VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

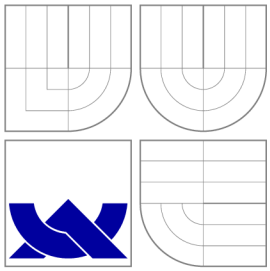
EXTRAKCE VZTAHŮ MEZI POJMENOVANÝMI ENTITAMI ZMÍNĚNÝMI V TEXTU

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

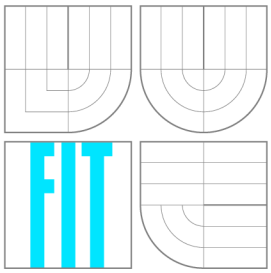
AUTOR PRÁCE
AUTHOR

ONDŘEJ VOHÁŇKA

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

EXTRAKCE VZTAHŮ MEZI POJMENOVANÝMI ENTITAMI ZMÍNĚNÝMI V TEXTU

EXTRACTION OF RELATIONS AMONG NAMED ENTITIES MENTIONED IN TEXT

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

ONDŘEJ VOHÁŇKA

VEDOUCÍ PRÁCE
SUPERVISOR

Doc. RNDr. PAVEL SMRŽ, Ph.D.

BRNO 2013

Abstrakt

Tato bakalářská práce se zabývá extrakcí vztahů. Vysvětluje základní znalosti nutné pro vývoj extrakčních systémů. Dále popisuje návrh, implementaci a srovnání tří vlastních systémů, které jsou řešeny jinými způsoby. Jsou použity metody jako regulární výrazy, NER a syntaktická analýza.

Abstract

This bachelor's thesis deals with relation extraction. Explains basic knowledge, that is necessary for creating an extraction system. Then describes design, implementation and comparison of three systems, which works differently. Following methods were used: regular expressions, NER, parser.

Klíčová slova

zpracování přirozeného jazyka, extrakce informací, extrakce vztahů, entita

Keywords

natural language processing, information extraction, relationship extraciton, entity

Citace

Ondřej Voháňka: Extrakce vztahů mezi pojmenovanými
entitami zmíněnými v textu, bakalářská práce, Brno, FIT VUT v Brně, 2013

Extrakce vztahů mezi pojmenovanými entitami zmíněnými v textu

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana doc. RNDr. Pavla Smrže, Ph.D.

.....
Ondřej Voháňka
14. května 2013

Poděkování

Děkuji doc. RNDr. Pavlu Smržovi, Ph.D. a Ing. Janu Kouřilovi za pomoc na konzultacích, Jakubovi Sznapkovi a Martinovi Šafářovi za pomoc s nástrojem NER a rodině a přátelům za podporu.

Obsah

Kapitola 1

Úvod

Extrakce vztahů je problematika, která spadá pod extrakci informací a využívá nástrojů pro zpracování přirozeného jazyka. Cílem je získat určené vztahy z přirozeného textu, kterému rozumí člověk. Počítač pro tento úkol potřebuje sadu speciálních nástrojů a postupů.

Vzhledem k rostoucímu objemu dat na internetu, vývoj extrakčních systémů nabývá na důležitosti. Vzniká obrovské množství dokumentů, článků a obecně textů. Sledování a ruční hledání informací zabírá příliž mnoho času. Proto přichází na řadu systémy schopné automaticky vyhledávat a získat dané informace z velkých kolekcí dat. Samostatnou kapitolu potom tvoří prohledávání internetu.

V textech můžeme najít pojmenované entity (např. člověk, lokace, organizace) a vztahy mezi nimi. Jsou různé druhy vztahů a různé způsoby extrakce. Často se ale setkáváme s nástroji jako name entity recognizer (NER), analyzátor syntaxe (parser), lexikální analyzátor (tokenizer), atd. Vztahy se objevují nejčastěji mezi entitami lidí, nebo mezi člověkem a organizací. Hojně jsou také hledány vztahy mezi z medicínského prostředí, kde vzniká velké množství odborných textů.

Tato práce popíše základní principy extrakce a vysvětlí důležité pojmy. Věnuje se i teorii vyhodnocování výsledků. Dále se můžeme dočíst o návrhu a implementaci tří různých extrakčních systémů. Nakonec uvidíme experimenty, výsledky a porovnání těchto systémů.

Kapitola 2

Extrakce vztahů mezi pojmenovanými entitami

Kapitola popisuje potřebný teoretický základ, který je nutný pro chápání dokumentů o extrakci. Vysvětluje základní pojmy, obsahuje krátký přehled obvyklých komponent systémů a nakonec nastíní metody pro vyhodnocování.

2.1 Základní pojmy extrakce informací

Abychom pochopili co znamená extrahovat vztahy, je dobré zařadit tuto disciplínu do většího celku a definovat některé pojmy. Obecně se totiž jedná o extrakci informací, což je proces, kterým můžeme získat chtěné informace z nějaké oblasti. To může zahrnovat hledání entit, relací a událostí.

Entita je objekt popsaný v textu představující osobu, organizaci nebo lokaci [7]. V textu bývají zastoupeny svými vlastostmi (i jméno je vlastnost objektu). Ty nejčastěji popisují jmenné fráze, např. *the Spanish Civil War*.

Relace (také vztah) můžeme definovat jako vztah nebo interakci mezi objekty zastoupenými jejich vlastnostmi, např. *Hank was married to Jennifer* [7]. Jeden objekt v takovém vztahu může být nahrazen i údaji jako datum, telefonní čislo, peněžní částka, atd. V tomto případě jsou chápany jako entity i přesto, že nereprezentují žádný objekt.

Událost odpovídá n-árním relacím. Mohli bychom ji popsat jako „kdo udělal co komu kdy a kde“ [3]. Snažíme se tedy zjistit co nejvíce informací o nějaké skutečnosti popsané v textu.

Pořadí těchto tří pojmů odpovídá komplikovanosti jejich extrakce. Je tomu mimojiné i proto, že každá relace se skládá z více entit a každá událost se skládá z více relací.

Dělení textů

Je třeba si uvědomit, že v rámci toho úkolu nejde o získávání informací z databází (tedy Data-mining). V nich jsou všechna data přehledně roztříděna a jejich interpretace je snadná díky nadpisům sloupců v tabulkách. Této skutečnosti si můžeme všimnout i v následujícím rozdělení textů, které popisuje J. R. Hobbs [3]:

Strukurovaný text Sémantika dat je daná jejich organizací (data v databázi).

Nestrukturovaný text Skládá se z vět přirozeného jazyka. Sémantiku je potřeba najít metodami pro zpracování a pochopení přirozeného jazyka a pro syntaktickou analýzu.

Semi-strukturovaný text Skládá se z vět přirozeného jazyka v dokumentu, kde jeho fyzické rozložení upravuje jeho interpretaci.

Mluvíme-li o extrakci vztahů, máme na mysli zpracování nestrukturovaných textů v přirozeném jazyce (tzn. články, knihy, popisky, atd.). To znamená, že proti datům v databázi máme k dispozici mnohem větší prostor a rozpětí pro hledané informace. Na druhou stranu je nutné počítat s tím, že počítač sám o sobě nerozumí jazyku lidí. A vzhledem ke komplikovanosti a variabilitě textů se to stává největším problémem této disciplíny. Potřebujeme tedy, abychom stroj naučili hledat v určitých souvislostech vyvozených z pravidel jazyka, se kterým pracujeme. Tento úkol se správně nazývá Zpracování přirozeného jazyka.

