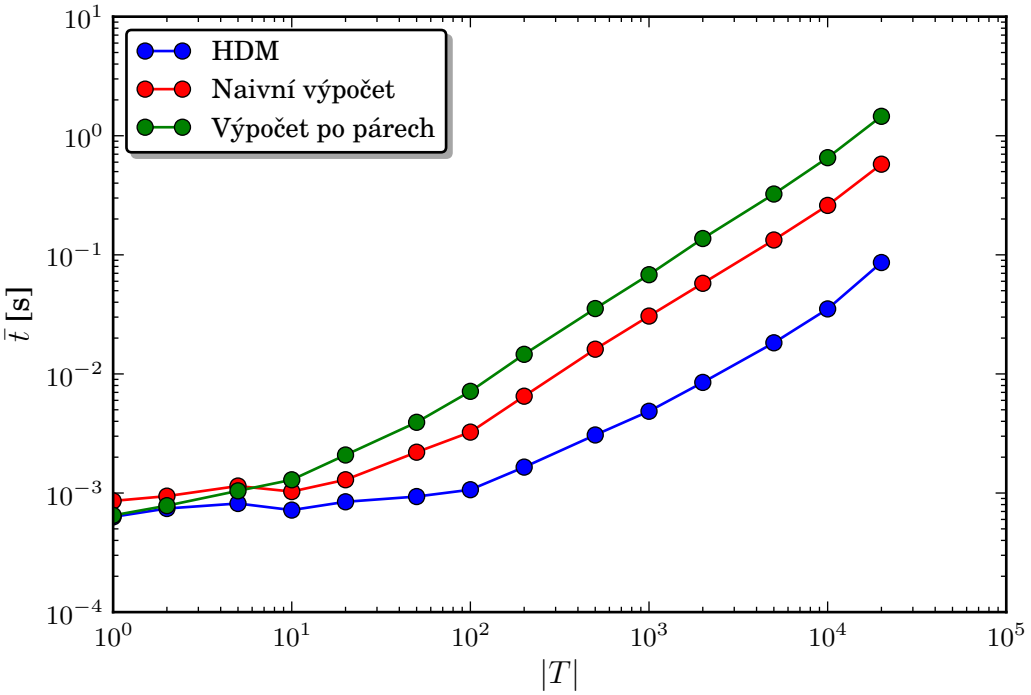
Obrázek 10.6: Časová náročnost algoritmu použitého ve vstupní vrstvě HDM, fonémová mřížka.



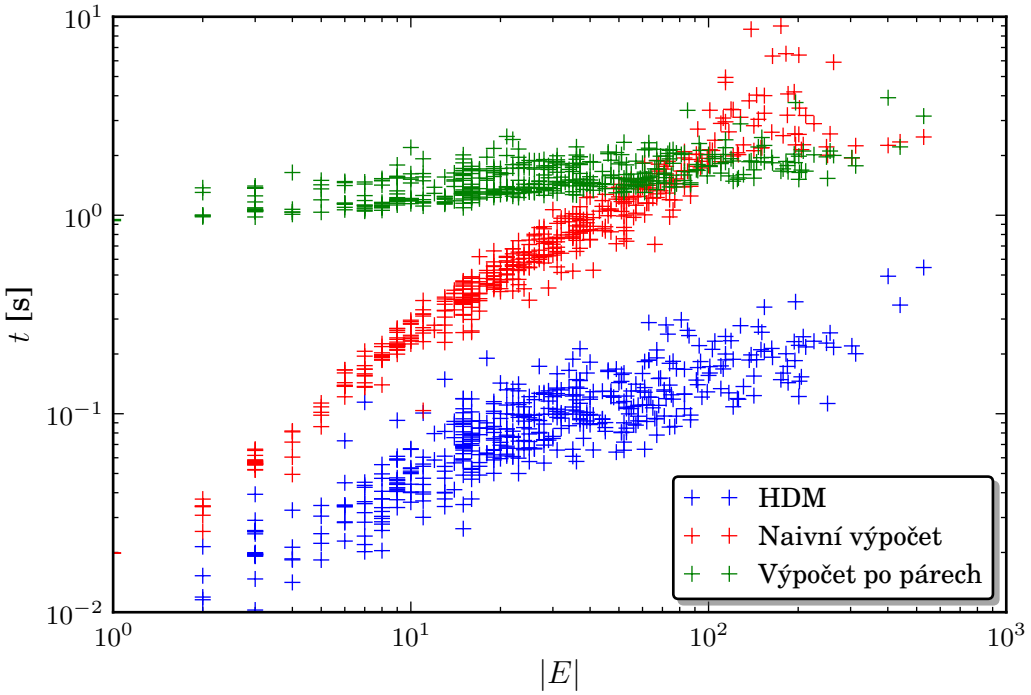
Obrázek 10.7: Časová náročnost naivního výpočtu, fonémová mřížka.



Obrázek 10.8: Časová náročnost výpočtu po párech, fonémová mřížka.



Obrázek 10.9: Závislost mediánu doby výpočtu racionální jádrové funkce v závislosti na rozsahu trénovací množiny $\mathcal{T}$, fonémová mřížka.



Obrázek 10.10: Závislost doby výpočtu racionální jádrové funkce na počtu přechodů vstupní mříky U, velikost trénovací sady fixována na hodnotu $|\mathcal{T}| = 20\,000$, fonémová mřížka.

10.2 Parametry HDM

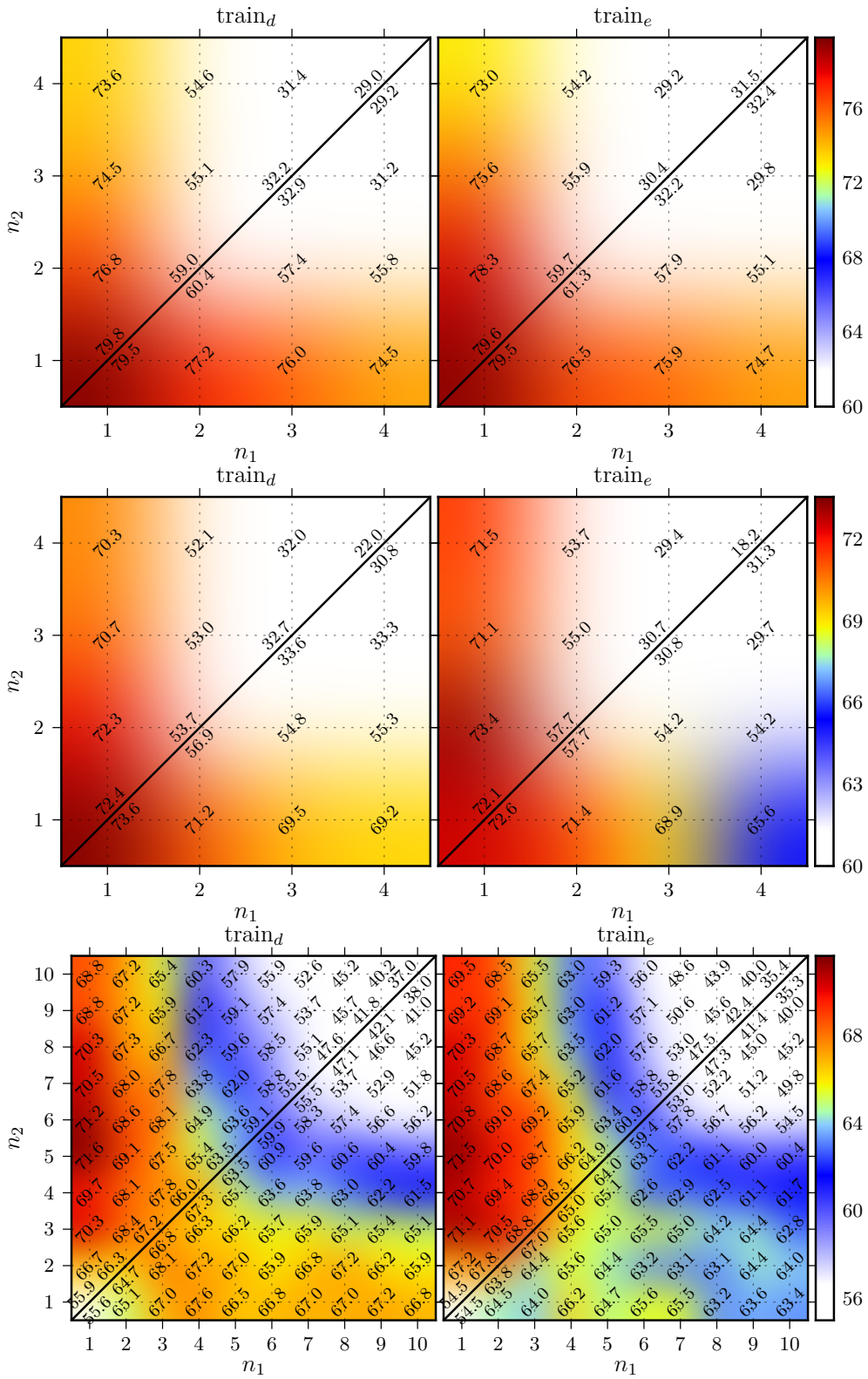
HDM má řadu parametrů a metaparametrů, které budou v následujícím textu souhrně označeny pojmem parametry modelu:

- *Vstupní vrstva* popsaná v kapitole 7.1 je parametrizována použitou racionální jádrovou funkcí. Omezíme se pouze na n-gramové racionální jádrové funkce s minimálním řádem n a maximálním řádem m definované pomocí transduceru $T_{n,m}$ z rovnice (5.81). Volitelně je možné použít normalizaci racionální jádrové funkce podle rovnice (5.44). Nastavení těchto parametrů je věnována kapitola 10.2.1.
- *Skrytá vrstva* je blíže popsána v kapitole 7.2. Při nastavování parametrů modelu HDM je nejprve zodpovězena otázka samotného přínosu skryté vrstvy. Je porovnávána konceptová přesnost HDM modelu sestaveného pouze ze vstupní a výstupní vrstvy s HDM modelem používajícím všechny tři vrstvy. Dále je určen práh N determinující množinu sémantických n-tic $|\mathcal{S}_N|$ podle rovnice (7.8). Nakonec je vybrán způsob nastavování regularizačního parametru $C^{\mathbf{t}}$ pro každý SVM klasifikátor predikující sémantickou n-tici $\mathbf{t}$. Popis experimentů týkajících se skryté vrstvy je uveden v kapitole 10.2.2.
- *Výstupní vrstva* formálně popsaná v kapitole 7.3 je parametrizována jediným parametrem M určujícím trénovací množinu $\mathcal{T}_A$ pro klasifikátor g_A predikující potomky konceptu A ve výstupním sémantickém stromu. Množina $\mathcal{T}_A$ je definována v rovnici (7.27) a experimenty související s výstupní vrstvou HDM modelu jsou popsány v kapitole 10.2.3.

Pro nastavení parametrů byla trénovací množina korpusu HHTT (5240 promluv) rozdělena na menší trénovací množinu train_t (3815 promluv), development sadu train_d (371 promluv) a testovací sadu train_e (1054 promluv). Pro odhad parametrů modelu HDM byly použity následující typy dat: *slovní přepis*, *slovní mřížka* a *fonémová mřížka ph-fa*. Nastavení pro ostatní typy dat (jako slovní 1. hypotéza nebo pseudofonémové mřížky) pak od nich bylo odvozeno. V rámci všech následujících experimentů byla optimalizována míra konceptové přesnosti (*cAcc*). Ve všech tabulkách a grafech je tato míra uváděna v procentech.

10.2.1 Vstupní vrstva

Vstupní vrstva počítá n-gramovou racionální jádrovou funkci mezi vstupní promluvou u (reprezentovanou mřížkou U) a promluvami z trénovací množiny u_i (odpovídající mřížky U_i). Tato racionální jádrová funkce je definována transducerem $T_{n,m}$ dle rovnice (5.81). Minimální a maximální řád n, resp. m je nutné určit experimentálně. Grafy na obrázku 10.11 vyjadřují závislost konceptové přesnosti na zvolených hodnotách parametrů n a m. Zároveň tyto obrázky zachycují i vliv normalizace racionální jádrové funkce dle rovnice (5.44). Pro libovolný průsečík celých čísel v uvedených grafech platí, že minimální řád n-gramové racionální jádrové funkce $n = \min\{n_1, n_2\}$, maximální řád $m = \max\{n_1, n_2\}$ a normalizace racionální jádrové funkce je použita v případě, že $n_2 < n_1$, tj. hodnoty nad diagonálou odpovídají normalizovaným jádrovým funkcím, pod diagonálou nenormalizovaným. Hodnoty *cAcc* jsou vyčísleny vždy jak pro množinu train_d, která byla použita



Obrázek 10.11: Hodnoty konceptové přesnosti pro různé n-gramové jádrové funkce nad slovním přepisem (nahoře), slovními mřížkami (uprostřed) a fonémovými mřížkami (dole).

jako development sada, tak pro množinu train_e použitou jako testovací sada. Přiřazení barev konkrétním hodnotám *cAcc* odpovídá barevné škále vpravo, hodnoty menší než dolní hranice barevné škály jsou zobrazeny bíle.

Na základě hodnot prezentovaných v grafech na obrázku 10.11 byla normalizace racionální jádrové funkce použita pro všechny typy vstupních dat. Její přínos při hodnotách $n \neq m$ je zřejmý především pro data pocházející z výstupu systému automatického rozpoznávání řeči. Pro případy, kdy $n = m$, je přínos normalizace jádrové funkce diskutabilní, nicméně lze použít další kriterium, kterým je doba trénování HDM modelu. Časy nutné pro natrénování HDM modelu jsou shrnuty v tabulce 10.1. Z tabulky vyplývá, že při použití normalizované jádrové funkce je doba trénování méně než poloviční než při použití nenormalizované jádrové funkce. Toto je způsobeno především vysokou variabilitou nenormalizovaných hodnot $K(U, U_k) \in [0; +\infty)$ oproti normalizovaným $\tilde{K}(U, U_k) \in [0; 1]$ což negativně ovlivňuje časy nutné pro konvergenci trénovacího algoritmu SVM. Tento jev se projevuje jak u skryté vrstvy, kde konvergenci ovlivňují přímo hodnoty jádrové funkce, tak i u výstupní vrstvy, neboť vysoká variabilita nenormalizovaných hodnot jádrové funkce se promítne i do vysoké variability vzdáleností k rozhodovacím nadrovinám $\mathcal{H}^{\mathbf{t}}$ a tudíž vysokou variabilitou složek $d_{\mathbf{t}}(U_k)$ vektoru $\mathbf{d}(U_k)$.

Při použití vstupu na úrovni slov se ukazuje, že nejlepších výsledků je nad danými daty dosahováno pro $n = 1$ a s relativně malými řády m. Pro referenční slovní přepis dokonce pro $n = m = 1$. Toto je dáno především malým počtem trénovacích promluv vzhledem k velikosti slovníku dané úlohy a tím pádem i vzhledem k počtu různých n-gramů vyšších řádů – pravděpodobnost výskytu shodných bigramů (pro slovní přepis) mezi mřížkou U a trénovacími mřížkami U_k je nízká, pro vyšší řady n-gramů pak mizivá.

Na fonémové úrovni se tento jev neprojevuje, neboť různých fonémů je řádově menší počet než různých slov, proto i počet různých n-gramů je mnohem menší a tím pádem počet shodných n-gramů vyšších řádů mezi dvěma mřížkami může pozitivně ovlivnit konceptovou přesnost výsledného modelu. Navíc se na fonémové úrovni pozitivně projevuje vliv normalizace jádrové funkce, kdy v případě použití $n = 1$ a $m = 5$ je rozdíl v konceptové přesnosti téměř pět procent při porovnání normalizované a nenormalizované racionální jádrové funkce.

Typ dat	*Normalizace*	n	m	$cAcc(\text{train}_e)$	*Doba trénování*
Slovní přepis	ne	1	1	79,5	2:17:41
Slovní přepis	ano	1	1	79,6	1:16:32
Slovní mřížka	ne	1	1	72,6	2:15:05
Slovní mřížka	ano	1	1	72,1	1:04:23
Slovní mřížka	ne	1	2	71,4	1:26:36
Slovní mřížka	ano	1	2	73,4	0:50:33

Tabulka 10.1: Tabulka shrnující konceptovou přesnost a dobu trénování (h:m:s) pro vybrané volby parametrů.

Na základě výsledků popsaných v odstavcích výše byly pro další experimenty zvoleny následující parametry vstupní vrstvy HDM:

Typ dat	*Normalizace*	n	m
Slovní přepis	ano	1	1
Slovní mřížka	ano	1	2
Fonémová mřížka	ano	1	5

Tabulka 10.2: Parametry vstupní vrstvy HDM.

10.2.2 Skrytá vrstva

Experimenty provedené v rámci této podkapitoly mají za cíl zodpovědět, zda je vhodné vůbec používat skrytou vrstvu a pokud ano, jakým způsobem ji trénovat a jak zvolit práh N parametrizující množinu $\mathcal{S}_N$.

V kapitole 7 byl HDM popsán jako třívrstvá struktura skládající se ze vstupní, skryté a výstupní vrstvy. Vzhledem k použití SVM jak ve skryté, tak ve výstupní vrstvě HDM je možné hodnoty racionální jádrové funkce vypočítané vstupní vrstvou použít přímo ve výstupní vrstvě a tím pádem HDM degradovat na dvouvrstvou architekturu. Závěry ohledně používání dvou- nebo třívrstvé architektu HDM je možné formulovat na základě tabulky 10.3. V tabulce je porovnána přesnost dosahovaná dvouvrstvým hierarchickým diskriminativním modelem sestávajícím se pouze ze vstupní a výstupní vrstvy (HDM-2) s třívrstvým model se vstupní, skrytou a výstupní vrstvou (HDM-3). Pro toto porovnání jsou třívrstvé modely HDM-3 trénovány s dimenzí vektoru $\mathbf{d}(U)$ rovnou 40, tj. $|\mathcal{S}_N| = 40$. Je zřejmé, že třívrstvá architektura poskytuje výrazně vyšší hodnoty konceptové přesnosti. V případě třívrstvého modelu HDM-3 skrytá vrstva efektivně transformuje vstupní příznaky tvořené středními četnostmi n-gramů na příznaky reprezentující „jistotu" výskytu dané sémantické n-tice ve vstupní promluvě.

