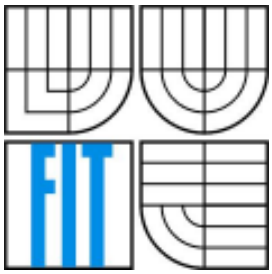
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ZNALEC ENCYKLOPEDIE
ENCYCLOPEDIA EXPERT

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. MARTIN KRČ

VEDOUCÍ PRÁCE
SUPERVISOR

doc. RNDr. PAVEL SMRŽ, Ph.D.

BRNO 2009

Abstrakt

Předmětem projektu je systém pro zodpovídání otázek formulovaných v přirozeném jazyce. Práce pojednává nejprve o problémech spjatých se systémy tohoto druhu a o některých uplatňovaných přístupech. Důraz je kladen na povrchové metody, které nejsou tolik náročné na dostupnost lingvistických zdrojů. V praktické části je pak popsán návrh systému, který zodpovídá faktografické otázky s využitím české Wikipedie jako zdroje informací. Extrakce odpovědí je založena zčásti na specifických rysech Wikipedie a zčásti na ručně předdefinovaných vzorech. Výsledky ukazují, že pro zodpovídání jednoduchých otázek je systém výrazně přínosnější než běžný vyhledávací stroj.

Klíčová slova

zodpovídání otázek, extrakce informací, vyhledávací stroj, Wikipedie, přirozený jazyk, čeština

Abstract

This project focuses on a system that answers questions formulated in natural language. Firstly, the report discusses problems associated with question answering systems and some commonly employed approaches. Emphasis is laid on shallow methods, which do not require many linguistic resources. The second part describes our work on a system that answers factoid questions, utilizing Czech Wikipedia as a source of information. Answer extraction is partly based on specific features of Wikipedia and partly on pre-defined patterns. Results show that for answering simple questions, the system provides significant improvements in comparison with a standard search engine.

Keywords

question answering, information extraction, search engine, Wikipedia, natural language, Czech

Citace

Krč Martin: Znalec encyklopedie. Brno, 2009, diplomová práce, FIT VUT v Brně.

Znalec encyklopedie

Prohlášení

Prohlašuji, že jsem tuto práci řešil samostatně pod vedením doc. RNDr. Pavla Smrže, Ph.D.
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Martin Krč
28. ledna 2009

Poděkování

Rád bych poděkoval doc. Pavlovi Smržovi za jeho podporu a pomoc při řešení obtíží spojených s tímto projektem a Bc. Stanislavovi Černému za rady ohledně derivačního slovníku. Poděkování patří rovněž Petrovi Vašíčkovi za pomoc při testování systému a mé rodině za projevenou trpělivost.

Obsah

1 Úvod

Vynález internetu ovlivnil zásadním způsobem dostupnost informací. I počítačový laik dnes dovede využívat služeb webových vyhledávačů a s jejich pomocí hledat informace v elektronických dokumentech rozličných formátů.

Stále je však co zdokonalovat. Klasické webové vyhledávače nedovolují uživateli vyjádřit přesné požadavky na hledanou informaci. Vyžadují od něj víceméně jen posloupnost klíčových slov a po skončení hledání mu prezentují seznam dokumentů, seřazený podle výskytu těchto slov. Tímto způsobem mohou uspokojit velkou část informačních potřeb uživatele, nikoliv však všechny. Hledaná informace bývá v textu formulována různými způsoby a uživatel musí často klíčová slova přepisovat, aby tuto formulaci postihl. Při hledání komplexnějších informací nemusí být uživatel ani schopen efektivní dotaz sestavit. Vyhledávače také zobrazují většinou celé textové pasáže obsahující klíčová slova, nikoliv přesně to, co uživatele zajímá. V době mobilních zařízení s malinkatým displejem přitom již nepostačuje schopnost nalézt relevantní text – je třeba jej prezentovat v co nejkompaktnější podobě.

Za nástupce klasických vyhledávačů lze považovat systémy, které dovedou zodpovídat přirozeně zapsané otázky. Potenciál otázek tkví v tom, že zachycují uživatelovy požadavky na hledanou odpověď věrněji než prostá posloupnost klíčových slov. Teoreticky tak může systém přesně odpověď lokalizovat a prezentovat ji uživateli bez matoucího textu navíc.

Cílem této práce je vytvořit český odpovídač na otázky, jemuž jako zdroj informací poslouží internetová encyklopedie Wikipedie.

V následující kapitole je popsána problematika související se systémy odpovídajícími na otázky. Čtenář se dozví o výzkumných konferencích zabývajících se touto oblastí a o některých existujících systémech. Důkladně jsou rozebrány obvyklé fáze činnosti systému a jsou nastíněny různé aspekty související s hledáním informací a zpracováním přirozeného jazyka.