Zpracování přirozeného jazyka Zavedený a hojně používaný anglický název je *Natural Language Processing - NLP*. Volně přeložená definice z angličtiny zní: *NLP je teoreticky motivovaný soubor výpočetních technik pro analyzování a prezentaci přirozených textů na jedné, nebo více úrovních lingvistické analýzy za účelem dosáhnout lidské úrovně zpracování jazyka* [4]. Toto odvětví se tedy zabývá vztahem mezi počítači a lidskými jazyky. Jde o určité zpracování textu za účelem získání informace nebo předání dat dalším systémům. Většina podúkolů zahrnuje snahu o pochopení významu textu v přirozeném jazyce. V rámci tohoto odvětví se můžeme zabývat úkoly jako: NER (viz sekce 2.2.1), koreference (viz sekce 2.2.2), POS tagging, syntaktická analýza (viz sekce 2.2.3), rozlišení významu slov, atd.

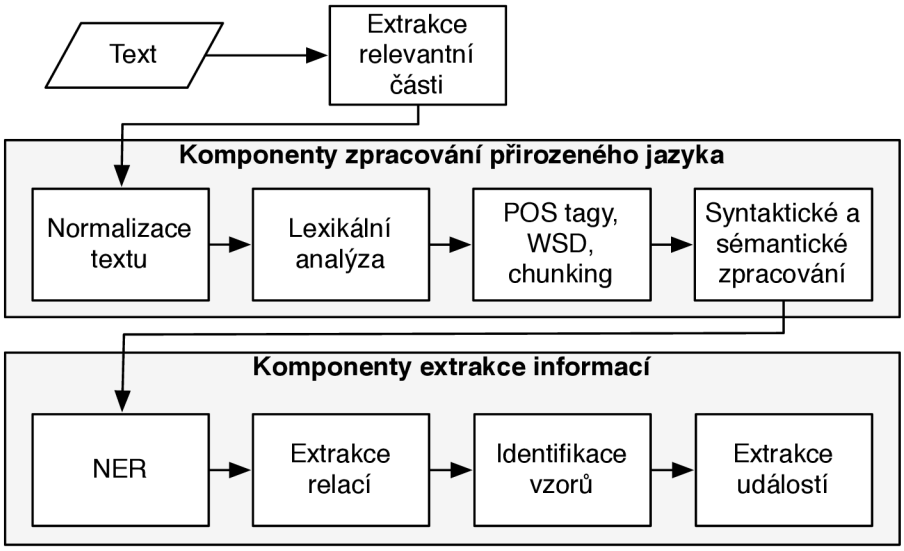
Přístup k dokumentům

V *Handbook of natural language processing* [3] se můžeme dočíst, že extrakční systémy jsou hojně používány vládou, armádou, v medicíně a pro nalezení specifických informací v textech na internetu. Potřeba těchto systémů je dobře znatelná například v medicíně, kde vychází více jako půl milionu článků ročně. Takové množství dokumentů je nutné prohledávat automaticky.

Přístupy k vývoji extrakčních systémů můžeme podle stejného zdroje dělit na:

Single-Document je přístup založený ja důležitosti každého jednoho dokumentu. A použijeme ho pokud potřebujeme mít přesný systém který extrahuje informaci z textu, i když je v něm obsažena pouze jednou. Pokud by to nezvládnul, je to považováno za chybu. Takový systém použijeme, když máme soubor dokumentů a v každém z nich se nachází důležitá informace, kterou nesmíme minout.

Multi-Document nelpí na důležitosti jednoho dokumentu a nevadí mu, pokud v něm požadovanou informaci nenajde. Operuje totiž nad velkým objemem dokumentů (nebo prohledává texty na internetu) a je pravděpodobné, že na stejnou informaci narazí v jiné formulaci a na jiném místě. Tyto systémy se začali vyvíjet hlavně kvůli rostoucí robustností internetu.



Obrázek 2.1: Ukázka klasického extrakčního systému

2.2 Komponenty extrakčních systémů

Systémy pro extrakci bývají většinou tvořeny několika nástroji řetězenými za sebe. To je výhodné, protože můžeme přistupovat individuálně ke každému, nebo využít jeden existující a zbytek vytvořit sami. Extrakce informací je postupný proces a každý nástroj nám umožní se přiblížit se k tomuto cíli. Každý funguje samostatně a přidává důležité informace, které mohou použít ty následující. Můžeme je rozdělit do dvou skupin podle toho, na jaké úrovni pracují: nástroje pro zpracování přirozeného jazyka a nástroje pro extrakci informací. První skupina většinou předchází té druhé, protože připraví text a přidá k němu užitečná metadata. Tyto návaznosti a dělení do skupin ilustruje obrázek 2.1. Rozdělení je ale pouze orientační, není pevně dané. V této sekci budou popsány obvykle řešené problémy a nástroje pro jejich řešení.

2.2.1 Identifikace entit v textu

Tento problém řeší nástroj *Name entity recognition (NER)*, který můžeme volně přeložit jako rozpoznávání jmenných entit. Cílem je identifikace částí textu, které zastupují nějakou entitu. Nejčastěji se jedná o jméno nebo název. Entity můžeme třídit do několika skupin, např. lidé, organizace, lokace, knihy, filmy, atd. NER bývá klíčový pro extrakci informací, protože se v řetězci nástrojů většinou nachází na začátku a ostatní nástroje jsou závislé na jeho výstupu. Horší NER tedy může negativně ovlivnit výsledky celého systému. Někdy je potřeba přidat k NERu klasifikátor, který umožňuje přiřadit identifikovanou entitu k externímu zdroji, kde můžeme najít definici (např. Wikipedia). Takový proces potom nazýváme named entity recognition and classification (NERC). Klasifikace entit alespoň na základní úrovni je potřebná, protože několik různých entit může mít stejný název (např. několik lidí se stejným jménem).

Pro řešení tohoto problému se používá široká škála přístupů: ručně psané vzory, metody strojového učení - s učitelem, bez učitele, velké vyhledávací tabulky, atd. Další informace jsou popsány například v *state of the art of event detection methods* [7].

2.2.2 Koreference

Každý systém, který se snaží extrahovat entity, by měl provést i analýzu koreference. Název vychází z reference, kterou obecně chápeme jako odkaz na něco. V kontextu extrakce informací se většinou jedná o odkaz na entitu. Koreference využívá jazykových prostředků, jako jsou zájmena nebo synonyma, pro odkazování na entitu zmíněnou na jiném místě. Pro ilustraci principu poslouží jednoduchý příklad: Ve větě *„John told us about his childhood.“* jsme nástrojem NER identifikovali entitu se jménem *John*, a pomocí analýzy koreference zjistíme, že zájmeno *his* na tuto entitu odkazuje. Takže je jasné, že *John* ve skutečnosti mluví o *svém* dětství. Vzhledem k hojnému používání takových zájmen a podobných prostředků je úspěšná analýza tohoto jevu klíčová pro extrakci informací. Pokud bychom na koreferenci nebrali zřetel, mohli bychom nalézt vztahy jenom ve větách, které vždy zmiňují entitu přímo (např. *„Rembrandt's work was influenced by the death of Rembrandt's wife“*).