Tabulka 10.4 pak odpovídá na otázku, jakým způsobem optimalizovat parametry skryté vrstvy. Byly porovnávány dva různé přístupy k trénování skryté vrstvy lišící se počtem volných parametrů a optimalizovanou kriteriální funkcí:

1. Optimalizace *oddělených hodnot* $C^{\mathbf{t}}$ pro každý klasifikátor $f_{\mathbf{t}}$ zvlášť tak, aby byla optimalizována přesnost predikce dané sémantické n-tice $\mathbf{t}$. V tomto případě nejsou development data daného sémantického korpusu vůbec použita.

2. Použití *sdíleného parametru* $C = C^{\mathbf{t}}$ společného všem klasifikátorům $f_{\mathbf{t}}$. Tento parametr je nastaven tak, aby výsledný HDM model optimalizoval konceptovou přesnost nad development daty použitého sémantického korpusu.

Poznamenejme, že na poli SVM je parametr C regularizační metaparametr vážící sumu slack proměnných (de facto chyb nad trénovací množinou) a normu vektoru parametrů. Velmi vysoké hodnoty C vedou na komplexnější oddělovací nadrovinu a nižší hodnotu sumy slack proměnných, ale na druhou stranu také k přetrénování SVM klasifikátoru. Oproti

Typ dat	*Typ modelu*	$cAcc(\text{train}_e)$
Slovní přepis	HDM-2	75,2
Slovní mřížka	HDM-2	68,5
Fonémová mřížka	HDM-2	64,4
Slovní přepis	HDM-3	78,6
Slovní mřížka	HDM-3	73,4
Fonémová mřížka	HDM-3	71,5

Tabulka 10.3: Srovnání konceptové přesnosti dvouvrstvého (HDM-2) a třívrstvého (HDM-3) hierarchického diskriminativního modelu.

tomu nízké nízké hodnoty C vedou k jednoduché oddělovací nadrovině a tím pádem i ke špatné schopnosti predikce.

Proto se doporučuje volit hodnotu parametru C jako čísla z dané množiny (např. geometrická posloupnost, viz [73]) a použít takovou hodnotu, která optimalizuje danou kriteriální funkci. Při použití optimalizace hodnoty $C^{\mathbf{t}}$ pro každý klasifikátor zvlášť byly tyto hodnoty nastavovány na prvky geometrické posloupnosti $(10^{-1}, 10^{-\frac{2}{3}}, 10^{-\frac{1}{3}}, 10^{0}, 10^{+\frac{1}{3}}, 10^{+\frac{2}{3}}, 10^{1}, 10^{+\frac{4}{3}}, 10^{+\frac{5}{3}}, 10^{2})$. Hodnoty sdíleného parametru C pak byly vybírány z posloupnosti (0.5; 0.75; 1.0; 1.25; 1.5; 1.75; 2.0; 2.25; 2.5).

Z tabulky 10.4 je zřejmé, že přístup optimalizace sdíleného parametru C poskytuje konzistentně vyšší hodnoty konceptové přesnosti. Důvodem je zřejmě přímá optimalizace konceptové přesnosti nad development daty, která se pak příznivě promítne i do konceptové přesnosti nad testovacími daty.

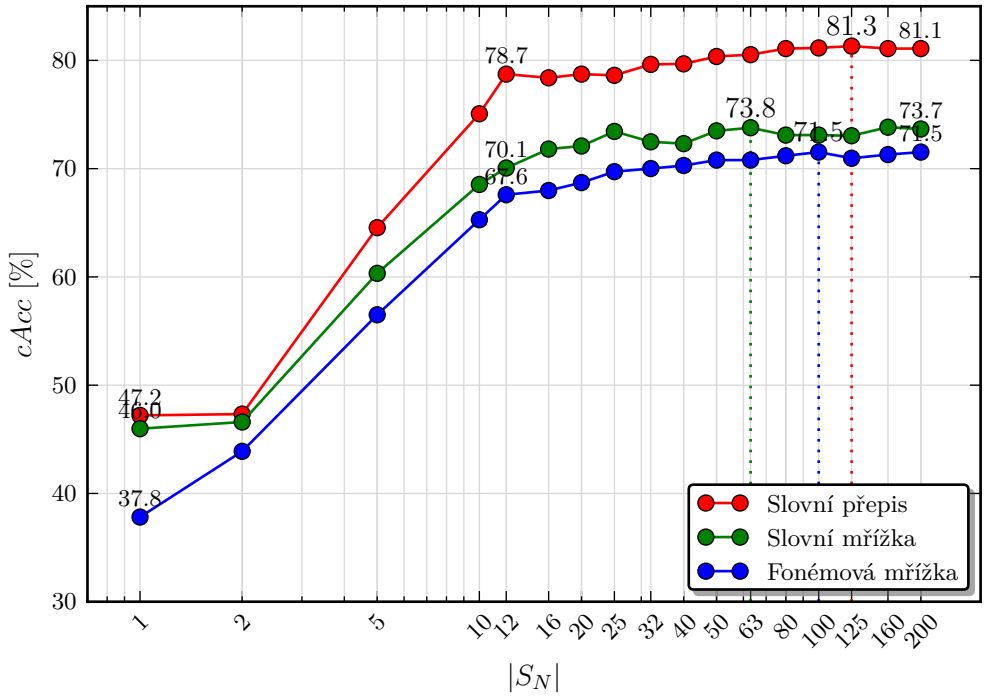
Při použití skryté vrstvy v hierarchickém diskriminativním modelu je nutné nejprve určit vhodný práh N, který určuje množinu (a počet) sémantických n-tic $\mathcal{S}_N$. Pro jeho určení byla nejprve určena konceptová přesnost pro různé velikosti množiny $|\mathcal{S}_N|$. Práh N byl volen tak, aby $|\mathcal{S}_N|$ nabývalo hodnot z řady 1, 2, 5, 10, 12, 16, 20, 25, 32, 40, 50, 63, 80, 100, 125, 160 a 200.

Závislost konceptové přesnosti $cAcc$ na $|\mathcal{S}_N|$ je vynesena na obrázku 10.12. Významné je, křivky mají obdobný průběh pro všechny typy vstupních dat, a tudíž tvar křivky závisí pouze na množině $\mathcal{S}_N$.

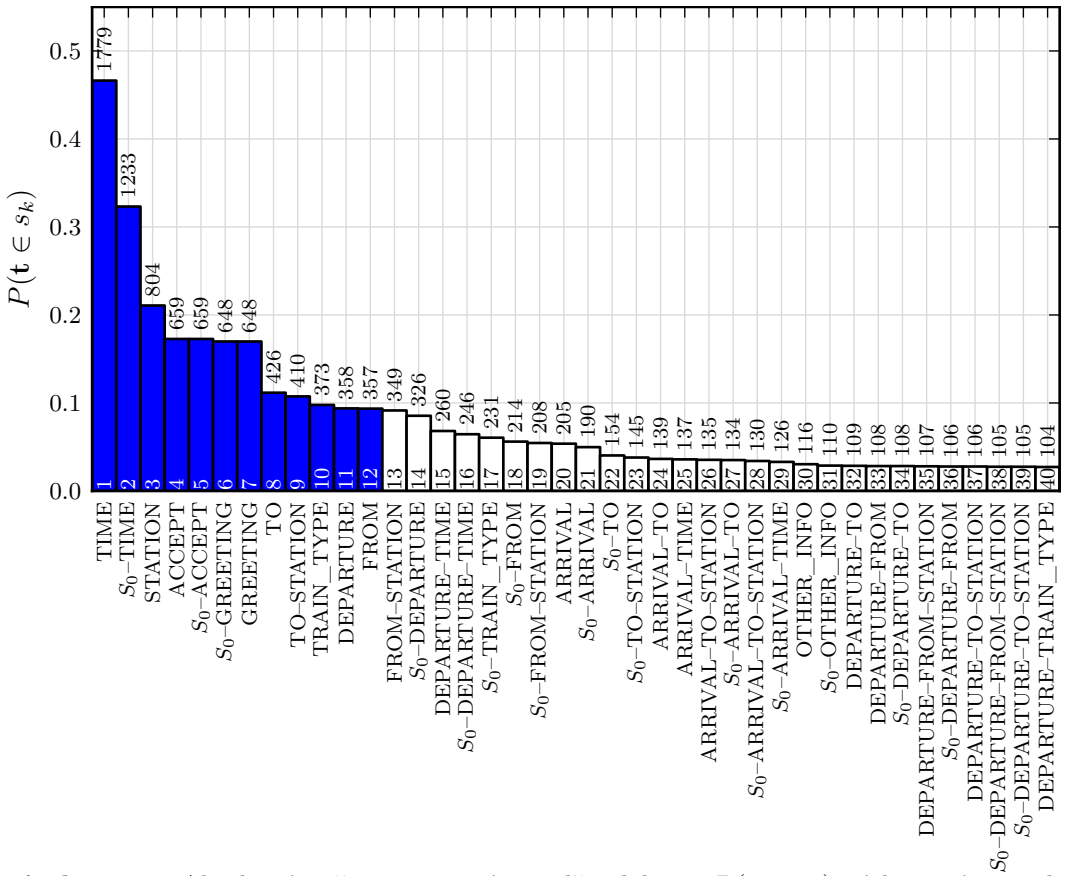
Z průběhů křivek je zřejmé, že 12 nejčastějších sémantických n-tic nese naprostou většinu sémantické informace. Sémantické n-tice uspořádané podle absolutní četnosti v trénovací množině jsou vyneseny do grafu 10.13. V tomto grafu jsou vyneseny absolutní četnosti

	Oddělené $C^{\mathbf{t}}$		Sdílené $C = C^{\mathbf{t}}$	
Typ dat	train_d	train_e	train_d	train_e
Slovní přepis	78,0	78,2	79,7	79,9
Slovní mřížka	70,7	71,2	72,4	73,5
Fonémová mřížka	68,4	69,4	71,5	71,4

Tabulka 10.4: Porovnání konceptové přesnosti při použití optimalizace jednotlivých oddělených parametrů $C^{\mathbf{t}}$ a při použití sdíleného parametru $C = C^{\mathbf{t}}$.



Obrázek 10.12: Hodnoty *cAcc* v závislosti na $|\mathcal{S}_N|$ pro různé typy dat, přesnost *cAcc* vyčíslena nad daty train_e.



Obrázek 10.13: Absolutní počet a apriorní pravděpodobnost $\mathrm{P}(\mathbf{t} \in s_k)$ výskytu sémantických n-tic ve stromech z trénovací množiny train_t.

prvních čtyřiceti nejčetnějších sémantických n-tic ze sady train_t spolu s odhadem apriorní pravděpodobnosti výskytu sémantické n-tice v libovolném stromu s_k, tj. $\text{P}(\mathbf{t} \in s_k)$.

Lze si všimnout, že prvních 12 nejčastějších sémantických n-tic je pak tvořeno pouze sedmi sémantickými koncepty: TIME, STATION, ACCEPT, GREETING, TO, TRAIN_TYPE a DEPARTURE (v grafu 10.13 jsou tyto n-tice modře podbarveny). Dalším poznatkem je, že prvních čtyřicet sémantických n-tic má stále relativně vysokou absolutní četnost – vyšší než 100 trénovacích příkladů – i při relativně nízké apriorní pravděpodobnosti 2,7%. Tímto je možné zdůvodnit další nárůst konceptové přesnosti i po zvyšování $|\mathcal{S}_N|$ nad hranici 12 sémantických n-tic.

Výše uvedený experiment vede na parametry skryté vrstvy uvedené v tabulce 10.5. První důležitý závěr je relativně malý počet pozitivních příkladů potřebných pro natrénování klasifikátorů ve skryté vrstvě (30 pozitivních příkladů). Je důležité poznamenat, že přestože tyto SVM nemusí nutně poskytovat dobrou přesnost predikce přítomnosti konkrétní sémantické n-tice, jimi poskytované vzdálenosti $d_{\mathbf{t}}(U)$ poskytují informativní příznaky pro trénování a predikci ve výstupní vrstvě. Na základě grafu 10.13 lze konstatovat, že při nastavení hodnoty $|\mathcal{S}_N|$ na hodnotu 63 jsou křivky pro konceptovou přesnost již nasyceny a další zvyšování složitosti modelu vede již na nepatrný nárůst konceptové přesnosti. Poznamenejme, že hodnoty $|\mathcal{S}_N| = 63$ na trénovací množině train_t dosáhneme použitím prahu $N = 30$.

Typ dat	$\lvert\mathcal{S}_N\rvert$	N	$cAcc(\text{train}_e)$
Slovní přepis	63	30	80,4
Slovní mřížka	63	30	73,7
Fonémová mřížka	63	30	71,4

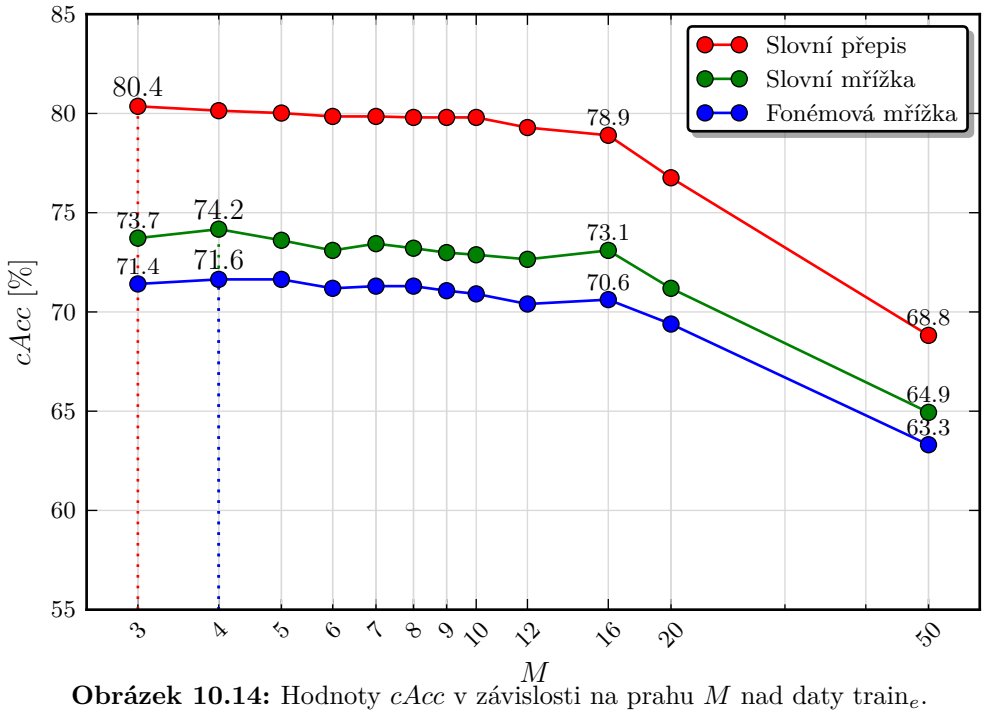
Tabulka 10.5: Parametry skryté vrstvy HDM.

10.2.3 Výstupní vrstva

Pro výstupní vrstvu modelu HDM je možné nastavovat jediný parametr M určující, které množiny potomků β sémantického konceptu A jsou zahrnuty do množiny $\mathcal{B}_A$. Tento parametr nastavuje práh četnosti výskytu pravidel ve tvaru $A \rightarrow \beta$ dle rovnice (7.26).

Ve výstupní vrstvě modelu HDM byly použity SVM klasifikátory klasifikující do více tříd podle schématu one-against-one (kapitola 5.1.5) a poskytující odhad aposteriorní pravděpodobnosti příslušnosti vstupního příznakového vektoru do cílové třídy (kapitola 5.1.6). Ve výstupní vrstvě SVM klasifikátorů byla použita RBF jádrová funkce z rovnice 5.27 s parametrem γ. Hodnoty parametru C a γ pro každý klasifikátor g_A byly určeny prohledáváním všech kombinací; hodnoty C byly vybírány z posloupnosti $(10^{-1}, 10^0, 10^1, 10^2)$ a hodnoty γ z posloupnosti $(2^{-7}, 2^{-6}, 2^{-5}, 2^{-4}, 2^{-3})$. Pro každý klasifikátor byla vybrána ta kombinace parametrů, která poskytovala nejvyšší přesnost predikce nad development daty.

Závislost konceptové přesnosti na parametru M pro různé typy dat je zobrazena na obrázku 10.14. Na základě těchto průběhů byla pro všechny typy vstupních dat zvolena hodnota prahu $M = 4$.



Obrázek 10.14: Hodnoty *cAcc* v závislosti na prahu M nad daty train_e.

Typ dat	M	$cAcc(\text{train}_e)$
Slovní přepis	4	79,7
Slovní mřížka	4	73,2
Fonémová mřížka	4	70,7

Tabulka 10.6: Parametry výstupní vrstvy HDM.

Tabulka 10.7 shrnuje všechny parametry modelu HDM, jejichž volba byla odůvodněna odstavcích výše. Tyto parametry jsou použity v experimentech ve zbytku práce. Poslední tři řádky tabulky pro úplnost uvádějí odvozené parametry i pro další typy dat – pro nejlepší slovní a fonémovou hypotézu a pro pseudofonémovou mřížku.

V tabulce 10.8 jsou uvedeny vlastnosti ovlivňující výpočetní složitost výpočtu racionální jádrové funkce ve vstupní vrstvě HDM pro různé typy dat. Nastavení parametrů racionální jádrové funkce je převzato z tabulky 10.7. Je vidět, že velikost transducerů R měřená počtem stavů a přechodů je pro fonémové mřížky výrazně vyšší než pro mřížky slovní. I přesto lze konstatovat, že maximální velikost transduceru R pro případ fonémové mřížky *ph-bh* a korpusu TIA – necelých 35 megabytů v binární reprezentaci – je velice přijatelná z pohledu možností současné výpočetní techniky.