2.2.3 Syntaktická analýza

Častěji se asi setkáme s anglickým názvem *parsing*. Obecně jde o analýzu řetězce podle nějakých pravidel (např. gramatiky). Tomuto procesu často předchází lexikální analýza (neboli *tokenization*), která řetězec přetvoří na posloupnost elementárních částic (tokenů). Těmi můžou být například i slova. Cílem syntaktické analýzy je získat strom závislostí mezi jednotlivými částicemi. V rámci extrakce informací si tento výstup můžeme představit jako strom složený ze slov, kde každá hrana popisuje druh závislosti mezi uzly, které spojuje. To je vidět například na obrázku 3.2. Takový výsledek můžeme dále zpracovat a využít pro sémantickou analýzu.

2.2.4 Označení slovních druhů

Někdy můžeme využít i možnosti označkovat si daný text slovními druhy. Nejčastěji takový nástroj nazýváme anglickým názvem *Part-of-Speech Tagger*, což můžeme přeložit jako značkovač slovních druhů. Tento nástroj se věnuje každému slovu zvlášť a spadá tak spíše pod lexikální analýzu. Označkovaný text pak může vypadat například takto: *The-AT representative-NN put-VBD chairs-NNS on-IN the-AT table-NN.* (příklad byl převzat z *Foundations of statistical natural language processing* [6]). Značení se může lišit podle použitého značkovacího standardu.

2.2.5 Rozlišení významu slov

Častěji se setkáme s anglickým překladem *Word sense disambiguation (WSD)*. Můžeme se setkat se situací, kdy zkoumané slovo může mít několik významů. A pokud ho vytrhneme z kontextu, nejsme schopni určit, který je ten správný. Například u slova *tree* na první pohled nemůžeme říci, zda se jedná o rostlinu, nebo druh grafu. Pro řešení tohoto problému se může použít označení slovních druhů. Těmi můžeme rozlišit například podstatné jméno od slovesa (např. slovo *work*) a rozlišit tak význam slova ve větě. V ostatních případech se většinou používá strojového učení a začlenění kontextu.

2.3 Vyhodnocování výsledků

Pro každý extrakční systém je vhodné provést určité vyhodnocení. Zjistíme tím dodatečné informace a míru korektnosti výsledků. Pro tento účel se můžeme obrátit na statistickou

Tabulka 2.1: Rozdělení získaných dat do tabulky

	Extrahováno	Neextrahováno
Relevatní	re (pravdivě pozitivní)	rn (falešně negativní)
Nerelevatní	ne (falešně pozitivní)	nn (pravdivě negativní)

vědu. Ta nám umožňuje získat určité poznání pozorováním a experimenty. Anna Gerylovová a Jan Holčík [2] píší, že se statistika využívá především při následujících činnostech:

Plánování a vlastní příprava statistického šetření Je dobré vycházet ze znalosti statistických metod, formulace pracovních hypotéz a odhad důledků, které mohou vyplynout.

Sběr dat Jde o přípravu a realizaci poměrně komplikovaného procesu, který by měl by odpovídat cíli statistického šetření, měl by se orientovat na chodné objekty a jejich podstatné vlastnosti.

Zpracování dat a technika jejich prezentace Data jsou obvykle protříděna a výsledky jsou prezentovány formou grafů a tabulek. Můžeme z nich vypočítat i další vhodné ukazatele.

Analýza dat a závěry Získaná a zpracovaná data mohou sloužit k odhadům vlastností zkoumaných objektů a k ověřování hypotéz.

Tomuto obecnému postupu by se měl podobat i postup při hodnocení extrakčního systému. Je tedy třeba začít tím, že si ujasníme cíle a naplánujeme šetření. Musíme si také připravit sadu textů pro experiment. Pro každý text musíme rozhodnout, zda je relevantní (obsahuje hledaný vztah) nebo není. Sběr dat provádíme ukládáním výsledků systému. Získaná data je vhodné přehledně zapsat do kontingenční tabulky jako v *Introduction to information retrieval* [5]. Může vypadat například jako tabulka 2.1. Na základě hodnot *re*, *ne*, *rn* a *nn* je možné vypočítat další údaje popisující systém. Jsou jimi dvojice přesnost/úplnost a správnost/chybovost. Získáme je pomocí následujících vztahů:

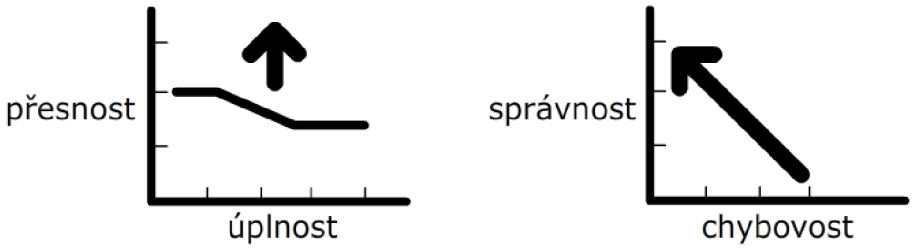
$$Presnost = \frac{re}{(re + ne)}$$

$$Uplnost = \frac{re}{(re + rn)}$$

$$Spravnost = \frac{re + nn}{(re + ne + rn + nn)}$$

$$Chybovost = \frac{ne + rn}{(re + ne + rn + nn)}$$

Přesnost určuje zlomek extrahovaných textů, které jsou relevantní. Úplnost nám naopak ukazuje zlomek relevantních textů, které jsou extrahovány. Tyto hodnoty jsou provázány.



Obrázek 2.2: Ideální směr vývoje extrakčních systémů

Manning [5] říká, že v dobrém systému přesnost klesá se zvyšující se úplností nebo s narůstajícím počtem extrahovaných textů. U vývoje extrakčních systémů se většinou snažíme o co možná nejvyšší přesnost. Čím vyšší bude, tím větší je pravděpodobnost, že výsledky budou bezchybné. Někdy ale můžeme ocenit i přesnost menší, protože implikuje větší hodnotu úplnosti. To oceníme hlavně, pokud je zadaná oblast extrakce menší a relevantních textů není mnoho.

Správnost je hodnota popisující podíl správného chování systému. Chybovost představuje pravý opak, tedy podíl chybného chování.

Tyto hodnoty můžeme nanést do grafu pro lepší prezentaci. Správnost a chybovost jsou propojené, takže jejich graf by vypadal jako úsečka, která propojuje hodnoty 1 a 1. Proto se běžne sestavuje hlavně graf přesnost/úplnost. V něm se snažíme nanést hodnoty pro všechny stupně úplnosti. Toho můžeme docílit tak, že si připravíme testovací data obsahující míru relevance. A při posunování prahové hodnoty dostaneme širší škálu hodnot úplnosti s odpovídajícími hodnotami přesnosti. Běžný vzhled grafů ilustruje obrázek 2.2. Šipky ukazují ideální postup při vývoji extrakčních systémů. Snažíme se tedy o co nejvyšší přesnost a správnost.

Pro zhodnocení kvalit systému se též používá tzv. F-measure. Jde o hodnotu zahrnující ve vztahu přesnost i úplnost. Můžeme ji vypočítat podle následujících vztahů:

$$F = \frac{1}{\alpha\frac{1}{P} + (1-\alpha))\frac{1}{R}} = \frac{(\beta^2+1)PR}{\beta^2 P + R}$$

Hodnoty α a β se určují podle důležitosti přesnosti a úplnosti. Klasicky ale volíme $\beta = 1$ a $\alpha = \frac{1}{2}$.

Kapitola 3

Návrh a realizace vlastního řešení

Kapitola obsahuje popis návrhu vlastního řešení, po kterém následuje sekce o tom, jak probíhala realizace. Návrh je ilustrován schématem. Každý ze tří vyvíjených systémů je popsán samostatně a na konci najdeme i výčet použitých nástrojů.

3.1 Návrh řešení

Cílem je vytvořit systém pro extrakci vztahů mezi entitami v textu. Je tedy potřeba zvážit, které komponenty zvolit. A také rozvrhnout, jak bude systém fungovat.

3.1.1 Požadavky na systém

Systém by měl být schopen získat údaje o vztahu z textu v přirozeném jazyce. Konkrétně se jedná o ovlivnění umělce (malíře) určitou událostí. K tomu je nutné najít slova, která ho reprezentují v textu, identifikovat událost a nakonec relaci mezi nimi.

3.1.2 Příprava

Pro implementaci některých částí extrakčních systémů je potřeba znát tvary vět ze zadané oblasti. Je tedy třeba určit několik konkrétních událostí, které budeme hledat. Pro tento projekt byly zvoleny tyto události:

- Smrt blízkého člověka
- Válka
- Cestování

Před samotnou implementací tedy proběhlo ruční vyhledávání vět s těmito klíčovými slovy. Každý nalezený tvar věty, nebo jiný způsob popisu události byl zaznamenán pro pozdější využití. Věty obsahující vztahy s těmito událostmi bývají většinou v podobných tvarech. Můžeme tedy zobecnit postupy a měnit se bude pouze název události.

3.1.3 Konceptuální model

Abychom mohli vyhodnotit několik různě složitých systémů, musíme je sestrojit a provést experimenty. Každý ze systémů se bude odlišovat metodami extrakce a výsledky. Je vhodné porovnat alespoň tři systémy. Budou na jiných úrovních složitosti. Ty jsou blíže popsány

v sekci 3.2. První a nejjednodušší pracuje pouze s regulárními výrazy a nástrojem NER. Středně pokročilý systém navazuje na informace zjištěné jednodušším systémem. Kontroluje správné pořadí prvků ve větě. Třetí z nich využívá parser pro získání stromu závislostí. Vzhledem k časové náročnosti tohoto nástroje nejdříve použije první systém pro filtraci vstupních textů. Bez ní by experimenty trvaly příliž dlouho. Další informace o tomto jevu jsou popsány v sekci 3.3.3. Vzhledem k těmto závislostem je vhodné sloučit extrakční systémy do jednoho většího programu. V něm budou logicky odděleny jednotlivé části.

Obrázek 3.1 představuje celý program zastřešující tři systémy. Ty jsou reprezentovány šedým pozadím. Můžeme si všimnout rozdělení na podúlohy a také vzájemných návazností. Tok dat je shora dolů. Vstupem je text v přirozeném jazyce. Výstupem by měla být relace mezi entitami obsaženými v textu. Kosočtverec značí data v určité formě, kdežto obdelníkový tvar představuje činnost nebo transformaci dat.

3.2 Úrovně složitosti

Tato sekce popíše postup při implementaci systémů na několika úrovních složitosti. Jazykem pro tento úkol se stal Python. A to díky jeho schopnosti jednoduše propojovat nástroje a pro snadnou práci se seznamy. Další výhodou byla skutečnost, že některé použité nástroje již byly napsány v Pythonu.

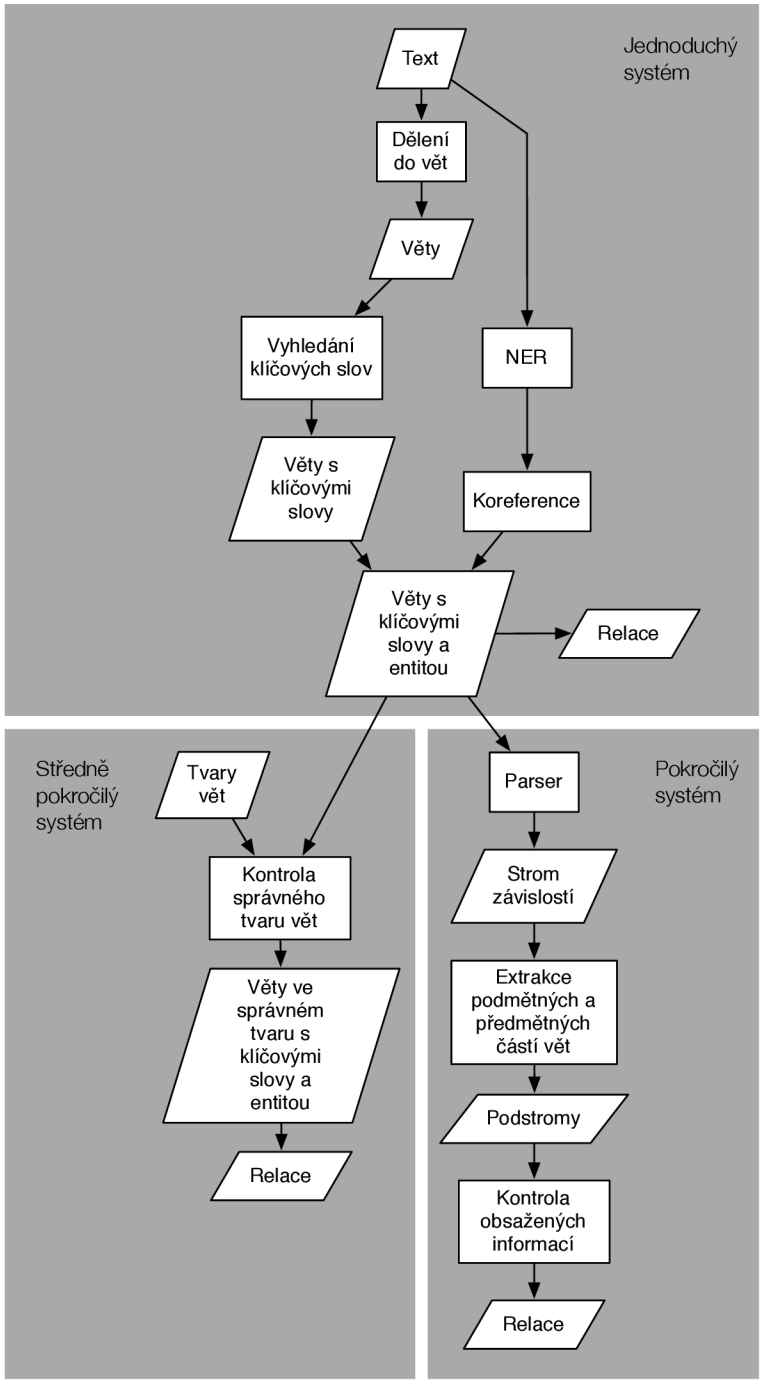
3.2.1 Nejjednodušší systém

Tento systém bude pracovat na nejnižší úrovni. Jeho složitost bude primitivní, tím pádem nebude moc přesný. Soustřeďuje se především na vyhledávání klíčových prvků. Těmi jsou:

- Spojení vyjadřující ovlivnění (např. was influenced by, had an impact)
- Slova označující událost, která má vliv na dílo umělce (např. the death of his father, World War I.)
- Slova pro vyjádření cíle ovlivnění (např. his painting, her work)
- Entita představující umělce. Ta může být obsažena ve jméně nebo v zájmeně. (např. Picasso, he, his)

Pro svojí funkcionalitu využívá předpokladu, který říká, že pouhá existence těchto prvků ve větě znamená výskyt relace. Reálně tento výrok pravdivý není. Standardně platí, že pokud věta obsahuje vztah, najdeme v ní i klíčové prvky. Výsledky tohoto systému tedy budou pouze částečně správné. Může se stát, že vztah označený systémem se ve větě vlastně vůbec nenachází. Vyhledávání probíhá za pomoci regulárních výrazů. Jedinou výjimku tvoří entita představující umělce. Kvůli ní je použito nástroje Decipher NER, který umí rychle vyhledávat v seznamech jmen a názvů a označuje jejich výskyty v textu. Dále pracuje s koreferencí, bez které by se počet kladných výstupů značně snížil. Více o tomto nástroji je psáno v sekci 3.3.2.