Typ dat	*Normalizace*	n	m	*Typ modelu*	N	M
Slovní přepis	ano	1	1	HDM-3	30	4
Slovní mřížka	ano	1	2	HDM-3	30	4
Fonémová mřížka	ano	1	5	HDM-3	30	4
Slovní 1. hypotéza	ano	1	2	HDM-3	30	4
Fonémová 1. hypotéza	ano	1	5	HDM-3	30	4
Pseudofon. mřížka	ano	1	5	HDM-3	30	4

Tabulka 10.7: Parametry hierarchického diskriminativního modelu použité v dalších experimentech. Sloupce *normalizace*, n a m ovlivňují vstupní vrstvu, *typ modelu* a N skrytou vrstvu a M vrstvu výstupní.

	HHTT				TIA			
Typ dat	$\|\mathcal{Q}_R\|$	$\|\mathcal{E}_R\|$	D_R	MB	$\|\mathcal{Q}_R\|$	$\|\mathcal{E}_R\|$	D_R	MB
Slovní přepis	1,5	22,3	1,7	0,36	2,1	30,5	3,1	0,49
Slovní 1. hypotéza	5,2	41,8	1,5	0,70	6,1	52,0	2,1	0,86
Slovní mřížka	8,3	77,7	2,0	1,28	7,6	71,7	2,2	1,18
Fonémová 1. hypotéza, *ph-fa*	28,7	396,6	4,2	6,38	45,5	583,4	3,5	9,42
Fonémová mřížka, *ph-fa*	45,4	576,5	4,4	9,32	91,6	1042,6	3,6	16,96
Fonémová mřížka, *ph-bh*	84,6	910,6	4,5	14,86	210,3	2131,3	3,7	34,93
Fonémová mřížka, *ph-ad*	48,4	618,0	4,4	9,99	103,4	1195,7	3,6	19,43
Pseudofon. mřížka, *ph-map*	22,7	571,1	5,3	8,97	27,6	680,8	4,2	10,70

Tabulka 10.8: Přehled vlastností transduceru R získaného z trénovací množiny korpusů HHTT a TIA. Vlastnosti byly vyčísleny pro různé typy dat. V tabulce je uveden počet stavů $|\mathcal{Q}_R|$, počet přechodů $|\mathcal{E}_R|$ a maximální počet přechodů z jednoho stavu D_R v transduceru R. Tyto veličiny jsou vyjádřeny v tisících. Sloupec MB pak ukazuje velikost binárního souboru v knihovně OpenFST v megabytech.

10.3 Vyhodnocení HDM nad neviděnými daty

Pro experimenty v předchozích odstavcích byla použita pouze trénovací množina korpusu HHTT rozdělená do sad train_t, train_d a train_e tak, aby se předešlo přetrénování, které se může vyskytnout při vyhodnocování velkého množství experimentů nad stále stejnou testovací množinou. Prezentujme nyní výsledky HDM nad korpusem HHTT při použití celé trénovací množiny a oddělené development a testovací sady nepoužité v předchozích experimentech. Výsledky budou rovněž porovnány s referenčními modely a vyhodnocení modelu HDM bude také provedeno nad korpusem TIA.

Porovnání výsledků hierarchického diskriminativního modelu a referenčních modelů přináší tabulky 10.9 a 10.10. Referenční HVS model byl trénován jako HVS parser s levopravým větvením bez parametrizace vstupním příznakovým vektorem, tj. pouze s využitím slov dané hypotézy. Tento model odpovídá modelu popsanému v rámci práce [46]. Model STC byl trénován jako HDM model avšak bez výstupní vrstvy, která byla nahrazena dekódovacím algoritmem popsaným v kapitole 5.7.2. Přestože tato implementace nepoužívá náhradu lexikálních tříd popsánou v kapitole 5.7, dosahuje lepších výsledků než HVS parser. Protože model HDM je rozšířením modelu STC o výstupní vrstvu a v žádném

z provedených experimentů nedosahovala použitá implementace STC modelu lepších výsledků z pohledu konceptové přesnosti než model HDM ve stejné konfiguraci, lze se tedy právem domnívat, že případná modifikace STC modelu zvyšující přesnost by vedla na zvýšení přesnosti i po doplnění výstupní vrstvy modelu HDM.

Nejprve je vhodné okomentovat výsledky dosažené referenčními modely a shrnuté v tabulce 10.9. Upozorněme na propad v konceptové přesnosti u modelu HVS při porovnání testovacích sad korpusu HHTT a TIA. Zde rozdíl činí téměř šest procentních bodů, pokud jsou modely trénovány a vyhodnoceny nad slovním přepisem od anotátora. Pokud však porovnáme konceptové přesnosti nad rozpoznanými daty (slovní 1. hypotéza), je tento rozdíl již téměř deset procentních bodů. Pokud se podíváme na výsledky nad development sadou, nejsou tyto rozdíly již tak markantní, lze se tedy domnívat, že při trénování HVS modelu nad korpusem TIA dochází k jistému přetrénování a následně k nízké přesnosti predikce sémantických stromů nad neviděnými daty.

Pro model STC došlo ke zlepšení konceptové přesnosti v porovnání s modelem HVS jak nad slovním přepisem od anotátora, tak nad rozpoznanými slovními posloupnostmi a to konzistentně jak pro development, tak pro testovací sady obou sémantických korpusů. Toto zlepšení je pro korpus HHTT přibližně čtyři procentní body, pro korpus TIA pak devět až deset procentních bodů.

Model HDM byl natrénován za použití parametrů z tabulky 10.7. Přínos tohoto modelu je zřejmý z tabulky 10.10. Podívejme se nejprve na výsledky získané při trénování ze slovního přepisu od anotátora a z první nejlepší rozpoznané hypotézy. Zde model HDM vylepšuje výsledky modelu STC nad korpusem HHTT o přibližně tři procentní body. V porovnání s modelem HVS jsou tyto výsledky lepší o více než osm procentních bodů nad slovním přepisem a o téměř sedm procentních bodů nad rozpoznanými daty. Mnohem markantnější rozdíl je nad sémantickým korpusem TIA, kde HDM přidává oproti modelu STC šest procentních bodů.

Z tabulky lze rovněž vyčíst přínos použití slovních mřížek v porovnání s první nejlepší slovní hypotézou ze systému automatického rozpoznávání řeči. Tento přínos je v případě korpusu HHTT přibližně jeden procentní bod a v případě korpusu TIA dva procentní body. Další řádky ukazují výsledky při použití fonémových mřížek namísto mřížek slovních. Je zde uveden i řádek odpovídající první nejlepší fonémové posloupnosti. Porovnáním tohoto výsledku s výsledky získanými z fonémových mřížek *ph-fa* lze konstatovat, že na fonémové úrovni je příspěvek ke konceptové přesnosti způsobený použitím mřížek ještě nižší než na úrovni mřížek slovních. Další řádky porovnávají konceptové přesnosti modelů trénovaných z mřížek generovaných pomocí různých fonémových jazykových modelů. Poslední řádek pak odkazuje na použití pseudofonémových mřížek, tj. fonémových mřížek vygenerovaných ze slovních mřížek pomocí výslovnostních slovníků. Diskuze výsledků bude uvedena v závěru na straně 151.

		HHTT		TIA	
Model	*Typ dat*	*devel*	*test*	*devel*	*test*
HVS	Slovní přepis	74,2	73,6	69,9	67,9
	Slovní 1. hypotéza	63,4	65,8	61,6	56,3
STC	Slovní přepis	78,7	78,0	73,7	76,8
	Slovní 1. hypotéza	67,4	69,2	69,5	66,4

Tabulka 10.9: Hodnoty konceptové přesnosti (*cAcc*) v procentech pro referenční modely.

		HHTT		TIA	
Model	*Typ dat*	*devel*	*test*	*devel*	*test*
HDM	Slovní přepis	82,7	81,9	80,9	82,6
	Slovní 1. hypotéza	70,2	72,3	73,9	72,9
	Slovní mřížka	70,7	73,5	76,1	74,8
	Fonémová 1. hypotéza, *ph-fa*	67,0	70,9	73,8	70,9
	Fonémová mřížka, *ph-fa*	67,0	71,0	73,8	71,5
	Fonémová mřížka, *ph-bh*	61,5	66,5	67,5	65,7
	Fonémová mřížka, *ph-ad*	68,8	69,8	70,7	69,6
	Pseudofon. mřížka, *ph-map*	73,4	75,6	76,6	75,5

Tabulka 10.10: Hodnoty konceptové přesnosti (*cAcc*) v procentech pro hierarchický diskriminativní model trénovaný z různých typů dat. Parametry HDM byly nastaveny podle tabulky 10.7.

10.4 Detekce sémantických entit

Kromě hierarchického diskriminativního modelu byl popsán i model detekce sémantických entit. Tento model byl rovněž experimentálně ověřen a výsledky provedených experimentů jsou prezentovány v následujících odstavích.

Model detekce sémantických entit byl ověřen na datech ze slovního rozpoznávače řeči v obou použitých úlohách, tj. jak nad korpusem HHTT, tak TIA. Pro ověření tohoto modelu bylo nutné získat expertně navržené gramatiky pro jednotlivé typy sémantických entit.

Gramatiky pro úlohu HHTT byly vyvinuty v rámci letního workshopu 2011 na Katedře kybernetiky Západočeské univerzity v Plzni. Hlasový dialogový systém vyvíjený v rámci tohoto workshopu byl zaměřen na vyvinutí znalostní verze hlasového dialogového systému pro úlohu poskytování informací o odjezdech a příjezdech vlaků [53]. Z tohoto hlasového dialogového systému byly použity gramatiky pro následující typy sémantických entit: *station* (jméno stanice v různých pádech), *time* (časový údaj nebo údaj o datu) a *train_type* (typ vlaku).

Gramatiky pro úlohy TIA byly zpracovány v rámci výzkumného projektu MPO TIP FR-TI1/518 Inteligentní telefonní asistentka. Pro popsané řečové korpusy byly použity gramatiky reprezentující následující typy sémantických entit: *jmeno* (křestní jméno osoby, příjmení nebo jejich kombinace), *vec* (jméno prostředku k rezervaci), *t* (časový údaj), *datum* (údaj o datu).

Vlastnosti bezkontextových gramatik použitých k popisu sémantických entit a odpovídajících transducerů získaných jejich kompilací jsou zachyceny v tabulce 10.11. Pro vyhodnocení modelu detekce sémantických entit je použita modifikovaná ROC křivka popsaná v kapitole 9.3.2. Dále je použita metrika odpovídající ploše pod touto křivkou při 0 až 1 falešném poplachu na jednu promluvu (AUC) definovaná v rovnici (9.27).

Vzhledem k tomu, že ani korpus HHTT ani korpus TIA neobsahují označené referenční sémantické entity, je nutné referenční data vygenerovat pomocí bezkontextových gramatik použitých následně i pro predikci. Tímto přístupem není možné vyhodnotit pokrytí sémantických entit gramatikami, tj. poměr sémantických entit detekovaných pomocí gramatik vůči všem referenčním sémantickým entitám. Je třeba zdůraznit, že maximalizace pokrytí zahrnuje netriviální množství expertní práce a proto není předmětem této práce, která se specializuje na kombinaci statistického a expertního přístupu, nikoli na zdokonalování stávajících expertních znalostí.

Referenční data pro detekci sémantických entit jsou vygenerována z referenčních textových přepisů jednotlivých promluv. Je tak možné vyčíslit především vliv systému automatického rozpoznávání řeči na přesnost detekce sémantických entit. V experimentech se zaměříme na zodpovězení otázky, zda lze při detekci sémantických entit zužitkovat informaci obsaženou ve slovních mřížkách a na porovnání výsledků s přístupem, kdy se pro detekci sémantických entit používá první nejlepší hypotéza.

Detekce sémantických entit byla prováděna nad testovacími množinami korpusů HHTT a TIA. Modifikované ROC křivky a odpovídající hodnoty AUC jsou zachyceny na obrázcích 10.15 a 10.16 (HHTT) a na obrázcích 10.17 a 10.18 (TIA). Dvojice grafů na těchto obrázcích porovnává modifikované ROC křivky při detekci sémantických entit ze slovních mřížek a z první nejlepší hypotézy. Je vidět, že při detekci sémantických entit z *první nejlepší hypotézy* není možné precizně volit vyvážení detekční schopnosti (svislá osa) a počtu falešných poplachů (vodorovná osa). Pro oba korpusy lze zvolit optimální pracovní bod odpovídající přibližně 0,2 falešného poplachu na jednu promluvu při správné detekci přibližně 80% sémantických entit. Poznamenejme, že detekci sémantických entit z první nejlepší hypotézy lze provádět i jednodušším způsobem, než je postup popsaný v kapitole 8, příkladem budiž náhrada lexikálních tříd používaná v STC modelu (kapitola 5.7).

Přínos detekce sémantických entit z mřížek je zřejmý. Z tvaru modifikované ROC křivky lze vyčíst, že jak pro četnost falešných poplachů nižší než 0,2 na jednu promluvu, tak i pro četnost vyšší, lze docílit lepší detekční schopnosti. Například pro korpus TIA lze při snížení četnosti falešných poplachů na 0,1 na jednu promluvu (tj. na 50%) docílit stále velice přijatelné detekční schopnosti kolem 75%. Z hodnot na opačném konci vodorovné osy lze vyčíst, že ze slovní mřížky lze detekovat přibližně o 5% (absolutně) více sémantických entit v porovnání s první nejlepší hypotézou (nárůst z 80% na 85%).

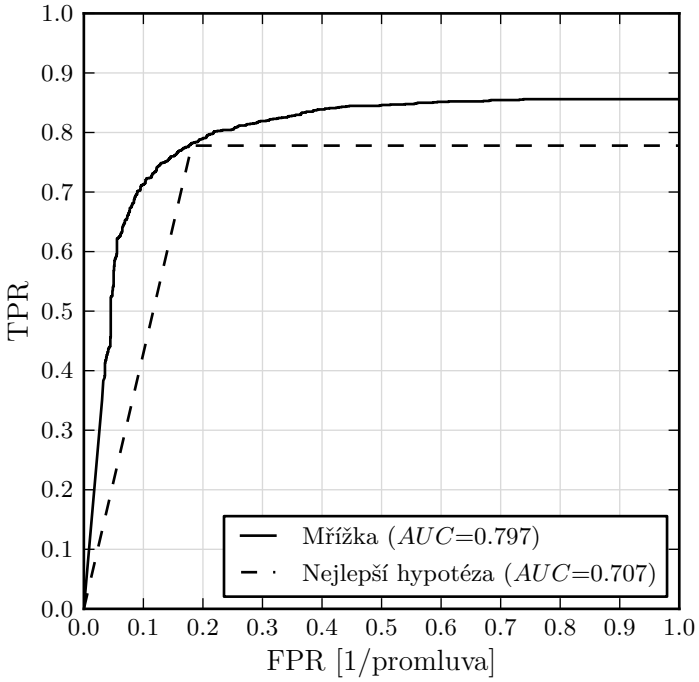
Grafy 10.16 a 10.18 pak zachycují modifikované ROC křivky pro jednotlivé typy sémantických entit. Při detekci byla použity vždy pouze bezkontextové gramatiky odpovídající danému typu sémantické entity, a to jak pro vygenerování referenčních dat, tak i pro získání hypotéz. Z těchto křivek lze vyčíst nepřímou úměrnost mezi složitostí gramatiky

Korpus	*Typ sém. entity z*	$\lvert\mathcal{A}\rvert$	$\lvert\mathcal{B}\rvert$	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{Q}\rvert$	N	$\mathcal{C}(z)$
HHTT	station	7516	3005	34405	5564	417	STATION
	time	437	221	13375	2898	791	TIME
	train_type	16	10	24	11	140	TRAIN_TYPE
TIA	jmeno	105	30	243	31	56	JMENO
	vec	46	14	100	20	31	VEC
	t	159	42	2471	688	414	T, INTERVAL
	datum	324	180	10904	2210	359	DATUM, RELATIVNI

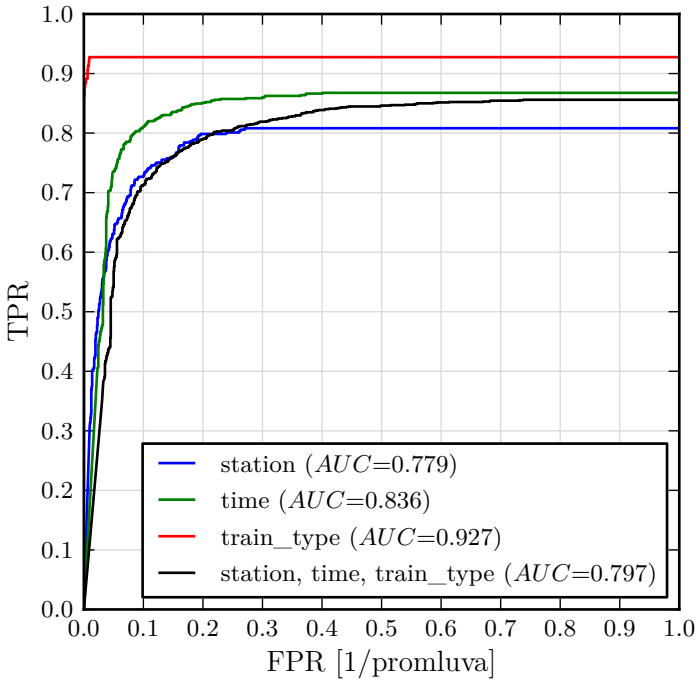
Tabulka 10.11: Tabulka shrnující vlastnosti gramatik a transducerů vzniklých jejich kompilací pro jednotlivé typy sémantických entit. Sloupce $\lvert\mathcal{A}\rvert$ a $\lvert\mathcal{B}\rvert$ vyjadřují velikost vstupní, resp. výstupní abecedy transduceru T_z získaného z gramatiky pro typ z, sloupce $\lvert\mathcal{E}\rvert$ a $\lvert\mathcal{Q}\rvert$ pak odpovídají počtu stavů, resp. přechodů v T_z a N zobrazuje celkový počet výskytů označených touto gramatikou v testovací sadě daného korpusu. V posledním sloupci $\mathcal{C}(z)$ je uvedena množina konceptů, na které lze sémantickou entitu typu z zarovnat v modelu zarovnání.

daného typu sémantické entity a přesností jeho detekce vyjádřenou pomocí hodnoty AUC. Zatímco relativně jednoduché sémantické entity typu *train_type* (HHTT) nebo *vec* (TIA) odpovídající zpravidla pouze výčtu několika (desítek) slov dosahují hodnot AUC přesahující 0,9, řádově složitější sémantické entity typu *station* (HHTT) nebo *datum* (TIA) pak odpovídají nižším hodnotám AUC kolem 0,8.