Téměř všechny operace je výhodné provádět nad jednotlivými větami. Proto je třeba text spolehlivě rozdělit do vět. Pro tento účel jsem využil Python knihovnu NLTK, která disponuje metodou `tokenize(text)`.



Obrázek 3.1: Konceptuální model systému pro extrakci vztahů

3.2.2 Středně pokročilý systém

Tento systém využívá předchozí systém a navazuje na něj. Snaží se o větší přesnost určování vztahů. Princip získávání informací pomocí regulárních výrazů zůstává, ale po dokončení vyhledávání je provedena kontrola. Ta se soustřeďuje na správné pořadí klíčových prvků ve větě.

Kontrola pracuje se vzory. Každý vzor je definován pomocí regulárního výrazu, který se snažíme najít ve větě (např. `'had .*?influence'`). Dále obsahuje řetězec reprezentující očekávané pořadí prvků, které kontrolujeme po nalezení vyhovujícího regulárního výrazu. Ve větě *„Trip to America had severe influence on his work.“* tedy dovedeme poznat již zmíněný vzor. A díky tomu víme, že na začátku věty by se měla nacházet slova označující událost (*Trip to America*). Na konci věty bychom měli najít slova pro vyjádření cíle ovlivnění (*his work*). Využijeme znalost indexů těchto prvků, kterou máme díky prvnímu systému, a vyhodnotíme, zda pořadí opravdu odpovídá vzoru. Pokud ano, kontrola proběhla úspěšně a můžeme dále tvrdit, že událost opravdu ovlivnila dílo daného umělce.

3.2.3 Pokročilý systém

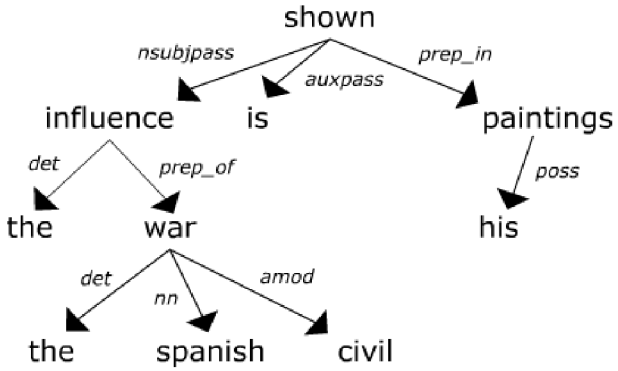
V obou předchozích systémech jsme pouze vyhledávali slova ve větách. Pokročilý systém pro extrakci by měl umět pracovat se syntaxí. Měl by vracet přesné výsledky a nespoléhat se na pravděpodobnost. Přesně o to se tento systém snaží. Pro zavedení syntaktické kontroly je třeba použít parser. Vhodným nástrojem je Stanford Parser (viz sekce 3.3.3), který umí vyhodnotit závislosti mezi slovy (Stanford Dependencies). Za ideálních okolností by tento systém mohl fungovat zcela samostatně. Kvůli časové složitosti parseru je ale nutné vyfiltrovat vstupy pouze na texty, které potenciálně obsahují vztah. K tomu slouží vyhledávání prvků v prvním systému. Je zde tedy nutná návaznost (viz obrázek 3.1).

Použitím Stanford Parseru získáme strom závislostí (viz obrázek 3.2). Můžeme si všimnout, že se jedná o závislosti mezi jednotlivými slovy. Ale naším cílem je najít vztah mezi většími celky. Je tedy nutné jednotlivá slova shlukovat. Podle Ani Thomas [8] je vhodné volit následující postup pro extrakci podmětných a předmětných částí:

1. Odstranění závislostí *det*, případně *predet* (a, an, the, some, all, atd.)
2. Tvorba jmenných frází. To znamená spojení závislostí *nn*, *amod* a *advmod*. Tím vznikne podstatné jméno se všemi modifikátory (např. *„individual computational elements“*).
3. Identifikace podmětné a předmětné části. Podmětnou najdeme pomocí závislotí *nsubj*, *nsubjpass* a *rcmod* a předmětnou pomocí *dobj* a *pobj*.

Praktikováním tohoto postupu se značně sníží počet uzlů stromu. Získání obou částí věty tímto způsobem je velmi užitečné pro extrakci. Nevadí nám totiž větší syntaktická složitost textu. Závislosti představují vztahy spíše na gramatické úrovni. Systém tedy extrahoval podmětnou, předmětnou a přísudečnou část. Následovně se v nich se pokusí regulárními výrazy najít klíčové prvky vztahu. Pokud je najde na správných místech, můžeme považovat za to, že věta nese informaci o vztahu.

Nakonec je provedena detekce negace, která by měla odhalit zápor ve větě. Například pokud před slovem *influenced* najdeme slovo *not*, sémanticky se z něj stane pravý opak a vztah už nemůžeme považovat za platný. Tato kontrola probíhá také nad stromem závislostí, takže je snadno dohledatelné, ke kterému slovesu se negace vztahuje.



Obrázek 3.2: Ukázka Stanford Dependencies

Takový systém je přesnější než oba dva předchozí. Navíc je, díky tvorbě jmenných frází, schopen lépe pojmenovat událost.

3.3 Integrace existujících nástrojů

Pro některé podproblémy projektu již existují nástroje, které se jim věnují. Proto je vhodné je použít a navázat na jejich výstupy. V této sekci jsou popsány ty nejdůležitější.

3.3.1 Python knihovna NLTK

Natural Language Toolkit je knihovna umožňující práci s přirozeným textem jazyce Python. Je to open source projekt vyvíjený komunitou. NLTK zastřešuje nejrůznější nástroje: POS tagger, NER, tokenizer, atd. V tomto projektu pro extrakci je ovšem použit jenom tokenizer pro oddělení jednotlivých vět.

3.3.2 Decipher NER

Decipher NER je nástroj vyvíjený výzkumnou skupinou KNOT na VUT na Fakultě Informatiky. Je použitelný pro vyhledávání umělců, stylů a děl v textu. Používá seznam vzniklý spojením dat z několika zdrojů: Freebase, Artcyclopedia, SCoT a další. Umožňuje také analýzu koreference. V tomto projektu je použit právě pro označení umělců a uměleckých děl.

3.3.3 Stanford Parser

Stanford Parser je nástroj, který umožňuje najít gramatickou strukturu vět. Je implementován v Javě, ale výstup lze jednoduše transformovat pro použití v Pythonu. Jeho výstupem jsou tzv. Stanford Dependencies (viz obrázek 3.2). Autoři nástroje o nich v manuálu [1] píší, že tyto závislosti byly vytvořeny pro jednoduchý popis gramatických vztahů ve větě tak, aby byly pochopitelné i pro lidi bez lingvistické praxe. Jsou tedy dostatečně triviální pro pozdější zpracování jinými nástroji.