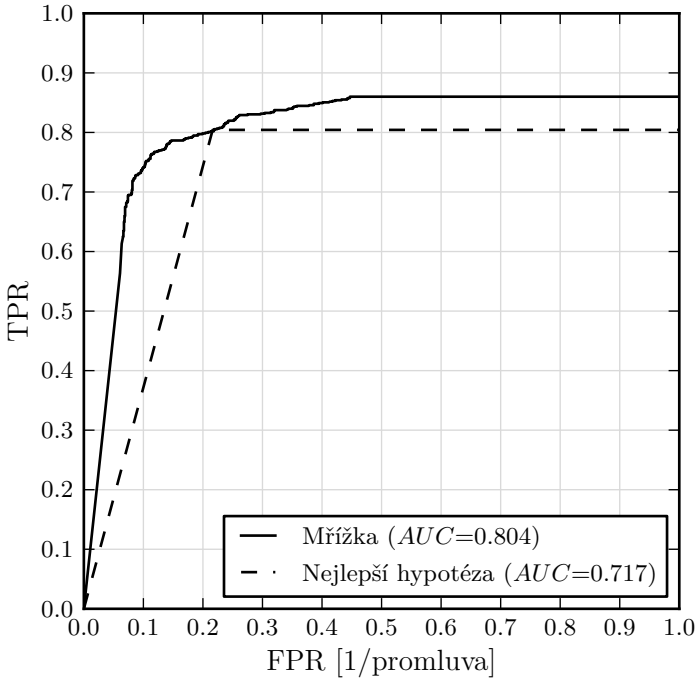
Z výše uvedeného vyplývá, že detekce sémantických entit z mřížek má svůj přínos především při redukci četnosti falešných poplachů. Dále díky tomu, že poskytuje i aposteriorní pravděpodobnost výskytu dané posloupnosti sémantických entit ve vstupní promluvě, je vhodná i pro nasazení v hlasových dialogových systémech se stochastickým řízením dialogu, které jsou schopny tuto neurčitost zužitkovat. V neposlední řadě lze detekci sémantických entit použít i v kombinaci s hierarchickým diskriminativním modelem pro porozumění mluvené řeči.



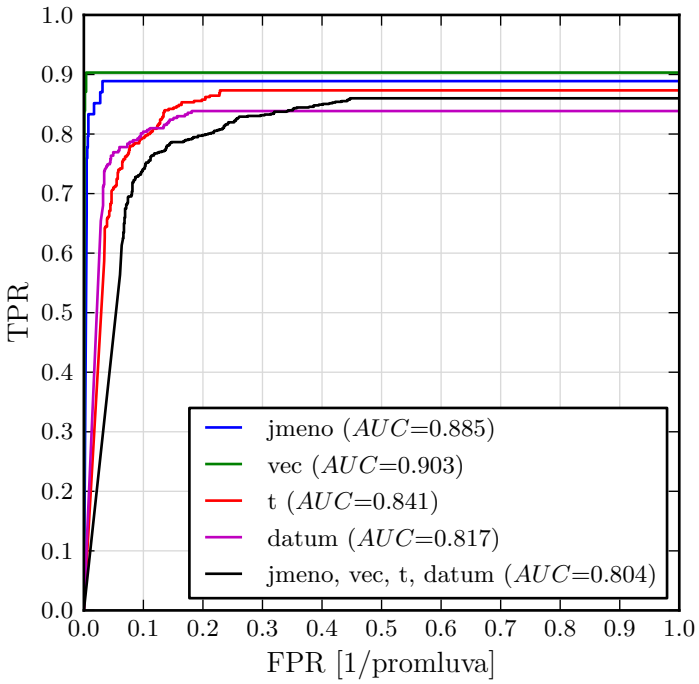
Obrázek 10.15: ROC křivka pro detekci sémantických entit nad slovní mřížkou a nad první nejlepší hypotézou v korpusu HHTT.



Obrázek 10.16: Detailní ROC křivka pro detekci jednotlivých typů sémantických entit *station*, *time* a *train_type* ze slovních mřížek v korpusu HHTT. Černá křivka odpovídá detekci všech typů najednou.



Obrázek 10.17: ROC křivka pro detekci sémantických entit nad slovní mřížkou a nad první nejlepší hypotézou v korpusu TIA.



Obrázek 10.18: Detailní ROC křivka pro detekci jednotlivých typů sémantických entit *jmeno*, *vec*, *t* a *datum* ze slovních mřížek v korpusu TIA. Černá křivka odpovídá detekci všech typů najednou.

10.5 Kombinace HDM a detekce sémantických entit

Popišme nyní experimentální ověření kombinace detekce sémantických entit $\mathrm{P}(\mathsf{E}|\mathsf{U})$, konceptového modelu $\mathrm{P}(\mathsf{C}|\mathsf{E},\mathsf{U})$ a modelu zarovnání $\mathrm{P}(\mathsf{A}|\mathsf{E},\mathsf{C})$ podle teorie popsané v kapitole 6. V této kapitole budeme uvažovat konfiguraci modelu detekce sémantických entit shodnou s předchozí kapitolou 10.4. Jako konceptový model použijeme hierarchický diskriminativní model s parametry popsanými v kapitole 10.2.

Pro kombinaci těchto modelů je nutné použít model zarovnání $\mathrm{P}(\mathsf{A}=1|\mathsf{E}=e,\mathsf{C}=c)$. Předpokládejme, že e je posloupnost sémantických entit e_i, entita e_i má typ z_i a množina sémantických konceptů, které mohou být zarovnány s daným typem z_i nazvěme $\mathcal{C}(z_i)$. Model zarovnání byl použit v následujícím tvaru:

$$\mathrm{P}(\mathsf{A}=1|\mathsf{E}=e,\mathsf{C}=c)=\lambda^n\cdot(1-\lambda)^m \tag{10.1}$$

kde n je počet sémantických entit z posloupnosti e, pro které lze jejich typ z_i zarovnat s některým sémantickým konceptem c_k ze stromu c (tj. $c_k\in\mathcal{C}(z_i)$). Dále m je počet sémantických entit, které výše uvedeným způsobem zarovnat nelze. Platí $|e|=n+m$. Hodnota parametru $\lambda=0,95$ byla experimentálně určena na development sadách použitých korpusů. Množiny $\mathcal{C}(z)$ byly určeny expertně a pro jednotlivé typy sémantických entit z jsou uvedeny v tabulce 10.11.

Pro vyhodnocení kombinace konceptového modelu (reprezentovaného hierarchickým diskriminativním modelem), modelu detekce sémantických entit a modelu zarovnání byl zrealizován experiment, kdy byla použita informace o sémantických entitách v HDM modelu a následně bylo provedeno zarovnání abstraktního sémantického stromu a posloupnosti sémantických entit, přičemž výsledná pravděpodobnost byla vážena pravděpodobností přiřazenou modelem zarovnání z rovnice (10.1). Tento postup je ekvivalentní reskórování (přeuspořádání) n-nejlepších sémantických stromů vzhledem k možným posloupnostem sémantických entit.

Při trénování modelu HDM byla použita informace z modelu detekce sémantických entit. Výsledný model pak predikuje pravděpodobnost $\mathrm{P}(\mathsf{C}|\mathsf{E},\mathsf{U})$. Z mřížky sémantických entit E byly získány střední počty výskytů $cnt(E,z_k)$ jednotlivých typů sémantických entit z_k a z nich byl sestaven vektor příznaků $\mathbf{d}(E)=[cnt(E,z_k)]$, přičemž pro korpus HHTT bylo $z_k\in\{station,\ time,\ train\_type\}$ a pro korpus TIA $z_k\in\{jmeno,\ vec,\ t,\ datum\}$. Následně byl vektor příznaků na výstupu skryté vrstvy doplněn o vektor $\mathbf{d}(E)$, jak je popsáno na straně 79.

Pro vyhodnocení byla použita míra *cAcc* abstraktního sémantického stromu predikovaného kombinovaným modelem. Tabulka 10.12 zobrazuje konceptové přesnosti dosahované samotným HDM modelem a modelem, ve kterém je použita kombinace HDM s detekcí sémantických entit a modelem zarovnání. Z tabulky je zřejmé, že využitím informace o sémantických entitách lze dosáhnout zvýšení přesnosti predikce abstraktních sémantických stromů. K tomuto nárůstu dochází konzistentně na různých typech dat a zároveň jak na development, tak testovací sadě, přičemž k výraznějšímu nárůstu sémantické přesnosti dochází na korpusu HHTT.

Pouze mírný nárůst v přesnosti při přidání příznaků a informace o sémantických entitách v korpusu TIA je způsoben s největší pravděpodobností strukturou gramatik použitých pro detekci sémantických entit. Tyto gramatiky byly získány z prototypu aplikace Telefonní inteligentní asistentka. Proto obsahují především jména a prostředky (sémantické

entity typu *jmeno* a *vec*) zanesené v databázi aplikace TIA. Tyto gramatiky však pouze z malé části pokrývají všechny možné lexikální realizace konceptů JMENO a VEC použité v korpusu TIA. Testovací sada tohoto korpusu obsahuje 129 výskytů konceptu JMENO a 53 výskytů konceptu VEC. Z tabulky 10.11 je však zřejmé, že v téže sadě bylo pomocí gramatik sémantických entit detekováno pouze 56 sémantických entit typu *jmeno* a 31 sémantických entit typu *vec*. Gramatikami je tak pokryto pouze 43%, resp. 58% lexikálních realizací těchto konceptů.

Pro detailnější vyčíslení přínosu detekce sémantických entit v kombinovaném modelu byly vyčísleny míry přesnosti P_C, úplnosti R_C a F-míry F_C (rovnice (9.9) až (9.11)) pro sémantické koncepty náležící do množin $\mathcal{C}(z)$ z tabulky 10.11. Hodnoty těchto měr jsou zaneseny do tabulky 10.13. Odtud je zřejmé, že nejvyšší přínos v úloze HHTT je způsobem nárůstem F-míry pro koncepty TRAIN_TYPE (typ vlaku, z 84,8% na 89,6%) a STATION (stanice, z 79,5% na 84,8%). U korpusu TIA se pak největší nárůst F-míry projevil u konceptů VEC (prostředek k rezervaci, z 86,8% na 88,5%) a T (čas, z 65,6% na 67,6%).

		HHTT		TIA	
Model	*Typ dat*	*devel*	*test*	*devel*	*test*
konceptový model	Slovní přepis	82,7	81,9	80,9	82,6
$\mathrm{P}(\mathsf{C}\|\mathsf{U})$	Slovní mřížka	70,7	73,5	76,1	74,8
	Pseudofon. mřížka	73,4	75,6	76,6	75,5
kombinace	Slovní přepis	83,8	83,4	82,1	83,4
$\mathrm{P}(\mathsf{C}\|\mathsf{E},\mathsf{U})\cdot$	Slovní mřížka	72,2	76,2	77,2	75,1
$\cdot\mathrm{P}(\mathsf{E}\|\mathsf{U})\mathrm{P}(\mathsf{A}\|\mathsf{E},\mathsf{C})$	Pseudofon. mřížka	74,1	76,9	76,7	75,4

Tabulka 10.12: Porovnání konceptové přesnosti samotného konceptového modelu a kombinace konceptového modelu, modelu detekce sémantických entit a modelu zarovnání.

		konceptový model			*kombinace*		
Korpus	*Sémantický koncept C*	P_C	R_C	F_C	P_C	R_C	F_C
HHTT	TIME	93,5	92,3	**92,9**	93,0	93,9	**93,4**
	TRAIN_TYPE	94,0	77,3	**84,8**	94,5	85,1	**89,6**
	STATION	93,3	69,3	**79,5**	93,9	77,3	**84,8**
TIA	INTERVAL	89,4	91,5	**90,4**	89,1	92,9	**91,0**
	DATUM	88,7	88,0	**88,4**	87,0	90,0	**88,4**
	VEC	86,8	86,8	**86,8**	90,2	86,8	**88,5**
	JMENO	91,2	64,3	**75,4**	93,2	63,6	**75,4**
	T	88,9	52,0	**65,6**	83,0	57,1	**67,7**
	RELATIVNI	88,5	46,9	**61,3**	85,2	46,9	**60,5**

Tabulka 10.13: Porovnání přesnosti P_C, úplnosti R_C a F-míry F_C pro různé sémantické koncepty C. Hodnoty byly vyčísleny pro samotný konceptový model a pro jeho kombinaci s modelem detekce sémantických entit a modelem zarovnání.

10.6 Křivky učení

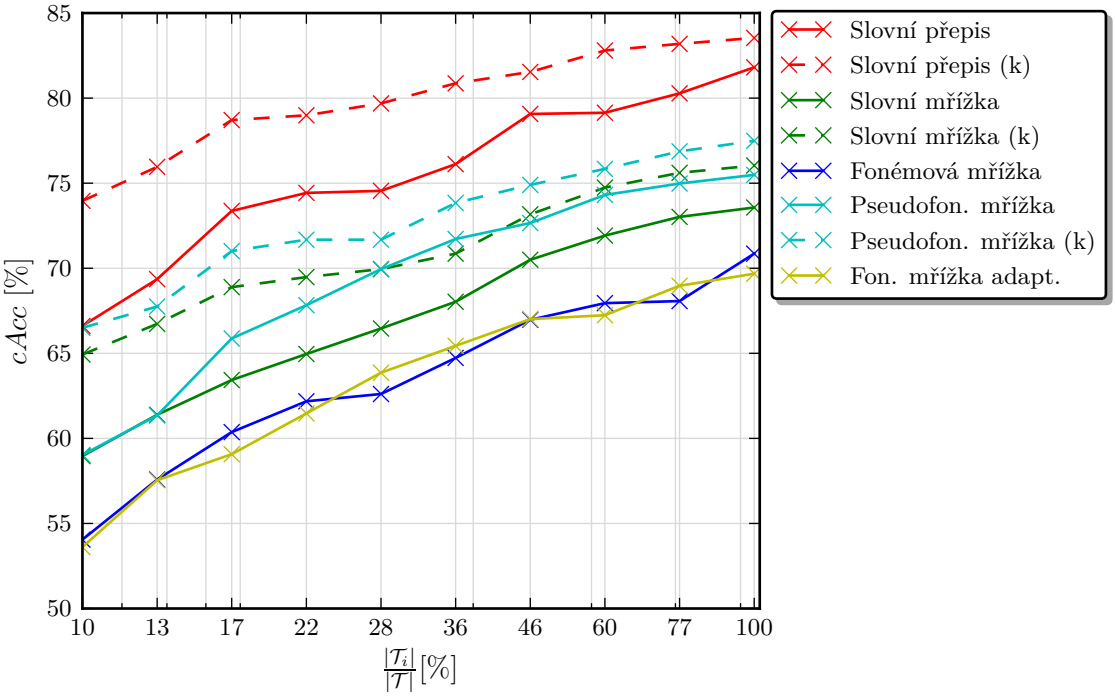
Poslední sada experimentů, která byla provedena, sloužila ke stanovení *křivek učení*, tj. závislostí konceptové přesnosti natrénovaného modelu porozumění na množství trénovacích dat. Tyto křivky jsou vyneseny do grafů na obrázcích 10.19 a 10.20. Na vodorovné ose v těchto grafech je poměr počtu použitých trénovacích promluv k počtu promluv v celé trénovací sadě $\frac{|\mathcal{T}_i|}{|\mathcal{T}|}$. Podmnožiny $\mathcal{T}_i \subseteq \mathcal{T}$ byly vybírány náhodně. Svislá osa zachycuje konceptovou přesnost nad testovací sadou daného korpusu při použití dané trénovací množiny $\mathcal{T}_i$. Poznamenejme, že tyto křivky jsou svým způsobem nadhodnocené, neboť se v nich bere v úvahu pouze množství sémanticky anotovaných vět. Počet vět použitých pro trénování jazykových modelů jak na slovní, tak fonémové úrovni, je konstantní a odpovídá konfiguraci experimentů z předchozích kapitol.

Červené křivky v grafech odpovídají HDM modelům trénovaným z referenčních textových přepisů. Zelené křivky pak modelům trénovaným nad slovními mřížkami. Tyrkysovou barvou jsou vyneseny výsledky modelů trénovaných nad pseudofonémovými mřížkami generovanými ze slovních mřížek. Pro tyto modely jsou dále čárkovanou čarou vyneseny křivky, které odpovídají kombinaci modelu HDM s detekcí sémantických entit a modelem zarovnání. Pro srovnání jsou uvedeny i křivky učení pro HDM modely trénované z fonémových mřížek. Fonémovým mřížkám získaným z fonémového jazykového modelu *ph-fa* odpovídá modrá křivka, fonémovým mřížkám získaným z adaptovaného fonémového jazykového modelu *ph-ad* žlutá křivka.