Vzhledem ke své důkladnosti a kvalitním výsledkům trpí tento parser větší časovou náročností. To je potřeba zvážit při jeho začleňování do většího celku. Pokud bychom ho totiž nechali zpracovávat všechny vstupní texty, délka chodu programu by se stala neúnosnou. Je tedy potřeba zavést určitou kontrolu a filtraci textů, které přijdou na vstup tohoto nástroje.

3.3.4 Elastic Search a PyES

Tyto dva nástroje byly použity pro přístup se kolekcím dat nasbíraným v rámci projektu Decipher. Sloužili jako vstup velkého objemu dat do extrakčních systémů. PyES je knihovna jazyka Python umožňující práci s ElasticSearch a tím vyhledávání v zaindexovaných datech.

Kapitola 4

Experimentální výsledky a jejich analýza

Tato kapitola se zabývá experimentálními výsledky třech vytvořených systémů. Nejprve jsou popsány použitá data, poté popis experimentů a nakonec zhodnocení získaných informací.

4.1 Charakteristika datové sady

Pro vyhodnocení kvalit extrakčního systému jsou potřeba data, která by mohl zpracovat. Je dobré se zamyslet nad tím, jaká data zvolit. Mělo by se jednat o koleci dokumentů, ve kterých se může, nebo také nemusí vyskytovat vztah.

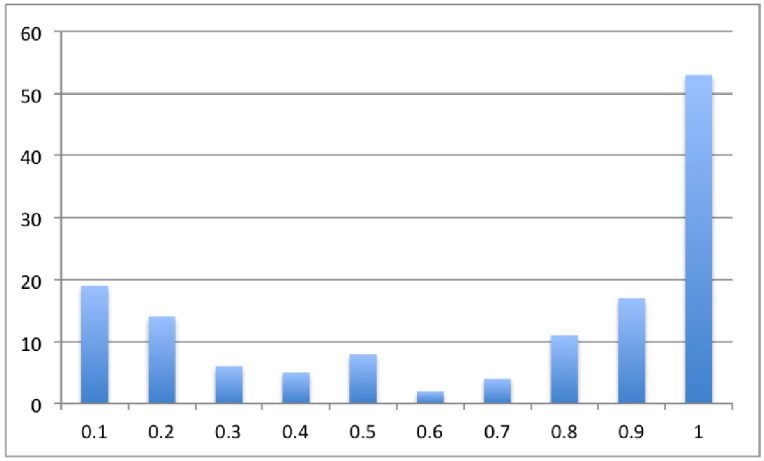
Bude nás zajímat, jak se systémy budou chovat při zpracování textu, který se nějakým způsobem zmiňuje o hledaných entitách. Méně přesné systémy totiž mohou extrahovat vztah, který se ve skutečnosti v textu nenachází. První systém, který pouze vyhledává klíčová slova v textu, například označí větu *"He was also able to paint in a romantic style, which had become more fashionable after the Civil War."* jako relevantní. Obsahuje totiž všechny potřebné části, ale bez syntaktické kontroly už systém nezjistí, jestli se skutečně jedná o hledaný vztah. Může pouze předpokládat. Proto je tedy zajímavé vybírat texty obsahující klíčová slova, abychom mohli odhalit případné nedostatky systému.

V datové sadě můžeme vynechat texty, které nejsou relevantní, ani neobsahují klíčová slova. Jedná se o skupinu *nn* z tabulky 2.1. Můžeme si všimnout, že více jak v polovině uvedených vzorečků s touto hodnotou ani nepočítáme. Důležitější jsou tři zbývající hodnoty. Datovou sadu tedy tvoří texty, ve kterých by některý systém potenciálně mohl najít vztah.

Menší část sady tvoří skupina, která byla vyhledána ručně a sloužila jako trénovací sada při vývoji. Jedná se o 55 relevantních textů. Zbytek (366) byl vyhledán automaticky pomocí prvního systému. To znamená, že každý takový text obsahuje větu s klíčovými slovy.

Tabulka 4.1: Datová sada

	Počet
Celkem	421
Relevantních	139
Nerelevantních	282
Ručně vyhledané	55
Automaticky vyhledané	366



Obrázek 4.1: Četnosti míry relevance v datové sadě

Zdrojem dat pro hledání byla kolekce krátkých článků o malířích, která byla sestavena v rámci projektu Decipher (viz 3.3.4). Těmito způsoby bylo získáno celkem 421 textů, z nichž 139 je relevantních. V tabulce 4.1 můžeme najít krátký přehled těchto číselných údajů.

Každý text v sadě je označen číselným údajem od 0 do 1, který určuje míru relevance. V některých větách je vztah přímo zmíněn, v jiných je formulace složitější a méně jasná. Následující seznam je orientační stupnicí relevance (řazeno sestupně):

- Přímý vztah (*was influenced by*), popis (*work inspired by*), slovní spojení (*travelling artist*)
- Nepřímý vztah (*painting resulted from mourning after death, experiences of war*)
- Složení vztahu a vlivu (*war influenced him, and this is expressed in painting*)
- Přímá návaznost (*when wife died, paintings became..., paintings became sad after wife died*)
- Nastínění události, která měla vliv (*he accompanied group of artists on a trip, which influenced work*)
- Nepřímá návaznost (*after death, sincerity becomes evident in work*)
- Případné ovlivnění (*may have influenced, would have influenced*)
- Nepřímá zmínka (*anti-war movement*)
- Možná návaznost (*after war he moved from style to style*)
- Jiná formulace
- Žádný vztah

Konečné rozhodnutí o míře relevance je ale individuální a záleží na konkrétní větě. Četnosti těchto hodnot můžeme vidět v grafu 4.1.

Tabulka 4.2: Orientační výsledky systémů pro prahovou hodnotu 0.8

Systém 1		
	Extrahováno	Neextrahováno
Relevatní	59	22
Nerelevatní	335	5
Systém 2		
	Extrahováno	Neextrahováno
Relevatní	34	47
Nerelevatní	45	295
Systém 3		
	Extrahováno	Neextrahováno
Relevatní	26	55
Nerelevatní	2	338

Tabulka 4.3: Orientační kvalitativní hodnoty systémů pro prahovou hodnotu 0.8

	Systém 1	Systém 2	Systém 3
Přesnost	0.14974	0.43037	0.92857
Úplnost	0.72839	0.41975	0.32098
Správnost	0.15201	0.78147	0.86460
Chybovost	0.84798	0.21852	0.13539

4.2 Uspořádání experimentů a přehled výsledků

S kompletní testovací sadou můžeme začít vyhodnocovat jednotlivé systémy. Orientační výsledky si můžeme prohlédnout v tabulkách 4.2 a 4.3. Jedná se o hodnoty naměřené s prahovou hodnotou nastavenou na 0.8.

Výsledky prvního systému

Výsledky přesnosti a úplnosti tohoto systému najdeme v tabulce 4.4. Vizualizace těchto dat je zbytečná, protože se všechny hodnoty pohybují v minimálním rozptylu. Přesnost je nízká (0.18) dle očekávání, úplnost je tím pádem vyšší (0.75). Je to způsobeno metodou extrakce (tento systém pouze vyhledává klíčová slova ve větách). To znamená, že velké procento označených textů není relevantní. A úplnost je vysoká, protože systém označí velkou část relevantních textů.

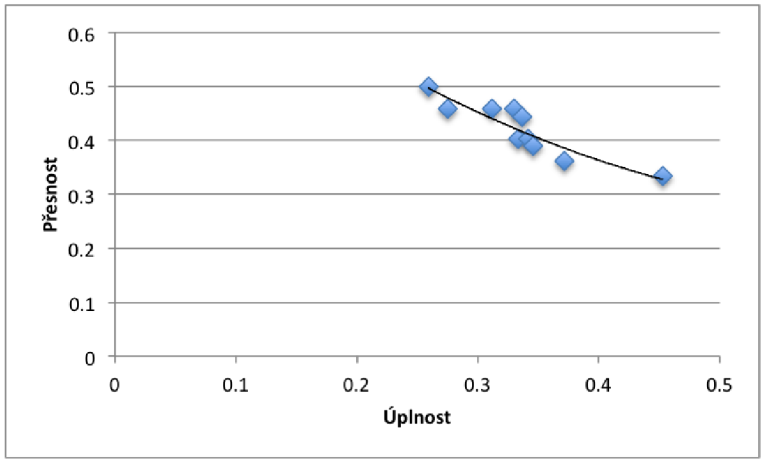
Správnost také odpovídá předpokladu a pohybuje se kolem 0.15. Je nízká, protože systém se rozhodne špatně ve většině případů. Mezi špatnými rozhodnutími převládá označování nerelevantních textů.

Výsledky druhého systému

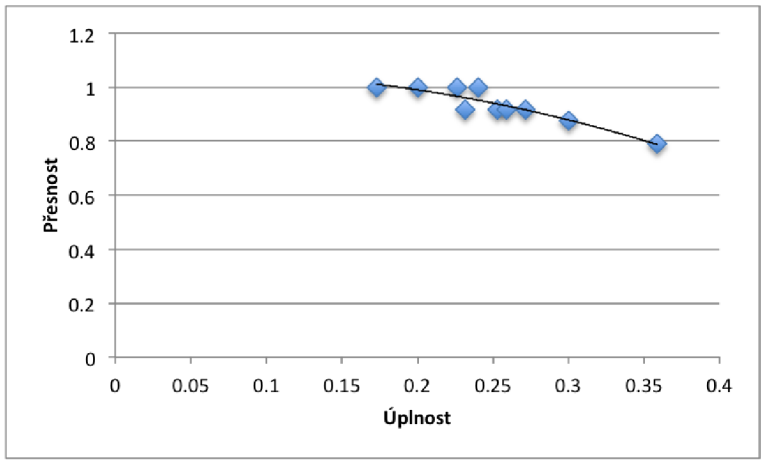
Výsledky přesnosti a úplnosti tohoto systému najdeme vizualizované v grafu 4.2. Výsledné hodnoty tohoto systému se významně liší od prvního systému. Můžeme si všimnout vyšší přesnosti (kolem 0.43), nižší úplnosti a vysoké správnosti pohybující se kolem 0.76. To jsou významné a zajímavé změny, vzhledem k tomu, že tento systém pouze navazuje na výsledky prvního systému.

Tabulka 4.4: Přesnost a úplnost pro první systém

Přesnost	Úplnost
0.28426	0.80575
0.23604	0.77500
0.20304	0.75471
0.19035	0.75000
0.17766	0.73684
0.15989	0.72413
0.15482	0.71760
0.14974	0.72839
0.12944	0.72857
0.10406	0.77358
0.17893	0.74946



Obrázek 4.2: Graf přesnosti a úplnosti pro druhý systém



Obrázek 4.3: Graf přesnosti a úplnosti pro třetí systém

Výsledky třetího systému

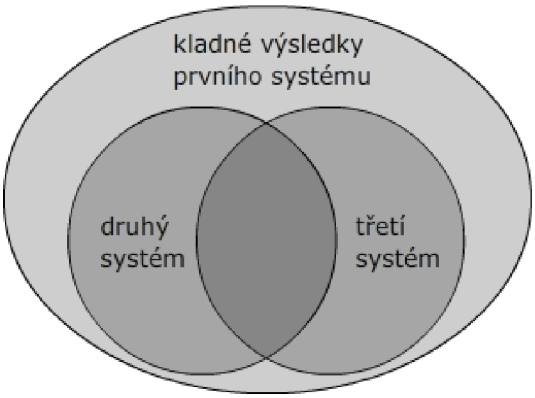
Výsledky přesnosti a úplnosti tohoto systému najdeme vizualizované v grafu 4.3. Třetí systém disponuje velkou přesností. Pro prahovou hodnotu 0.8 byla vyhodnocena dokonce 0.93. To znamená, že téměř každý označený text ve skutečnosti obsahoval vztah a systém se v tomto směru skoro nikdy nespletl. Vysoká přesnost ale znamená i nízkou úplnost, která se pohybuje okolo 0.32. Hodnoty správnosti jsou také vysoké, což je pozitivní výsledek.

4.2.1 Porovnání výsledků

Podle naměřených hodnot můžeme hodnotit jednotlivé systémy. Budou zde uvedeny hrozdíly při měření s prahovou hodnotou 0.8. Platí, že čím sofistikovanější systémy jsou, tím lepší podávají výsledky. První systém se tedy zachová správně pouze v 15% případů, kdežto pokročilý třetí systém v 86%. To je sice dáno i výběrem datové sady, kde první systém skončí vždy o něco hůř. Ale i s větším obsahem nerelevantních textů bez klíčových slov by rozdíl byl na první pohled znatelný, protože třetí systém navazuje na výsledky prvního. Jeho skupina *re* (viz tabulka 2.1) bude tedy vždy podmnožinou skupiny *re*, kterou vyprodukoval první systém. To stejné platí pro vztah mezi prvním a druhým systémem. Tento fakt ilustruje obrázek 4.4. I bez experimentálních výsledků tedy můžeme předpokládat, že druhý a třetí budou přesnější než první. A tento předpoklad můžeme potvrdit jasným rozdílem mezi hodnotami přesnosti.

Víme, že druhý systém by měl podávat lepší výsledky než první. Za zmínku ale stojí velký skok mezi hodnotami, a to jak u přesnosti, tak i u správnosti. Přesnost je přibližně o 29% vyšší a správnost dokonce o 63%. Když vezmeme v úvahu, že druhý systém se oproti prvnímu liší v podstatě jen kontrolou pořadí prvků, tak se jedná o významný rozdíl.

Ještě větší skok mezi hodnotami nastane v porovnání prvního a třetího systému, zde je ale více očekávaný. Přesnost je vyšší o 78% a správnost o 71%. Tyto rozdíly odpovídají složitosti řešení. Třetí systém provádí syntaktickou analýzu a používá pokročilé nástroje a postupy pro extrakci.



Obrázek 4.4: Překrývání kladných výsledků systémů.

4.3 Diskuse

Po získání výsledků je dobré rozebrat, jak si je můžeme vyložit a co jsme tedy zjistili. Tato sekce obsahuje rozbor výsledků a chyb.

Rozbor výsledků

Máme výsledky tří různých extrakčních systémů. Každý funguje jinak a má jiné výsledky. Nyní se zaměříme na to, abychom z výsledků vyvodili informace o daném systému. Můžeme zjistit k čemu je dobrý a jaké jsou jeho výhody a nevýhody.

První systém má velmi nízké hodnoty přesnosti a správnosti. Ve větách, které označí, sice pravděpodobně najdeme zmínku o události, ale jenom v některých případech se bude jednat o větu obsahující vztah. Jako plnohodnostný extrakční systém tedy vhodný není. Můžeme ho ale použít pro filtraci vstupů do jiných systémů nebo pro výpis vět s klíčovými prvky. Jeho výhodou je právě vysoká hodnota úplnosti, díky které označí velký počet textů, se kterými můžeme dále pracovat.