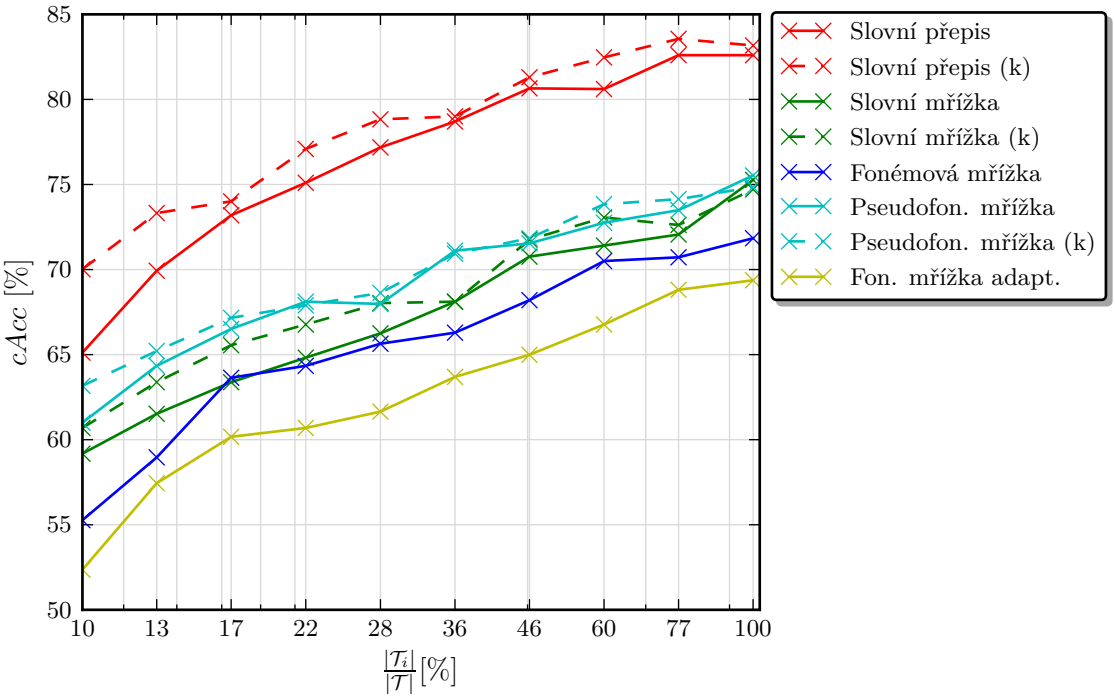
Z těchto křivek je možné vyčíst celou řadu závěrů:

- Přidáním expertní znalosti v podobě detekce sémantických entit založené na gramatikách lze zvýšit přesnost predikce sémantických stromů. Ukazuje se, že toto zvýšení je tím výraznější, čím méně je použito trénovacích dat. Přidání expertní znalosti vede vždy na zvýšení přesnosti predikce sémantických stromů. V několika málo bodech různých křivek u korpusu TIA ke zvýšení přesnosti nedošlo, což je způsobeno pravděpodobně různými náhodnými vlivy při vzorkování trénovací množiny $\mathcal{T}_i$ a při automatickém rozpoznávání řeči nebo při detekci sémantických entit.
- Použití pseudofonémové mřížky vede na zlepšení sémantické přesnosti v porovnání se slovní mřížkou a to prakticky pro všechny hodnoty $\frac{|\mathcal{T}_i|}{|\mathcal{T}|}$ a pro obě úlohy (HHTT a TIA)
- Pro úlohu HHTT vede použití fonémového jazykového modelu *ph-fa* a adaptovaného fonémového jazykového modelu *ph-ad* na téměř totožné křivky. Toto je pravděpodobně způsobeno použitím akustického modelu trénovaného z korpusu HHTT. Při adaptaci fonémového jazykového modelu je pak tímto způsobem nepřímo použita i slovní transkripce.[1] U úlohy TIA je rozdíl mezi těmito křivkami kolem tří procentních bodů *cAcc*. Do budoucna by proto bylo jistě zajímavé provést experiment, v němž by byl adaptován i akustický model fonémového rozpoznávače.
- Pro zvolené hodnoty poměru $\frac{|\mathcal{T}_i|}{|\mathcal{T}|}$ je odstup křivek modelů získaných ze slovních mřížek (zelená křivka) a modelů získaných z fonémových mřížek (modrá a žlutá křivka) téměř konstantní a odpovídá necelým pěti procentním bodům *cAcc*.

[1] Slovní transkripce je použita v Expectation fázi E-M algoritmu pro trénování akustického modelu založeného na HMM.



Obrázek 10.19: Křivky učení pro různé modely nad korpusem HHTT.



Obrázek 10.20: Křivky učení pro různé modely nad korpusem TIA.

Kapitola 11

Závěr

Vytvořený diskriminativní model pro porozumění řeči je vyhodnocen nad testovacími sadami korpusů HHTT a TIA. Výsledky porozumění z referenčního slovního přepisu (tabulka 11.1), porozumění z rozpoznaných slov (tabulka 11.2) a porozumění z rozpoznaných fonémů (tabulka 11.3) jsou vyjádřeny v podobě konceptové přesnosti *cAcc* spolu s 95% intervalem spolehlivosti. Rovněž jsou zde uvedeny hodnoty větné přesnosti *sAcc*.

První modelový problém – *porozumění z referenčního slovního přepisu* (tabulka 11.1) – srovnává výsledky referenčních modelů (HVS parser a STC model) s HDM modelem a s kombinací HDM modelu a modelu detekce sémantických entit. Tento modelový problém odpovídá porozumění z textového přepisu bez neurčitosti způsobené systémem automatického rozpoznávání řeči. Přestože HDM model nebyl pro tuto úlohu primárně navržen, ukazuje se, že i zde překonává výsledky obou referenčních modelů.

Druhý modelový problém – *porozumění z rozpoznaných slov* (tabulka 11.2) – srovnává opět výsledky referenčních modelů s HDM modelem a odpovídá skutečnému nasazení v hlasových dialogových systémech. V tabulce je zobrazen nárůst konceptové přesnosti při použití slovních mřížek namísto první nejlepší hypotézy a dále při převedení slovních mřížek na mřížky pseudofonémové. Jako poslední modifikace je k HDM trénovanému z pseudofonémových mřížek přidána detekce sémantických entit.

Třetí modelový problém – *porozumění z rozpoznaných fonémů* (tabulka 11.3) – odpovídá situaci, kdy není dostupné dostatečné množství dat pro získání přesného a aktuálního jazykového modelu. Pro redukci objemu prací nutných k získání slovního přepisu je možné v tomto případě použít adaptaci fonémového jazykového modelu (kapitola 9.4.4). Tabulka ukazuje vývoj konceptové přesnosti nejprve při trénování z fonémových mřížek získaných použitím obecného fonémového jazykového modelu trénovaného z korpusu BH (řádek *ph-bh*), dále pak použitím adaptovaného fonémového jazykového modelu (řádek *ph-ad*) a použitím fonémového jazykového modelu získaného ze zarovnaného slovního přepisu (řádek *ph-fa*). Pro srovnání je uveden i řádek odpovídající pseudofonémovým mřížkám generovaným ze slovních mřížek (řádek *ph-map*).

Z tabulky je vidět, že použití adaptovaných fonémových jazykových modelů zvýší přesnost porozumění v porovnání s obecným fonémovým jazykovým modelem, dosahované výsledky se téměř blíží výsledkům modelu trénovaného z mřížek získaných s využitím fonémového jazykového modelu ze zarovnaných slovních přepisů. Tento argument podporuje závěr, že na základě rozpoznané fonémové mřížky lze velice jednoduše provádět klasifikaci pro-

mluvy, a to nejen jednoduchými značkami, ale lze jim pomocí modelu HDM přiřazovat i komplexnější struktury v podobě abstraktních sémantických stromů.

Výsledek nad pseudofonémovými mřížkami také napovídá, že použitím výrazně lepšího fonémového rozpoznávače (srovnejte přesnost rozpoznávání při použití jednotlivých fonémových jazykových modelů uvedenou v tabulce 9.12) lze snadno dosáhnout lepších výsledků než při použití samotných slovních mřížek.

Experimentální ověření parametrů vstupní vrstvy modelu HDM (kapitola 10.2.1) dále ukázalo, že v případě trénování HDM modelu z fonémových mřížek má naprosto zásadní vliv normalizace jádrové funkce popsaná v kapitole 5.1.7. Bez použití normalizace lze odhadnout, že dosahované hodnoty konceptové přesnosti by mohly být o pět procentních bodů horší.

Přestože zde prezentované metody byly ověřeny nad sémantickými korpusy HHTT a TIA, které obsahovaly pouze promluvy v češtině, nejsou tyto metody omezeny pouze na tento jazyk. I když nárůst konceptové přesnosti při použití pseudofonémových mřížek (oproti mřížkám slovním) lze z jisté části přičíst flektivní povaze češtiny, může uvedené předzpracování vstupních dat přinést jistá pozitiva i pro neflektivní jazyky, obzvláště při použití systému automatického rozpoznávání řeči. Více je uvedeno v závěrečném zhodnocení cílů disertační práce (kapitola 11.1, *Cíl 2*).

	HHTT			TIA		
Model/modifikace	*cAcc*	95% i.s.	*sAcc*	*cAcc*	95% i.s.	*sAcc*
HVS	73,6	71,5÷75,8	68,9	67,9	65,2÷70,5	55,4
STC	78,0	76,1÷80,0	73,3	76,8	74,1÷79,5	73,4
slovní přepis	81,9	80,1÷83,7	77,0	82,6	80,3÷84,9	78,6
+ detekce sém. entit	**83,4**	81,7÷85,1	78,6	**83,4**	81,2÷85,7	79,0

Tabulka 11.1: Porozumění z referenčního slovního přepisu

	HHTT			TIA		
Model/modifikace	*cAcc*	95% i.s.	*sAcc*	*cAcc*	95% i.s.	*sAcc*
HVS	65,8	63,4÷68,2	63,2	56,3	52,7÷59,8	46,9
STC	69,2	66,9÷71,5	66,0	66,4	63,4÷69,3	59,7
1. hypotéza	72,3	70,2÷74,5	68,8	72,9	70,2÷75,6	66,0
slovní mřížka	73,5	71,4÷75,7	70,0	74,8	72,2÷77,4	68,2
pseudofon. mřížka	75,6	73,6÷77,6	71,7	75,5	73,0÷78,0	68,5
+ detekce sém. entit	**76,9**	75,0÷78,9	71,5	**75,4**	72,8÷78,1	69,1

Tabulka 11.2: Porozumění z rozpoznaných slov

	HHTT			TIA		
Model/modifikace	*cAcc*	95% i.s.	*sAcc*	*cAcc*	95% i.s.	*sAcc*
ph-bh	66,5	63,4÷68,7	63,1	65,7	62,8÷68,6	59,5
ph-ad	69,8	67,7÷72,0	66,1	69,6	66,8÷72,4	62,5
ph-fa	71,0	68,9÷73,1	67,0	71,5	68,7÷74,2	64,3
ph-map (pseudofon.)	75,6	73,6÷77,6	71,7	75,5	73,0÷78,0	68,5

Tabulka 11.3: Porozumění z rozpoznaných fonémů

Vysvětlivky k tabulkám 11.1-11.3: Tabulky shrnující výsledky pro tři modelové úlohy. Sloupec *cAcc* vyjadřuje konceptovou přesnost v %, ve sloupci 95% i.s. jsou zaneseny 95% intervaly spolehlivosti konceptové přesnosti v % a sloupec *sAcc* zobrazuje větnou přesnost v % (procento dekódovaných abstraktních sémantických stromů, které se shodují s referenční anotací). Výsledky byly vyhodnoceny nad testovacími sadami korpusů HHTT a TIA. Řádky uvedené jako HVS a STC obsahují výsledky referenčních modelů.

11.1 Splnění cílů disertační práce

Uveďme nyní seznam cílů disertační práce spolu s komentářem a s odkazy na podrobnější text, který popisuje způsob řešení a splnění těchto cílů. Samotné cíle a motivace, proč byly vybrány jako předmět této disertační práce, jsou shrnuty v kapitole 4.

Cíl 1: *Vyvinutí modelu porozumění schopného pracovat s neurčitostí vstupu i výstupu*

Neurčitost je základní jistotou ve všech systémech automatického zpracování řeči. Proto je vhodné, aby systém porozumění mluvené řeči byl schopen zpracovat nejen neurčitost na svém vstupu způsobenou subsystémem automatického rozpoznávání řeči, ale také aby byl schopen vygenerovat větší množství hypotéz o významu promluvy spolu s jejich aposteriorními pravděpodobnostmi.

Základní vlastnosti popsaného diskriminativního modelu jsou dány pravděpodobnostním modelem popsaným v kapitole 6. V rámci tohoto modelu je možné plně využít neurčitý hypotetický přepis vstupní promluvy. Z tohoto pohledu je významný fakt, že oba dále popsané modely – hierarchický diskriminativní model (kapitola 7) a model detekce sémantických entit (kapitola 8) – umožňují zpracovat mřížky ze systému automatického rozpoznávání řeči. V modelu HDM je schopnost zpracovat rozpoznanou mřížku dána použitím vstupní vrstvy založené na teorii racionálních jádrových funkcí (kapitola 7.1). Pro detekci sémantických entit pak model detekce sémantických entit používá heuristiku jednoznačného maximálního přiřazení (kapitola 8.1).

Více výstupních hypotéz pak umožňuje jednak struktura výstupní vrstvy hierarchického diskriminativního modelu (kapitola 7.3) a také algoritmus, umožňující rekonstruovat mřížku sémantických entit ze vstupní promluvy v modelu detekce sémantických entit (kapitola 8.2).

Z experimentálního vyhodnocení je zřejmé, že použitím mřížek lze dále zvýšit přesnost porozumění řeči v porovnání se slovními mřížkami. U hierarchického diskriminativního modelu je možné při použití slovní mřížky dosáhnout zvýšení přesnosti o jeden až dva procentní body konceptové přesnosti v porovnání s použitím první nejlepší hypotézy (kapitola 10.3). Při detekci sémantických entit je přínos slovních mřížek výraznější, při jejich použití lze dosáhnout zvýšení hodnoty AUC přibližně o devět procentních bodů (kapitola 10.4).

Výhody více výstupních hypotéz se projeví, pokud jsou výstupy modelu HDM a detekce sémantických entit zkombinovány pomocí modelu zarovnání. Takto lze přeuspořádat (přeskórovat) výstupní hypotézy modelu HDM tak, aby byly vybrány lepší hypotézy na základě sémantických entit obsažených ve vstupní promluvě. Experimenty podporující tento závěr jsou popsány v kapitole 10.5.

Cíl 2: *Využití fonémového rozpoznávače v oblasti porozumění řeči*

Druhý cíl této disertační práce je velice inovativní, v dostupné literatuře se nepodařilo nalézt publikace věnující se tomuto tématu. Nicméně, jak bylo ukázáno v celé kapitole 10 věnované experimentálnímu ověření navržených metod, porozumění z automatického fonémového přepisu je možné. Díky využití racionálních jádrových funkcí ve vstupní vrstvě modelu HDM je možné velice rychle vyčíslit vektor hodnot těchto funkcí nad vstupní promluvou a všemi promluvami z trénovací množiny.

Je nutné podotknout, že v byly zkoumány pouze metody pro přiřazení abstraktního sémantického stromu na základě automatického fonémového přepisu nebo fonémové mřížky. Nicméně i detekce sémantických entit z fonémových mřížek je řešena pod vedením autora a její výsledky jsou sepsány v diplomové práci [147].

Na základě popsaných experimentů se ukazuje, že díky použití fonémového rozpoznávače je možné realizovat predikci abstraktních sémantických stromů s propadem kolem tří procentních bodů konceptové přesnosti ve srovnání se slovními mřížkami (tabulka 10.10). Jak bylo ukázáno v kapitole 9.4.4, lze fonémový jazykový model dané úlohy získat pomocí adaptace obecného fonémového jazykového modelu.

Poznamenejme, že při použití pseudofonémových mřížek (kapitola 9.4.5) vygenerovaných ze slovních mřížek lze naopak dosáhnout vyšší přesnosti porozumění. Jak vyplývá z provedné analýzy, tento jev je způsoben tím, že převedením slovních mřížek na fonémové dochází k jistému lingvistickému předzpracování vstupní promluvy a tím pádem je možné dosáhnout vyšší přesnosti. Toto lingvistické předzpracování spočívá především ve schopnosti tohoto modelu vyčíslit i podobnost dvou různých slov, např. *jede* a *pojede*. Při použití slovní mřížky se jedná o dvě rozdílná slova, použití pseudofonémových mřížek však vede na nenulovou podobnost, která je vyčíslena racionální jádrovou funkcí ve vstupní vrstvě modelu HDM.

Porozumění z pseudofonémových mřížek velice těží z flektivní podstaty češtiny, nicméně i pro neflektivní jazyky by mohlo mít svůj přínos, neboť mnoho chyb v rozpoznávání řeči se vyznačuje tím, že rozpoznaná posloupnost slov – přestože se liší od správného přepisu – odpovídá referenci naprosté většině fonémů. To je nejvíce způsobeno rozkladem pravděpodobnostního modelu rozpoznávání řeči na model akustický a model jazykový. Při chybě rozpoznávání může dojít například k chybnému odhadu jazykové pravděpodobnosti a tím pádem k vygenerování chybné slovní hypotézy, nicméně tato slovní hypotéza je stále generována tak, aby ve smyslu akustického modelování odpovídala pozorovanému akustickému signálu.

Porozumění založené na fonémovém rozpoznávači nevyžaduje slovní jazykový model, díky tomu lze urychlit vývoj vstupního řetězce hlasového dialogového systému, neboť právě sběr dat pro jazykový model je často nejobsáhlejší a nejnáročnější položkou při vývoji takového systému.

I přesto, že není vyžadován slovní jazykový model, je nutné pro trénování modelu porozumění získat sémantické anotace nad nemalým množstvím trénovacích dat. Zde však lze často použít buď zjednodušené anotace (hovory se ručně klasifikují do cílových tříd, není však pořizován jejich slovní přepis) nebo anotace založené na existujících znalostech o hovoru (například velké množství provozovatelů call-center provádí vlastní klasifikaci hovorů). Celkově tak lze výsledný systém nasadit v kratším čase a s nižšími náklady.

Cíl 3: *Formulace plně pravděpodobnostního diskriminativního modelu*

Diskriminativní model z kapitoly 6 obsahující konceptový model reprezentovaný v této práci hierarchickým diskriminativním modelem, dále pak detekci sémantických entit a model zarovnání, je plně popsán pomocí pravděpodobnostní Bayesovské sítě na obrázku 6.1 a odtud vyplývající rovnice 6.3. Takto formulovaný diskriminativní model těží z pravděpodobnostní formulace dílčích modelů a z jejich schopnosti modelů generovat neurčitý,

pravděpodobnostmi oceněný výstup. Celkový diskriminativní model tvořený kombinací modelů dílčích byl experimentálně ověřen v kapitole 10.5.

V tomto diskriminativním modelu je možné použít i jiné modely, než popsané v této práci, například namísto modelu HDM lze použít model STC nebo místo detekce sémantických entit lze použít detektor klíčových slov. Dalším možnou modifikací je model zarovnání trénovaný z ručně označených, zarovnaných dat; například ze zarovnaných sémantických stromů. Struktura diskriminativního modelu však zůstává totožná.

Cíl 4: *Návrh vhodné metody pro kombinaci statistického a znalostního přístupu*

Diskriminativní model popsaný v kapitole 6 ustanovuje plně pravděpodobnostní rámec pro kombinaci statistického hierarchického diskriminativního modelu (kapitola 7) a na expertních znalostech založené detekce sémantických entit (kapitola 8).

Statistického přístupu je použito pro predikci abstraktního sémantického stromu, čili významu vyšší úrovně bez vazby na konkrétní lexikální realizaci. Sémantické entity pak reprezentují význam nižší úrovně. Význam vyšší úrovně je přiřazován globálně celé promluvě (větě), oproti tomu sémantické entity jsou striktně lokální, výskyt určité sémantické entity v nějakém časovém intervalu ve vstupní promluvě příliš neovlivňuje pravděpodobnost výskytu dalších entit v jiných časových intervalech.

Význam vyšší úrovně je odvozován statisticky, neboť mnoho sémantických konceptů nemá pouze jednu lexikální realizaci. Typickým představitelem takových konceptů jsou koncepty pro souhlas a nesouhlas. Na základě expertních pravidel je velice obtížné například určit význam promluvy *ne to je ono* (souhlas) nebo *ano to nechci* (nesouhlas). Při použití vhodného statistického modelu a dostatečně rozsáhlé a reprezentativní trénovací množiny však lze tato pravidla pro klasifikaci promluv automaticky odvodit. Jeden z možných statistických modelů je právě hierarchický diskriminativní model popsaný v kapitole 7.

Problémem statistického přístupu však zůstává dostatek trénovacích dat. Jak bylo ilustrováno v úvodu kapitoly 6, velké množství lexikálních realizací sémantických konceptů není v trénovacích datech vůbec obsaženo. Nicméně lexikální realizace mnoha sémantických konceptů lze získat buď z doménové databáze nebo přenosem z jiné úlohy. Příkladem budiž sémantický koncept STATION reprezentující stanice v korpusu HHTT. Z doménové databáze lze získat seznam všech stanic a pomocí poloautomatického přístupu lze tyto lexikální realizace získat i v mnoho dalších tvarech (gramatické pády, zkrácené tvary, slang). Relativně jednoduše přenositelné mezi doménami jsou pak například znalosti o různých obecných sémantických entitách jako je čas nebo datum. Proto byl v kapitole 8 popsán přístup umožňující na základě bezkontextových gramatik definovaných expertem v dané oblasti detekovat sémantické entity v neurčité vstupní promluvě reprezentované slovní mřížkou.

Kombinace obou přístupů je pak založena na použití modelu zarovnání (kapitola 6), který integruje znalost ze statistického konceptového modelu a znalostního modelu detekce sémantických entit. Model zarovnání je navržen jako relativně jednoduchý, na pravidlech založený model, nicméně jako jedno z možných budoucích rozšíření si lze představit i statistický model zarovnání. Jako celek je pak tato kombinace ověřena v kapitole 10.5. Křivky učení v kapitole 10.6 pak ukazují, že přínos expertní znalosti je tím výraznější, čím méně je pro danou úlohu dostupných dat. Avšak i při dostatku trénovacích dat kombinace znalostního a statistického přístupu přináší další zvýšení přesnosti porozumění.

Cíl 5: *Ověření modelu nad více cílovými doménami*

Výhodou všech popsaných modelů je jednoduché přenesení na novou cílovou doménu. V případě modelu HDM toto přenesení spočívá v pouhém přetrénování s novými trénovacími daty. Tato vlastnost však není samozřejmostí, jak je zřejmé z výsledků referenčního modelu HVS, který po přenesení z úlohy HHTT na úlohu TIA zaznamenal zřetelný propad v dosahované konceptové přesnosti.

Jak bylo ukázáno v kapitole 10.2, lze parametry ovlivňující strukturu hierarchického diskriminativního modelu (například prahy N a M, řád racionální jádrové funkce apod.) nastavit na jednom sémantickém korpusu a následně tytéž parametry použít i pro jiný sémantický korpus. Tento přenos byl ilustrován používáním referenčního korpusu HHTT pro vývoj modelu HDM a po určení výše zmíněných parametrů (tabulka 10.7) byl model natrénován nad korpusem TIA, přičemž dosahované přesnosti porozumění nad oběma korpusy jsou porovnatelné, jak se lze přesvědčit v kapitole 10.3.

Přenos modelu detekce sémantických entit na novou problémovou oblast spočívá v použití nové sady bezkontextových gramatik definujících sémantické entity. Některé z gramatik navíc mohou být sdíleny mezi více doménami. Díky tomu je usnadněn vývoj a údržba hlasových dialogových systémů. Navíc bezkontextové gramatiky jsou velice přirozeným způsobem pro zápis expertních znalostí, čímž je vývoj hlasového dialogu dále urychlen. Detailní vyhodnocení detekce sémantických entit jak nad korpusem HHTT, tak TIA, je provedeno v grafech na obrázcích 10.16, resp. 10.18.

11.2 Možné další směry výzkumu

Cíle vytyčené na začátku disertační práce byly splněny a jejich výsledky byly shrnuty v předchozích odstavcích. Uveďme ještě krátce další možné směry výzkumu:

Hlasové dialogové systémy

Popsaný model porozumění byl navrhován s ohledem na konkrétní použití v hlasových dialogových systémech. Bude použit jako modul porozumění řeči v hlasovém dialogovém systému pro podávání informací o odjezdech a příjezdech vlaků a v hlasovém dialogovém systému telefonní inteligentní asistentka, které jsou vyvíjené na Katedře kybernetiky Západočeské univerzity v Plzni.

Z dlouhodobějšího hlediska pak bude jistě zajímavé sledovat vývoj v oblasti inkrementálních hlasových dialogových systémů. Zde popsaný model porozumění by následně mohl přispět k vytvoření „spojitého" hlasového dialogového systému. Tento systém by sice stále pracoval v diskrétních časových okamžicích, ale tyto časové okamžiky by díky velice rychlému vyhodnocení v modulu porozumění mohly být velice krátké (řádově desetiny vteřiny). V možnostech takového hlasového dialogového systému by bylo sledování stavu hlasového dialogu v reálném čase a možnost okamžité reakce. V současných hlasových dialozích, které jsou založené na obrátkách (angl. turn), je nutné vždy vyčkat na konec promluvy uživatele hlasového dialogu. Pokud by však stav hlasového dialogu byl aktualizován průběžně již během uživatelovy promluvy, mohl by dialogový manažer reagovat na nastalé události prakticky okamžitě. Například v případě nejednoznačnosti by se mohl vbořit do uživatelovy promluvy a vyžádat si od něj okamžitou reakci (např. *Domluvte mi schůzku*

s panem Šmídem ... – Myslíte pana Šmída nebo pana Šmídla?). Stejně tak by bylo možné častěji a přirozeněji realizovat potvrzování v hlasovém dialogu – v případě přijetí libovolné informace by bylo možné ji s pozměněnou prozodií přečíst uživateli již během jeho promluvy (*Chtěl bych jet do Prahy ... – Ano, do Prahy – ... zítra ve tři*).

Zde popsaný hierarchický diskriminativní model umožňuje díky efektivní vstupní vrstvě velice časté vyhodnocování významu vstupní promluvy. Je možné jej volat pro každou částečnou mřížku na výstupu systému automatického rozpoznávání řeči. Navíc, oddělená skrytá a výstupní vrstva umožňuje zavést vnitřní dynamiku systému porozumění řeči – skrytá vrstva může pracovat jako systém bez vnitřní dynamiky a tudíž vytvářet posteriogram pro sémantické n-tice (srovnejte s metodou pro detekci klíčových slov popsanou v kapitole 3.5) a výstupní vrstva pak může predikovat výstupní významovou strukturu na základě příznaků získaných z tohoto posteriogramu.

Porozumění řeči a audiovizuální archivy

Kromě výzkumu na poli porozumění řeči jsou dalším z hlavních oborů autora této práce audiovizuální archivy a související metody (detekce klíčových slov, spoken term detection). Tyto dva rozdílné světy však pojí velice silné analogie. Zatímco při porozumění řeči je analyzována jediná promluva, v audiovizuálních archivech se často nachází obrovské množství promluv. Přesto je však možné se na obě úlohy dívat jako na úlohu vyhledávání informace – v úloze porozumění řeči je prohledávána jediná promluva, ale dotaz na různé významy může být extrémně komplikovaný; v úloze audiovizuálních archivů je pak dotaz zpravidla relativně jednoduchý, je nutné jej však realizovat nad velkým množstvím promluv. Obě úlohy jsou svázány relativně striktními požadavky na výpočetní náročnost, rychlý výpočet v obou případech pozitivně ovlivní použitelnost systémů.

Z tohoto úhlu pohledu se do budoucna zdá velice perspektivní možnost rychlého vyhledávání výskytu sémantických entit v audiovizuálních archivech. Vzhledem k tomu, že tyto archivy lze postavit na metodách využívajících výhod faktorového automatu, je doplnění těchto vyhledávacích metod o detekci sémantických entit triviálním a z pohledu praktických aplikací velice žádaným rozšířením.

Poloautomatická klasifikace a shlukování hovorů

V praxi jsou též velice často žádané metody pro automatické shlukování hovorů, popřípadě pro jejich klasifikaci do vybraných tříd. Tyto metody poptávají provozovatelé velkých call-center, kteří si od nich slibují především zvýšení kvality poskytovaných služeb. Zajímají se o automatické vytipování vhodných zákazníků, o automatickou detekci vulgárních promluv operátorů, o automatické směrování hovorů na operátory specialisty a o další podobné aplikace.

Všechny tyto úlohy mají společného jmenovatele, a to nedostatek trénovacích dat pro jazykový model nebo dokonce pro rozpoznávací slovník. Proto byly zkoumány metody umožňující využití fonémového rozpoznávače pro automatické přiřazení významu vstupní promluvě. Není však nutné promluvě pouze přiřazovat význam v podobě abstraktního sémantického stromu, promluvu lze označovat pomocí „štítků" označujících příslušnost do dané třídy, přičemž tříd může být celá řada (hovor týkající se fakturace, v hovoru se vyskytly vulgarity, v hovoru se zákazník zajímá o nové služby ...).

Zde lze s výhodou použít adaptované fonémové jazykové modely, které z pohledu porozumění řeči a klasifikace promluvy umožňují dosáhnout obdobné přesnosti jako fonémové jazykové modely trénované ze zarovnaných slovních přepisů. Přestože výsledky dosahované při porozumění z dat na slovní úrovni jsou vyšší, jsou vykoupeny nutností sestavit slovní jazykový model a rozpoznávací slovník. V praktických úlohách však toto může být velice problematické, neboť získání vhodných trénovacích dat nemusí být možné nebo data mohou mít velice dynamickou povahu, například v čase velice proměnný slovník. Potom lze s výhodou použít právě metody založené na použití fonémového rozpoznávače.

Pro poloautomatickou klasifikaci hovorů mohou být dostupná trénovací data získaná z interních databází provozovatelů call-center. Hovory realizované v takových centrech jsou zpravidla již v současné době ručně klasifikovány, přičemž klasifikace slouží pro pozdější sestavování statistik a analýz týkajících se provozu centra. Na základě získaných statistik je možné prakticky okamžitě natrénovat klasifikátor, který bude s využitím adaptovaného fonémového jazykového modelu provádět tuto klasifikaci do tříd naprosto automaticky.

Protože racionální jádrové funkce použité ve vstupní vrstvě hierarchického diskriminativního modelu nemusí sloužit pouze jako podpora SVM klasifikátorů, ale i v dalších úlohách strojového učení, lze si představit i automatické shlukování podobných hovorů pomocí shlukovacích metod založených na racionálních jádrových funkcích.

Literatura

[1] Alan M. Turing. "Computing Machinery and Intelligence". In: *Mind* 59.236 (1950), s. 433–460. ISSN: 00264423.

[2] Gokhan Tur a Renato De Mori. *Spoken language understanding: Systems for extracting semantic information from speech*. Hoboken, New Jersey: Wiley, 2011. ISBN: 978-0-470-68824-3.

[3] D. Jurafsky, J.H. Martin, A. Kehler, et al. *Speech and language processing*. New York: Prentice Hall, 2000. ISBN: 978-0-13-187321-6.

[4] Frederick Jelinek. "Continuous Speech Recognition by Statistical Methods". In: *Proceedings of IEEE* 64.4 (1976), s. 532–556. ISSN: 0018-9219. DOI: 10.1109/PROC.1976.10159.

[5] Steve Young. *Frederick Jelinek 1932 – 2010: The Pioneer of Speech Recognition Technology*. 2010.

[6] Mehryar Mohri, Fernando C. N. Pereira a Michael Riley. "Weighted automata in text and speech processing". In: *Proceedings of the 12th biennial European Conference on Artificial Intelligence*. Budapest: John Wiley a Sons, 1996, s. 46–50.

[7] Mehryar Mohri, Fernando C. N. Pereira a Michael Riley. "Weighted finite-state transducers in speech recognition". In: *Computer Speech & Language* 16.1 (2002), s. 69–88. DOI: 10.1006/csla.2001.0184.

[8] Mehryar Mohri. "Weighted automata algorithms". In: *Handbook of weighted automata* (2009).

[9] Jason Eisner. "Expectation semirings: Flexible EM for learning finite-state transducers". In: *Proceedings of the ESSLLI workshop on finite-state methods in NLP*. August. 2001, s. 1–5.

[10] Corinna Cortes a Patrick Haffner. "Rational kernels: Theory and algorithms". In: *The Journal of Machine Learning* 5 (2004), s. 1035–1062.

[11] Joseph Weizenbaum. "ELIZA—a computer program for the study of natural language communication between man and machine". In: *Communications of the ACM* 9.1 (led. 1966), s. 36–45. ISSN: 00010782. DOI: 10.1145/365153.365168.

[12] Thomas Hempel. *Usability of Speech Dialog Systems*. Berlin Heidelberg: Springer, 2008, s. 175. ISBN: 978-3-540-78342-8. DOI: 10.1007/978-3-540-78343-5.

[13] Roberto Pieraccini, Esther Levin a Wieland Eckert. "AMICA: The AT&T mixed initiative conversational architecture". In: *Proc. Eurospeech*. Sv. 97. Citeseer, 1997.

[14] Esther Levin, S. Narayanan, Roberto Pieraccini, et al. "The AT&T-DARPA Communicator mixed-initiative spoken dialog system". In: *Proc. of the International Conference of Spoken Langurage Processing,(Beijing, China)*. Citeseer, 2000, s. 122–125.

[15] Marilyn A. Walker, Rebecca Passonneau a Julie E. Boland. "Quantitative and qualitative evaluation of Darpa Communicator spoken dialogue systems". In: *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics - ACL '01*. 1. Morristown, NJ, USA: Association for Computational Linguistics, 2001, s. 515–522. DOI: 10.3115/1073012.1073078.

[16] Steve Young. "Talking to machines (statistically speaking)". In: *Proceedings of International Conference on Spoken Language Processing*. Denver, 2002, s. 9–16.

[17] Steve Young. "Still talking to machines (cognitively speaking)". In: *Keynote Interspeech*. 2. Chiba, Japan: International Speech Communication Association, 2010, s. 10.

[18] David Schlangen a Gabriel Skantze. "A general, abstract model of incremental dialogue processing". In: *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*. April. Athens, Greece: Association for Computational Linguistics, 2009, s. 710–718.

[19] Gabriel Skantze a David Schlangen. "Incremental dialogue processing in a micro-domain". In: *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*. April. Athens, Greece: Association for Computational Linguistics, 2009, s. 745–753.

[20] Kenji Sagae, Gwen Christian, David Devault, et al. "Towards Natural Language Understanding of Partial Speech Recognition Results in Dialogue Systems". In: *NAACL-Short '09 Proceedings of Human Language Technologies*. June. Boulder, Colorado: Association for Computational Linguistics, 2009, s. 53–56.

[21] Steve Young, Milica Gašić, Simon Keizer, et al. "The Hidden Information State model: A practical framework for POMDP-based spoken dialogue management". In: *Computer Speech & Language* 24.2 (dub. 2010), s. 150–174. ISSN: 08852308. DOI: 10.1016/j.csl.2009.04.001.

[22] Jason D. Williams a Steve Young. "Scaling POMDPs for Spoken Dialog Management". In: *IEEE Transactions on Audio, Speech and Language Processing* 15.7 (zář. 2007), s. 2116–2129. ISSN: 1558-7916. DOI: 10.1109/TASL.2007.902050.

[23] Blaise Thomson a Steve Young. "Bayesian update of dialogue state: A POMDP framework for spoken dialogue systems". In: *Computer Speech & Language* 24.4 (říj. 2010), s. 562–588. ISSN: 08852308. DOI: 10.1016/j.csl.2009.07.003.

[24] Blaise Thomson, Jost Schatzmann a Steve Young. "Bayesian update of dialogue state for robust dialogue systems". In: *IEEE International Conference on Acoustics, Speech and Signal Processing* (břez. 2008), s. 4937–4940. ISSN: 1520-6149. DOI: 10.1109/ICASSP.2008.4518765.

[25] Jason D. Williams a Steve Young. "Partially observable Markov decision processes for spoken dialog systems". In: *Computer Speech & Language* 21.2 (dub. 2007), s. 393–422. ISSN: 08852308. DOI: 10.1016/j.csl.2006.06.008.

[26] Chiori Hori, Kiyonori Ohtake, Teruhisa Misu, et al. "Weighted finite state transducer based statistical dialog management". In: *Automatic Speech Recognition & Understanding, 2009. ASRU 2009. IEEE Workshop on*. IEEE, 2009, s. 490–495. ISBN: 9781424454808.

[27] Yasuhiro Minami, Ryuichiro Higashinaka, Kohji Dohsaka, et al. "Trigram dialogue control using POMDPs". In: *Spoken Language Technology Workshop (SLT), 2010 IEEE*. Sv. 1. IEEE, 2010, s. 336–341. ISBN: 9781424479030.

[28] BH Juang a LR Rabiner. *Automatic speech recognition–A brief history of the technology development*. Tech. zpr. Rutgers University a the University of California, Santa Barbara, 2005, s. 1–24.