Druhý systém navazuje na výstupy prvního a přidává kontrolu pořadí prvků. Pro zhotovení takového systému je potřeba ručně sestavit pravidla, podle kterých pořadí kontrolujeme. Což se může jevit jako nevýhoda. Výhodou ovšem jsou lepší výsledky, takže systém už je spolehlivější. Hodnota správnosti vzrostla o 63% a přesnost činí 43%. To je zajímavý nárůst, ale pořád bude každá druhá věta označena špatně. Takový systém bychom tedy mohli použít například pro důkladnější filtraci.

Poslední systém vykazuje nejpřesnější výsledky. Z plného počtu relevantních vět sice označil jenom třetinu, ale téměř každé označení proběhlo v relevantní větě. To znamená, že výsledků nebude mnoho, ale budou velice přesné. Nicméně takový systém bychom ještě nemohli využít pro extrakci kvůli jeho nižší hodnotě úplnosti. Jeho základ funguje dobře, ale bylo by potřeba ho rozšířit, aby byl schopen identifikovat více větných staveb. Takto upravený by už byl připraven pro automatickou extrakci vztahů, které můžeme použít ve větších projektech.

Rozbor chyb

Při bližším prozkoumání výsledků můžeme zjistit, že některé relevantní věty nebyli označeny. Je třeba zjistit, v jaké části procesu extrakce se objevila chyba. Nevýhodou postupného zpracování (viz obrázek 3.1) je, že pokud selže jeden nástroj, proces skončí a další se na řadu

Tabulka 4.5: Výsledky systémů s potenciálním dokonalým NERem

Reálný NER			Ideální NER	
Systém 1				
	Extrahováno	Neextrahováno	Extrahováno	Neextrahováno
Relevatní	19	35	49	5
Nerelevatní	0	0	0	0
Systém 2				
	Extrahováno	Neextrahováno	Extrahováno	Neextrahováno
Relevatní	18	36	46	8
Nerelevatní	0	0	0	0
Systém 3				
	Extrahováno	Neextrahováno	Extrahováno	Neextrahováno
Relevatní	14	40	32	22
Nerelevatní	0	0	0	0

Tabulka 4.6: Výsledky systémů s potenciálním dokonalým NERem

Reálný NER				Ideální NER		
	Systém 1	Systém 2	Systém 3	Systém 1	Systém 2	Systém 3
Přesnost	1.0	1.0	1.0	1.0	1.0	1.0
Úplnost	0.35185	0.33333	0.25925	0.90740	0.85185	0.59259
Správnost	0.35185	0.33333	0.25925	0.90740	0.85185	0.59259
Chybovost	0.64814	0.66666	0.74074	0.09259	0.14814	0.40740

ani nedostanou. Po prohledání pomocných výpisů je zřejmé, že hlavním zdrojem neúspěšných extrakcí je nástroj NER. Ten se nachází již na začátku celého procesu a jeho selhání negativně ovlivní všechny tři systémy. Kvalitnější nástroj zastávající stejnou práci by tedy pravděpodobně zlepšil konečné výsledky.

Pro ilustraci a důkaz tohoto tvrzení se můžeme podívat do tabulek 4.5 a 4.6. Na levé straně jsou uvedeny výsledky systému za normálních okolností a na pravé jsou výsledky, které by nastali pokud by NER byl dokonalý. Tento jev byl simulován vyřazením nástroje a nahrazením funkcí, která podvrhne výsledky a vrací „nalezenou“ entitu. Ta však ve skutečnosti neexistuje a děje se tak pouze proto, abychom mohli zmást zbytek systému. Ten se potom chová stejně, jako kdyby ve větě nějakou entitu člověka skutečně našel. Zkušební datová sada pro tento experiment se skládá z 54 ručně nasbíraných textů. U nich totiž máme jistotu, že jsou relevantní, a můžeme na nich jednoduše pozorovat změny ve výsledcích. Už na první pohled vidíme, že každý systém je schopný extrahovat více vztahů. Číselné ohodnocení nám potvrzuje toto zlepšení oproti normálu. Přesnost bude vždy maximální, protože všechny texty jsou relevantní. Ale můžeme si všimnout markantního rozdílu v hodnotách úplnosti a správnosti. Největší pokrok najdeme u prvního systému, ale je nutno poznamenat, že druhý systém zlepšil svoji úplnost o 52%. Vzhledem k tomu, že výsledky NER takto mocně ovlivňují výsledky všech tří systémů, je nutné použít opravdu kvalitní nástroj tohoto typu při vývoji extrakčních systémů.

Dalším zdrojem neúspěchu bývá složitější stavba věty, nebo použití neobvyklých obratů. V takovém případě text neprojde filtrací prvního systému, nebo ostatní systémy mají problém rozpoznat vztah ve větě.

Shrnutí

Podle statistik jsme tedy zjistili, že kvalita výstupů odpovídá složitosti implementace systémů. Výsledky těchto tří systémů se od sebe tolik liší, protože každý funguje úplně jinak a na úplně jiné úrovni. Pro budování extrakčních systémů v praxi je tedy opravdu potřeba spoléhat se na pokročilé nástroje a postupy, protože jenom ty nám poskytnou vyšší přesnost a důvěryhodné výsledky.

Kapitola 5

Závěr

Kapitola obsahuje zhodnocení dosažených výsledků, popis vlastního přínosu a zamyšlení nad možnostmi dalšího rozvoje.

Dosažené výsledky

Byly navrženy a implementovány tři různé extrakční systémy. Každý na jiné úrovni a s odlišnými výsledky. První systém je jednoduchý a pracuje s regulárními výrazy a nástrojem NER. Druhý navazuje na práci prvního a přidává kontrolu pořadí klíčových prvků. Tato jednoduchá kontrola má velký vliv na přesnost a značně zlepšuje výsledky. Třetí systém přidává práci s nástrojem pro syntaktickou analýzu a vykazuje ještě větší přesnost.

Přínos práce

Přínosem je srovnání jednotlivých metod extrakce. Čtenář se může dozvědět, jaké nástroje volit a na co si dát pozor při případném vývoji podobného systému.

Možnosti dalšího rozvoje

Pro praktické využití by se mohl hodit třetí systém, díky použitým nástrojům a metodám. Je otevřený úpravám a vylepšením. Zejména těm, které by mohli navýšit hodnotu úplnosti, aby byl schopen označit více textů. To znamená například zvýšit počet větných staveb, které by mohl rozpoznat.

Literatura

[1] De Marneffe, M.-C.; Manning, C. D.: Stanford typed dependencies manual. 2008.

[2] Gerylovová, A.; Holčík, J.: *Úvod do statistiky. Text pro semináře.* Masarkova Univerzita, 2009.

[3] Hobbs, J. R.; Riloff, E.: Information extraction. *Handbook of natural language processing*, ročník 2, 2010.

[4] Liddy, E.: Natural Language Processing. ročník 2, 2001.

[5] Manning, C. D.; Raghavan, P.; Schütze, H.: *Introduction to information retrieval*, ročník 1. Cambridge University Press Cambridge, 2008.

[6] Manning, C. D.; Schütze, H.: *Foundations of statistical natural language processing.* MIT press, 1999.

[7] Smrž, P.; Mrnuštík, M.: Decipher-D4.1.1-WP4-BUT State of the art of event detection methods-PU. Technická zpráva, Brno University of Technology, December 2011.

[8] Thomas, A.; Kowar, M. K.; Sharma, S.: Extracting Noun Phrases in Subject and Object Roles for Exploring Text Semantics. January 2011.