[29] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Ed. Michael Jordan, J. Kleinberg a B. Schölkopf. New York: Springer, 2007, s. 738. ISBN: 0387310738.

[30] J. Psutka, L. Müller, J. Matoušek, et al. *Mluvíme s počítačem česky*. Praha: Academia, 2006, s. 752. ISBN: 80-200-1309-1.

[31] Finn V. Jensen. *Bayesian Networks and Decision Graphs*. Ed. Michael Jordan, Steffen L. Lauritzen, Jerald F. Lawless, et al. New York: Springer, 2001, s. 268. ISBN: 0-387-95259-4.

[32] Josef Psutka, Pavel Ircing, Jan Hajič, et al. "Issues in annotation of the Czech spontaneous speech corpus in the MALACH project". In: *Proceedings of fourth international conference on language resources and evaluation*. Lisbon: European Language Resources Association, 2004, s. 607–610. ISBN: 2-9517408-1-6.

[33] Paul Mermelstein. "Distance measures for speech recognition, psychological and instrumental". In: *Pattern recognition and artificial intelligence* (1976), s. 374–388.

[34] Hynek Heřmanský. "Perceptual linar predictive (PLP) analysis of speech". In: *Journal of the Acoustical Society of America* 87.4 (1990), s. 1738–1752. DOI: 10.1121/1.399423.

[35] M. Oerder a Hermann Ney. "Word graphs: an efficient interface between continuous-speech recognition and language understanding". In: *IEEE International Conference on Acoustics Speech and Signal Processing*. Minneapolis: Ieee, 1993, 119–122 vol.2. ISBN: 0-7803-0946-4. DOI: 10.1109/ICASSP.1993.319246.

[36] Daniel Povey, Mirko Hannemann, Gilles Boulianne, et al. "Generating Exact Lattices in the WFST Framework". In: *IEEE International Conference on Acoustics Speech and Signal Processing*. Sv. 213850. 102. Kyoto, Japan: IEEE, 2012, s. 4213–4216. ISBN: 978-1-4673-0044-5. DOI: 10.1109/ICASSP.2012.6288848.

[37] Martha Larson a Stefan Eickeler. "Using Syllable-based Indexing Features and Language Models to improve German Spoken Document Retrieval". In: *Proceedings of EUROSPEECH*. ISCA, 2003, s. 1217–1220.

[38] Kai-Fu Lee a Hsiao-Wuen Hon. "Speaker-Independent Phone Recognition Using Hidden". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 37.11 (1989), s. 1641–1648. DOI: 10.1109/29.46546.

[39] Igor Szoke, Lukáš Burget, Jan Černocký, et al. "Sub-word modeling of out of vocabulary words in spoken term detection". In: *Spoken Language Technology Workshop*. 3. Goa: IEEE Comput. Soc. Press, 2008, s. 273–276. ISBN: 978-1-4244-3471-8. DOI: 10.1109/SLT.2008.4777893.

[40] Cyril Allauzen, Mehryar Mohri a Brian Roark. "Generalized algorithms for constructing statistical language models". In: *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - ACL '03*. Sv. 1. July. Morristown, NJ, USA: Association for Computational Linguistics, 2003, s. 40–47. DOI: 10.3115/1075096.1075102.

[41] Frank Wessel, Ralf Schluter, Klaus Macherey, et al. "Confidence measures for large vocabulary continuous speech recognition". In: *IEEE Transactions on Speech and Audio Processing* 9.3 (březen 2001), s. 288–298. ISSN: 10636676. DOI: 10.1109/89.906002.

[42] Lucie Skorkovská, Pavel Ircing, Aleš Pražák, et al. "Automatic Topic Identification for Large Scale Language Modeling Data Filtering". In: *Text, Speech and Dialogue* 6836 (2011), s. 64–71. ISSN: 0302-9743. DOI: 10.1007/978-3-642-23538-2_9.

[43] V. I. Levenshtein. "Binary Codes Capable of Correcting Deletions, Insertions and Reversals". In: *Doklady Akademii Nauk SSSR* 163.4 (1965), s. 845–848.

[44] Roberto Pieraccini, Evelyne Tzoukermann, Zakhar Gorelov, et al. "Progress report on the Chronus system: ATIS benchmark results". In: *Proceedings of the workshop on Speech and Natural Language*. Harriman, New York: Association for Computational Linguistics, 1992, s. 67–71. ISBN: 1-55860-272-0. DOI: 10.3115/1075527.1075543.

[45] Christian Raymond a Giuseppe Riccardi. "Generative and Discriminative Algorithms for Spoken Language Understanding". In: *Proceedings of Interspeech 2007*. International Speech Communication Association, 2007.

[46] Filip Jurčíček. "Statistical approach to the semantic analysis of spoken dialogues". Dis. University of West Bohemia, 2007.

[47] Yulan He a Steve Young. "Semantic processing using the Hidden Vector State model". In: *Computer Speech & Language* 19.1 (led. 2005), s. 85–106. ISSN: 08852308. DOI: 10.1016/j.csl.2004.03.001.

[48] Daniel Jurafsky, Chuck Wooters, Jonathan Segal, et al. "Using a stochastic context-free grammar as a language model for speech recognition". In: *IEEE International Conference on Acoustics Speech and Signal Processing*. Sv. 1. Detroit: IEEE, 1995, s. 189–192. ISBN: 0-7803-2431-5. DOI: 10.1109/ICASSP.1995.479396.

[49] Frederick Jelinek a John D. Lafferty. "Computation of the probability of initial substring generation by stochastic context-free grammars". In: *Computational Linguistics* 17.3 (1991), s. 315–323.

[50] S. McGlashan, D. C. Burnett, J. Carter, et al. *Voice Extensible Markup Language (VoiceXML) Version 2.0*. W3C Recommendation. Břez. 2004.

[51] A. Hunt a S. McGlashan. *Speech Recognition Grammar Specification Version 1.0*. W3C Recommendation. Břez. 2004.

[52] Tomáš Valenta a Luboš Šmídl. "Automatic switchboard operator". In: *Text, Speech and Dialogue* 6836 (2011), s. 57–63. ISSN: 0302-9743. DOI: 10.1007/978-3-642-23538-2_8.

[53] Tomáš Valenta, Jan Švec a Luboš Šmídl. "Spoken Dialogue System Design in 3 Weeks". In: *Text, Speech and Dialogue* 7499.IV (2012), s. 624–631. ISSN: 0302-9743. DOI: 10.1007/978-3-642-32790-2_76.

[54] Miloslav Konopík. "Hybrid Semantic Analysis". PhD. Thesis. Department of Informatics, University of West Bohemia, 2009, s. 107.

[55] Yulan He a Steve Young. "Hidden vector state model for hierarchical semantic parsing". In: *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003.1* 1 (2003), s. 268–271. DOI: 10.1109/ICASSP.2003.1198769.

[56] Charles T. Hemphill, John J. Godfrey a George R. Doddington. "The ATIS spoken language systems pilot corpus". In: *Proceedings of the workshop on Speech and Natural Language - HLT '90*. Morristown, NJ, USA: Association for Computational Linguistics, 1990, s. 96–101. DOI: 10.3115/116580.116613.

[57] Filip Jurčíček, Jiří Zahradil a Libor Jelínek. "A human-human train timetable dialogue corpus". In: *Proceedings of EUROSPEECH, Lisboa* (2005), s. 1525–1528.

[58] Jan Švec. "Sémantická analýza promluv systému NÁDRAŽÍ". Diplomová práce. Katedra kybernetiky, Západočeská univerzita v Plzni, 2007, s. 74.

[59] François Mairesse, Milica Gašić, Filip Jurčíček, et al. "Spoken language understanding from unaligned data using discriminative classification models". In: *IEEE International Conference on Acoustics, Speech and Signal Processing, 2009. ICASSP 2009.* Taipei: IEEE, 2009, s. 4749–4752. ISBN: 978-1-4244-2353-8. DOI: 10.1109/ICASSP.2009.4960692.

[60] Luke S. Zettlemoyer a Michael Collins. "Online learning of relaxed CCG grammars for parsing to logical form". In: *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. June. Citeseer, 2007, s. 678–687.

[61] Fabrice Lefevre, François Mairesse a Steve Young. "Cross-lingual spoken language understanding from unaligned data using discriminative classification models and machine translation". In: *Proceedings of Interspeech 2010*. International Speech Communication Association, 2010, s. 2–5.

[62] Eric Brill. "Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging". In: *Computational linguistics* 21.4 (1995), s. 543–565.

[63] Giorgio Satta a Eric Brill. "Efficient transformation-based parsing". In: *Proceedings of the 34th annual meeting on Association for Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics, 1996, s. 255–262. DOI: 10.3115/981863.981897.

[64] Ken Samuel, Sandra Carberry a K. Vijay-Shanker. "Dialogue act tagging with Transformation-Based Learning". In: *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics* 2 (1998), s. 1150–1156. DOI: 10.3115/980691.980757.

[65] Filip Jurčíček, Simon Keizer, François Mairesse, et al. "Transformation-based Learning for Semantic parsing". In: *Proceedings of Interspeech 2009*. International Speech Communication Association, 2009, s. 2719–2722.

[66] Radu Florian, JC Henderson a Grace Ngai. "Coaxing confidences from an old friend: probabilistic classifications from transformation rule lists". In: *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora*. Hong Kong: Association for Computational Linguistics, 2000, s. 26–34. DOI: 10.3115/1117794.1117798.

[67] Petr Fousek a Hynek Heřmanský. "Towards ASR Based on Hierarchical Posterior-Based Keyword Recognition". In: *IEEE International Conference on Acoustics Speech and Signal Processing*. Sv. 1. IEEE, 2006, ISBN: 1-4244-0469-X. DOI: 10.1109/ICASSP.2006.1660050.

[68] Cyril Allauzen, Mehryar Mohri a Murat Saraclar. "General indexation of weighted automata–application to spoken utterance retrieval". In: *HLT-NAACL 2004 Workshop: Interdisciplinary Approaches to Speech Indexing and Retrieval*. Ed. Bhuvana Ramabhadran a Oard Douglas. Boston, Massachusetts, USA: Association for Computational Linguistics, 2004, s. 33–40.

[69] Dogan Can a Murat Saraclar. "Lattice Indexing for Spoken Term Detection". In: *IEEE Transactions on Audio, Speech and Language Processing* 19.8 (2011), s. 2338–2347. DOI: 10.1109/TASL.2011.2134087.

[70] Mehryar Mohri, Pedro Moreno a Eugene Weinstein. "Factor automata of automata and applications". In: *Implementation and Application of Automata* 4783 (2007), s. 168–179. DOI: 10.1007/978-3-540-76336-9_17.

[71] Corinna Cortes a Vladimir Vapnik. "Support-Vector Networks". In: *Machine learning* 20.3 (1995), s. 273–297. DOI: 10.1007/BF00994018.

[72] Christopher CJ Burges. "A tutorial on support vector machines for pattern recognition". In: *Data mining and knowledge discovery* 2.2 (1998), s. 121–167. DOI: 10.1023/A:1009715923555.

[73] Chih-wei Hsu, Chih-chung Chang a Chih-jen Lin. *A Practical Guide to Support Vector Classification*. Tech. zpr. 1. 2010, s. 1–16.

[74] Maurizio Filippone, Francesco Camastra, Francesco Masulli, et al. "A survey of kernel and spectral methods for clustering". In: *Pattern Recognition* 41.1 (led. 2008), s. 176–190. ISSN: 00313203. DOI: 10.1016/j.patcog.2007.05.018.

[75] Bernhard Scholkopf a Alexander J. Smola. "Nonlinear Component Analysis as a Kernel Eigenvalue Problem". In: *Neural Computation* 10.5 (1998), s. 1299–1319. ISSN: 0899-7667. DOI: 10.1162/089976698300017467.

[76] Chih-wei Hsu a Chih-jen Lin. "A comparison of methods for multiclass support vector machines". In: *IEEE Transactions on Neural Networks* 13 (2002), s. 415–425.

[77] Leon Bottou, Corinna Cortes, John S. Denker, et al. "Comparison of classifier methods: a case study in handwritten digit recognition". In: *Proceedings of the 12th IAPR International Conference on Pattern Recognition (Cat. No.94CH3440-5)*. Sv. 2. IEEE Comput. Soc. Press, 1994, s. 77–82. ISBN: 0-8186-6270-0. DOI: 10.1109/ICPR.1994.576879.

[78] Ulrich H.-G. Kressel. "Advances in kernel methods". Ed. Bernhard Schölkopf, Christopher J. C. Burges a Alexander J. Smola. Cambridge, MA, USA: MIT Press, 1999. Kap. Pairwise classification and support vector machines, s. 255–268. ISBN: 0-262-19416-3. DOI: 10.1109/72.991427.

[79] John C. Platt a Nello Cristianini. "Large margin DAGs for multiclass classification". In: *Advances in neural* (2000), s. 547–553.

[80] Chih-Chung Chang a Chih-Jen Lin. "LIBSVM: A Library for Support Vector Machines". In: *ACM Transactions on Intelligent Systems and Technology* 2.3 (dub. 2011), s. 1–27. ISSN: 21576904. DOI: 10.1145/1961189.1961199.

[81] John C. Platt. *Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods*. Ed. A.J. Smola, P.L. Bartlett, B. Schölkopf, et al. Cambridge: MIT Press, 2000, s. 61–74.

[82] Hsuan-Tien Lin, Chih-Jen Lin a Ruby C. Weng. "A note on Platt's probabilistic outputs for support vector machines". In: *Machine Learning* 68.3 (srp. 2007), s. 267–276. ISSN: 0885-6125. DOI: 10.1007/s10994-007-5018-6.

[83] Ting-Fan Wu, Chih-jen Lin a Ruby C. Weng. "Probability estimates for multi-class classification by pairwise coupling". In: *The Journal of Machine Learning Research* 5 (2004), s. 975–1005.

[84] Arnulf B.A. Graf, Alexander J. Smola a Silvio Borer. "Classification in a normalized feature space using support vector machines." In: *IEEE Transactions on Neural Networks* 14.3 (led. 2003), s. 597–605. ISSN: 1045-9227. DOI: 10.1109/TNN.2003.811708.

[85] R Graepel. "A PAC-Bayesian margin bound for linear classifiers: Why SVMs work". In: *Advances in Neural Information Processing Systems*. Sv. 13. MIT Press, 2001, s. 224–230. ISBN: 0262122413.

[86] Mehryar Mohri, Pedro Moreno a Eugene Weinstein. "General suffix automaton construction algorithm and space bounds". In: *Theoretical Computer Science* 410.37 (zář. 2009), s. 3553–3562. ISSN: 03043975. DOI: 10.1016/j.tcs.2009.03.034.

[87] Graeme Blackwood. "Lattice rescoring methods for statistical machine translation". Dis. Cambridge University, 2010, s. 140.

[88] N. Friburger a D. Maurel. "Finite-state transducer cascades to extract named entities in texts". In: *Theoretical Computer Science* 313.1 (2004), s. 93–104. DOI: 10.1016/j.tcs.2003.10.007.

[89] Daniel Povey, Arnab Ghoshal, Nagendra Goel, et al. "The Kaldi speech recognition toolkit". In: *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. Big Island, Hawaii: IEEE, 2011.

[90] Libor Jelínek. "Porozumění telefonickému dotazu pro automatickou informační službu". Dis. Pilsen, Czech Republic: University of West Bohemia, Faculty of Applied Sciences, 2003, s. 141.

[91] Cyril Allauzen, Michael Riley a Johan Schalkwyk. "OpenFst: A general and efficient weighted finite-state transducer library". In: *Implementation and Application of Automata* 4783 (2007), s. 11–23. DOI: 10.1007/978-3-540-76336-9_3.

[92] Cyril Allauzen, Michael Riley a Johan Schalkwyk. "Filters for efficient composition of weighted finite-state transducers". In: *Implementation and Application of Automata* 6482 (2011), s. 28–38. DOI: 10.1007/978-3-642-18098-9_4.

[93] Mehryar Mohri. "Generic epsilon-removal and Input epsilon-normalization Algorithms for Weighted Transducers". In: *International Journal of Foundations of Computer Science* 13.1 (2002), s. 129–143. DOI: 10.1142/S0129054102000996.

[94] Mehryar Mohri. "Minimization algorithms for sequential transducers". In: *Theoretical Computer Science* 234.1-2 (2000), s. 177–201. DOI: 10.1016/S0304-3975(98)00115-7.

[95] Brian Roark, Richard Sproat, Cyril Allauzen, et al. "The OpenGrm open-source finite-state grammar software libraries". In: *Proceedings of the ACL 2012 System Demonstrations*. July. Jeju Island, Korea, 2012, s. 61–66.

[96] Mehryar Mohri. "Semiring frameworks and algorithms for shortest-distance problems". In: *Journal of Automata, Languages and Combinatorics* 7 (2002), s. 321–350.

[97] Christina Leslie, Jason Weston, Eleazar Eskin, et al. "Mismatch String Kernels for SVM Protein Classification". In: *Proceedings of NIPS*. Vancouver, Canada: MIT Press, 2002.

[98] David Haussler. *Convolution Kernels on Discrete Structures (UCSC-CRL-99-10)*. Tech. zpr. University of California in Santa Cruz, Computer Science Department, 1999, s. 1–38.

[99] K. Lari a Steve Young. "The estimation of stochastic context-free grammars using the Inside-Outside algorithm". In: *Computer Speech & Language* 4.1 (1990), s. 35–56. DOI: 10.1016/0885-2308(90)90022-X.

[100] Zhiyi Chi a Stuart Geman. "Estimation of Probabilistic Context-Free Grammars". In: *Computational Linguistics* 24.2 (1997), s. 299–305.

[101] Yves Schabes a RC Waters. "Lexicalized context-free grammars". In: *Proceedings of the 31st annual meeting on Association for Computational Linguistics*. Stroudsburg: Association for Computational Linguistics, 1993, s. 121–129. DOI: 10.3115/981574.981591.

[102] Miroslav Ressler. *Informační věda a knihovnictví: výkladový slovník české terminologie z oblasti informační vědy a knihovnictví : výběr z hesel v databázi TDKIV*. Praha: Vysoká škola chemicko-technologická v Praze a Národní knihovna České republiky, 2006. ISBN: 9788070805992.

[103] Eugene Charniak. "Immediate-Head Parsing for Language Models". In: *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*. 3. Toulouse, France: ACM, 2001, s. 124–131. DOI: 10.3115/1073012.1073029.

[104] Jason D. Williams a Steve Young. "Using Wizard-of-Oz simulations to bootstrap Reinforcement-Learning-based dialog management systems". In: *4th SIGdial Workshop on Discourse and Dialogue*. Sapporo, Japan, 2003.

[105] M Stuttle, Jason D. Williams a Steve Young. "A framework for dialogue data collection with a simulated ASR channel". In: *The 8th International Conference on Spoken Language Processing (ICSLP)*. Jeju Island, Korea, 2004, s. 241–244.

[106] Ian H. Witten a Timothy C. Bell. "The Zero-Frequency Problem: Estimating the Probabilities of Novel Events in Adaptive Text Compression". In: *IEEE Transactions on Information Theory* 37.4 (1991), s. 1085–1094. ISSN: 0018-9448. DOI: 10.1109/18.87000.

[107] Andreas Stolcke. "SRILM - An Extensible Language Modeling Toolkit". In: *Proceedings of International Conference on Spoken Language Processing* (2002), s. 901–904.

[108] Filip Jurčíček, Jan Švec a Luděk Müller. "Extension of HVS semantic parser by allowing left-right branching". In: *Proceedings of IEEE International Conference on Acoustics Speech and Signal Processing 2008*. 1. IEEE, 2008, s. 4993–4996. ISBN: 1424414849. DOI: 10.1109/ICASSP.2008.4518779.

[109] Deborah A. Dah, Madeleine Bates, Michael Brown, et al. "Expanding the scope of the ATIS task: The ATIS-3 corpus". In: *Proceedings of the workshop on Human Language Technology*. Stroudsburg, 1994, s. 43–48. ISBN: 1-55860-357-3. DOI: 10.3115/1075812.1075823.

[110] Jeff Bilmes. *GMTK: the graphical models toolkit*. Tech. zpr. 2002.

[111] Maxime Crochemore. "Transducers and Repetitions". In: *Theoretical Computer Science* 45 (1986), s. 63–86.

[112] Stuart J Russell a Peter Norvig. *Artificial Intelligence, A Modern Approach - 3rd Edition*. Ed. Michael Hirsch. Sv. 82. Prentice Hall series in artificial intelligence 1-2. Upper Saddle River, New Jersey: Prentice Hall, 2009, s. 1152. ISBN: 0136042597. DOI: 10.1016/0004-3702(96)00007-0.

[113] C. M. Bishop. *Neural networks for pattern recognition*. Oxford, USA: Oxford University Press, 1995. ISBN: 0-19-853864-2.

[114] Leo Breiman. "Random forests". In: *Machine learning* 45.1 (2001), s. 5–32. DOI: 10.1023/A:1010933404324.

[115] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learningč* 12 (led. 2012), s. 2825–2830. arXiv:1201.0490.

[116] Josef Psutka, Jan Švec, Josef V. Psutka, et al. "System for Fast Lexical and Phonetic Spoken Term Detection in a Czech Cultural Heritage Archive". In: *EURASIP Journal on Audio, Speech, and Music Processing* 2011.1 (2011), s. 10. ISSN: 1687-4722. DOI: 10.1186/1687-4722-2011-10.

[117] Ralph Grishman a Beth Sundheim. "Message understanding conference-6: A brief history". In: *Proceedings of COLING*. Copenhagen, Denmark: Association for Computational Linguistics, 1996, s. 466–471. DOI: 10.3115/992628.992709.

[118] Jana Kravalová a Zdeněk Žabokrtský. "Czech named entity corpus and SVM-based recognizer". In: *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration*. August. Association for Computational Linguistics, 2009, s. 194–201.

[119] Frédéric Béchet, Allen L Gorin, Jeremy H Wright, et al. "Detecting and extracting named entities from spontaneous speech in a mixed-initiative spoken dialogue context: How May I Help You?" In: *Speech Communication* 42.2 (ún. 2004), s. 207–225. ISSN: 01676393. DOI: 10.1016/j.specom.2003.07.003.

[120] Christian Raymond, Frédéric Béchet, Renato De Mori, et al. "On the use of finite state transducers for semantic interpretation". In: *Speech Communication* 48.3-4 (břez. 2006), s. 288–304. ISSN: 01676393. DOI: 10.1016/j.specom.2005.06.012.

[121] Christophe Servan, Christian Raymond, Frédéric Béchet, et al. "Conceptual decoding from word lattices: application to the spoken dialogue corpus media". In: *Proceedings of International Conference on Spoken Language Processing*. Pittsburgh: ISCA, 2006, s. 1–4.

[122] Dilek Hakkani-Tür, Frédéric Béchet, Giuseppe Riccardi, et al. "Beyond ASR 1-best: Using word confusion networks in spoken language understanding". In: *Computer Speech & Language* 20.4 (říj. 2006), s. 495–514. ISSN: 08852308. DOI: 10.1016/j.csl.2005.07.005.

[123] Mehryar Mohri a Fernando C. N. Pereira. "Dynamic compilation of weighted context-free grammars". In: *Proceedings of the 17th international conference on Computational linguistics*. Montreal, Quebec, Canada: Association for Computational Linguistics, 1998, s. 891–897. DOI: 10.3115/980432.980716.

[124] Fernando C. N. Pereira a Rebecca N. Wright. "Finite-state approximation of phrase structure grammars". In: *Proceedings of the 29th annual meeting on Association for Computational Linguistics ACL '91*. Berkeley: Association for Computational Linguistics, 1991, s. 246–255. DOI: 10.3115/981344.981376.

[125] Alan W. Black. "Finite state machines from feature grammars". In: *Finite State Machines from Feature Grammars*. Ed. Masaru Tomita. Pittsburgh: Carnegie Mellon University, 1989, s. 277–285.

[126] Michael Brown a Bruce Buntschuh. "A Context-free Grammar Compiler for Speech Understaing System". In: *Proceedings of 3rd International Conference on Spoken Language Processing (ICSLP 94)*. September. Yokohama: ISCA, 1994, s. 21–24.

[127] Marilyn A. Walker a Rebecca Passonneau. "DATE: a dialogue act tagging scheme for evaluation of spoken dialogue systems". In: *Proceedings of the first international conference on Human language technology research*. Stroudsburg, 2001, s. 1–8. DOI: 10.3115/1072133.1072148.

[128] Jindřich Matoušek, Daniel Tihelka a Luboš Šmídl. "On the Impact of Annotation Errors on Unit-Selection Speech Synthesis". In: *Text, Speech and Dialogue* 7499 (2012), s. 456–463. ISSN: 0302-9743. DOI: 10.1007/978-3-642-32790-2_55.

[129] Ondřej Koupil. "Jazykové modelování pro dialogové systémy". Diplomová práce. Katedra kybernetiky, Západočeská univerzita v Plzni, 2011, s. 61.

[130] Cyril Goutte a Eric Gaussier. "A probabilistic interpretation of precision, recall and F-score, with implication for evaluation". In: *Advances in Information Retrieval* 3408 (2005), s. 345–359. DOI: 10.1007/978-3-540-31865-1_25.

[131] J. Klavans, M. Liberman, M. Marcus, et al. "Procedure for quantitatively comparing the syntactic coverage of English grammars". In: *Proceedings of the workshop on Speech and Natural Language - HLT '91*. Ed. E. Black. Morristown, NJ, USA: Association for Computational Linguistics, 1991, s. 306–311. DOI: 10.3115/112405.112467.

[132] Michael Collins. "Three generative, lexicalised models for statistical parsing". In: *Proceedings of the 35th annual meeting on Association for Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics, 1997, s. 16–23. DOI: 10.3115/976909.979620.

[133] Kaizhong Zhang, Rick Statman a Dennis Shasha. "On the editing distance between unordered labeled trees". In: *Information Processing Letters* 42.3 (květ. 1992), s. 133–139. ISSN: 00200190. DOI: 10.1016/0020-0190(92)90136-J.

[134] Kaizhong Zhang a Dennis Shasha. "Simple Fast Algorithms for the Editing Distance between Trees and Related Problems". In: *SIAM Journal on Computing* 18.6 (pros. 1989), s. 1245–1262. ISSN: 0097-5397. DOI: 10.1137/0218082.

[135] Philip N Klein. "Computing the Edit-Distance Between Unrooted Ordered Trees". In: *Algorithms - ESA'98* 1461 (1998), s. 91–102. DOI: 10.1007/3-540-68530-8_8.

[136] Eiichi Tanaka a Keiko Tanaka. "The tree-to-tree editing problem". In: *Journal of pattern recognition and artificial intelligence* 2.2 (1988), s. 221–240. DOI: 10.1142/S0218001488000157.

[137] Kaizhong Zhang. "A new editing based distance between unordered labeled trees". In: *Combinatorial Pattern Matching* 684 (1993), s. 254–265. DOI: 10.1007/BFb0029810.

[138] Martin Emms. "Tree-distance and some other variants of evalb". In: *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*. Ed. Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, et al. Marrakech, Morocco: European Language Resources Association (ELRA), 2008. ISBN: 2-9517408-4-0.

[139] David Pallett, WM Fisher a JG Fiscus. "Tools for the analysis of benchmark speech recognition tests". In: *IEEE International Conference on Acoustics Speech and Signal Processing*. Albuquerque, NM: IEEE Comput. Soc. Press, 1990, s. 97–100. DOI: 10.1109/ICASSP.1990.115546.

[140] Juan Miguel Vilar. "Efficient computation of confidence intervals forword error rates". In: *IEEE International Conference on Acoustics, Speech and Signal Processing*. Las Vegas: IEEE, 2008, s. 5101–5104. ISBN: 1424414849. DOI: 10.1109/ICASSP.2008.4518806.

[141] M. Bisani a Hermann Ney. "Bootstrap estimates for confidence intervals in ASR performance evaluation". In: *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*. Sv. 1. IEEE, 2004, ISBN: 0-7803-8484-9. DOI: 10.1109/ICASSP.2004.1326009.

[142] S. Young, G. Evermann, D. Kershaw, et al. *The HTK book*. 1995.

[143] Peter E Brown, Vincent J Della Pietra, Robert L Mercer, et al. "An estimate of an upper bound for the entropy of English". In: *Computational Linguistics* 18.1 (1992), s. 31–40. ISSN: 0891-2017.

[144] Aleš Pražák, Zdeněk Loose, Jan Trmal, et al. "Novel Approach to Live Captioning Through Re-speaking: Tailoring Speech Recognition to Re-speaker's Needs". In: *Proceedings of Interspeech 2012*. Red Hook, 2012, s. 1370–1373. ISBN: 978-1-62276-759-5.

[145] Robert McGill, John W. Tukey a Wayne A. Larsen. "Variations of box plots". In: *The American Statistician* 32.1 (1978), s. 12–16. DOI: 10.2307/2683468.

[146] John D. Hunter. "Matplotlib: a 2D graphics environment". In: *Computing in Science & Engineering* 9.3 (2007), s. 90–95. DOI: 10.1109/MCSE.2007.55.

[147] Adam Chýlek. "Kombinace výstupů rozpoznávačů založených na odlišných principech". Diplomová práce. Katedra kybernetiky, Západočeská univerzita v Plzni, 2013, s. 50.

Seznam publikací

[A1] Jáchym Kolář, Jan Švec a Josef Psutka. "Automatic punctuation annotation in Czech broadcast news speech". In: *SPECOM´ 2004*. Saint-Petersburg: SPIIRAS, 2004, s. 319–325. ISBN: 5-7452-0110-X.

[A2] Jáchym Kolář, Jan Švec, Stephanie Strassel, et al. "Czech spontaneous speech corpus with structural metadata". In: *Proceedings of Interspeech 2005*. Lisbon: International Speech Communication Association, 2005, s. 1165–1168.

[A3] Filip Jurčíček, Jan Švec, Jiří Zahradil, et al. "Use of negative examples in training the HVS semantic model". In: *Text, Speech and Dialogue* 4188 (2006), s. 605–612. ISSN: 0302-9743. DOI: 10.1007/11846406_76.

[A4] Jan Švec, Filip Jurčíček a Luděk Müller. "Parameterization of the Input in Training the HVS Semantic Parser". In: *Text, Speech and Dialogue* 4629 (2007), s. 415–422. ISSN: 0302-9743. DOI: 10.1007/978-3-540-74628-7_54.

[A5] Jan Švec. "Sémantická analýza promluv systému NÁDRAŽÍ". Diplomová práce. Katedra kybernetiky, Západočeská univerzita v Plzni, 2007, s. 74.

[A6] Jáchym Kolář a Jan Švec. "Structural Metadata Annotation of Speech Corpora: Comparing Broadcast News and Broadcast Conversations". In: *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*. Sv. 94. Marrakech, Morocco: European Language Resources Association, 2008. ISBN: 2-9517408-4-0.

[A7] Filip Jurčíček, Jan Švec a Luděk Müller. "Extension of HVS semantic parser by allowing left-right branching". In: *Proceedings of IEEE International Conference on Acoustics Speech and Signal Processing 2008*. 1. IEEE, 2008, s. 4993–4996. ISBN: 1424414849. DOI: 10.1109/ICASSP.2008.4518779.

[A8] Aleš Pražák, Pavel Ircing, Jan Švec, et al. "Efficient combination of n-gram language models and recognition grammars in real-time LVCSR decoder". In: *Proceedings of 9th International Conference on Signal Processing, 2008*. Beijing: IEEE, 2008, s. 587–591. ISBN: 978-1-4244-2178-7. DOI: 10.1109/ICOSP.2008.4697201.

[A9] Jan Švec. "Hlasové dialogové systémy". Odborná práce ke státní doktorské zkoušce. Katedra kybernetiky, Západočeská univerzita v Plzni, 2009, s. 45.

[A10] Jan Švec a Jiří Zahradil. "BigList: Speech-based selection of items from huge lists". In: *Proceedings of the 9th WSEAS international conference on signal, speech and image processing*. Budapest: World Scientific, Engineering Academy, a Society, 2009, s. 57–62. ISBN: 978-960-474-114-4.

[A11] Jáchym Kolář a Jan Švec. "The Czech Broadcast Conversation Corpus". In: *Text, Speech and Dialogue* 5729 (2009), s. 101–108. ISSN: 0302-9743. DOI: 10.1007/978-3-642-04208-9_17.

[A12] Jan Švec a Filip Jurčíček. "Extended Hidden Vector State Parser". In: *Text, Speech and Dialogue* 5729 (2009), s. 403–410. ISSN: 0302-9743. DOI: 10.1007/978-3-642-04208-9_55.

[A13] Josef Psutka, Jan Švec, Josef V. Psutka, et al. "Fast Phonetic/Lexical Searching in the Archives of the Czech Holocaust Testimonies: Advancing Towards the MALACH Project Visions". In: *Text, Speech and Dialogue* 6231.III (2010), s. 385–391. ISSN: 0302-9743. DOI: 10.1007/978-3-642-15760-8_49.

[A14] Josef Psutka, Jan Švec, Josef V. Psutka, et al. "System for Fast Lexical and Phonetic Spoken Term Detection in a Czech Cultural Heritage Archive". In: *EURASIP Journal on Audio, Speech, and Music Processing* 2011.1 (2011), s. 10. ISSN: 1687-4722. DOI: 10.1186/1687-4722-2011-10.

[A15] Jan Švec a Luboš Šmídl. "Prototype of Czech Spoken Dialog System with Mixed Initiative for Railway Information Service". In: *Text, Speech and Dialogue* 6231.IV (2010), s. 568–575. ISSN: 0302-9743. DOI: 10.1007/978-3-642-15760-8_72.

[A16] Jan Švec, Jan Hoidekr a Daniel Soutner. "Web Text Data Mining for Building Large Scale Language Modelling Corpus". In: *Text, Speech and Dialogue* 6836 (2011), s. 356–363. ISSN: 0302-9743. DOI: 10.1007/978-3-642-23538-2_45.

[A17] Jan Švec a Luboš Šmídl. "Real-time large vocabulary spontaneous speech recognition for spoken dialog systems". In: *Proceedings of 4th International Congress on Image and Signal Processing, 2011*. Sv. 5. Shanghai: IEEE, říj. 2011, s. 2431–2436. ISBN: 978-1-4244-9306-7. DOI: 10.1109/CISP.2011.6100773.

[A18] Tomáš Valenta, Jan Švec a Luboš Šmídl. "Spoken Dialogue System Design in 3 Weeks". In: *Text, Speech and Dialogue* 7499.IV (2012), s. 624–631. ISSN: 0302-9743. DOI: 10.1007/978-3-642-32790-2_76.

[A19] Petr Stanislav a Jan Švec. "Unsupervised Synchronization of Hidden Subtitles with Audio Track Using Keyword Spotting Algorithm". In: *Text, Speech and Dialogue* 7499.III (2012), s. 422–430. ISSN: 0302-9743. DOI: 10.1007/978-3-642-32790-2_51.

[A20] Jan Švec a Pavel Ircing. "Efficient Algorithm for Rational Kernel Evaluation in Large Lattice Sets". In: *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, 2013*. Vancouver, Canada: IEEE, 2013, s. 3133–3137. ISBN: 978-1-4799-0355-9.

[A21] Jan Švec, Luboš Šmídl a Pavel Ircing. "Hierarchical Discriminative Model for Spoken Language Understanding". In: *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, 2013*. Vancouver, Canada: IEEE, 2013, s. 8322–8326. ISBN: 978-1-4799-0355-9.

[A22] Jan Lehečka a Jan Švec. "Improving Speech Recognition by Detecting Foreign Inclusions and Generating Pronunciations". In: *Text, Speech and Dialogue* (2013), (accepted for publication).

[A23] Jan Švec a Luboš Šmídl. "On the Use of Phoneme Lattices in Spoken Language Understanding". In: *Text, Speech and Dialogue* (2013), (accepted for publication).

[A24] Jan Vavruška, Jan Švec a Pavel Ircing. "Phonetic Spoken Term Detection in Large Audio Archive Using the WFST Framework". In: *Text, Speech and Dialogue* (2013), (accepted for publication).