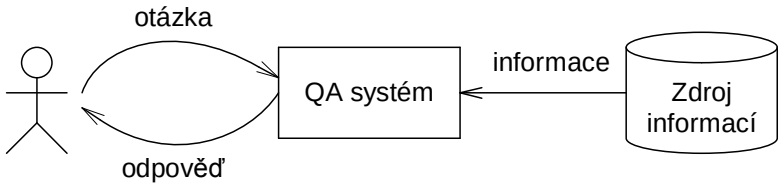
Kapitola 3 pojednává o návrhu vlastního systému a o použitých technologiích a lingvistických zdrojích. Zabývá se procesem indexace Wikipedie a využitím některých jejích prvků pro extrakci odpovědí. Popisuje, jakým způsobem program zpracovává vstupní otázku a jak využívá externí vyhledávací stroj. Rozebrán je také přístup k hodnocení výskytu klíčových slov.

V kapitole 4 je pomocí testovací sady otázek uskutečněno vyhodnocení systému. Jsou posouzeny nejen celkové výsledky hledání odpovědí, ale také úspěšnost jednotlivých fází činnosti.

2 Přehled problematiky

2.1 Odpovídání na otázky

Odpovídání na otázky (question answering, QA) je disciplínou, která spadá do oblasti **získávání informací** (information retrieval). Zabývá se systémy, jejichž cílem je poskytnout co nejpřesnější odpověď na otázku zadanou v přirozeném jazyce. Na rozdíl od běžných vyhledávacích strojů se QA systémy nespokojují s jednoduchým nalezením celých dokumentů nebo velkých úseků textu, ale snaží se o extrakci co nejkonkrétnější textové entity představující odpověď. To činí na základě požadavků uživatele získaných ze vstupní otázky.



Obrázek 2.1 Kontextový diagram QA systému.

Zaměření systémů

Podle charakteru možných otázek se QA systémy dělí do dvou skupin (viz [1]):

- Systémy pro otázky z **uzavřené domény –** Zaměřují se na nějakou užší oblast lidského vědění (typicky např. na medicínu, kulturní dění, zeměpisná témata). Výhoda takových systémů z hlediska jejich tvorby spočívá zejména ve snazším interpretování významu kladené otázky, ve snazším definování případných vzorů pro nalezení odpovědi a v možnosti využít odborných textových kolekcí.
- Systémy pro otázky z **otevřené domény –** Jejich smyslem je zodpovědět otázku z jakékoliv nebo skoro jakékoliv oblasti. Informace čerpají většinou z rozsáhlých zdrojů textu, které nebývají nijak zvlášť specializované (např. web či obecná encyklopedie). Konstrukce těchto systémů může být dosti komplexní, na druhou stranu je možné využít vyšší redundance výskytu hledané odpovědi.

Zdroj informací

Jako zdrojový korpus pro hledání odpovědí může sloužit buď znalostní báze obsahující informace ve strojově čitelné podobě, nebo kolekce textových dokumentů zprostředkovaná klasickým vyhledávacím strojem.

Znalostní báze představuje logicky konzistentní databázi znalostí z určité oblasti, zachycující například existenci jistých entit a vztahů mezi nimi. Koncepce znalostních bází souvisí už od svého vzniku především se specializovanými systémy a uzavřenou doménou otázek, při čemž znalostní data bývají produktem ruční práce expertů v oboru. Existují však také moderní systémy, jejichž znalostní báze mají širší zaměření a jsou plněny automaticky. [2]

Větší pozornost se dnes ubírá na systémy založené na **vyhledávacích strojích**, jimiž systémy prohledávají velký textový korpus (většinou otevřené domény) a snaží se z textu extrahovat příslušnou odpověď. Výhodou tohoto přístupu je, že není zapotřebí znalosti ručně připravovat, ale mohou být využity již dostupné informační zdroje (např. web). QA systém při tom komunikuje s vyhledávačem pomocí dotazů tvořených hledanými slovy, tedy podobně jako lidský uživatel při prohledávání textu. Rozpoznání odpovědi v textu je však pro QA systém nelehkým úkolem, jehož různé aspekty jsou předmětem intenzivního bádání a budou nastíněny i touto prací.

Zpracování textu

Odpovědní systém může k textu přistupovat dvěma způsoby (viz [1]):

- **Hloubkový přístup** – Využívá nejrůznější metody oblasti zpracování přirozeného jazyka, jako je syntaktická analýza otázky a textu zdrojového korpusu, zjednoznačňování významu slov (sémantická desambiguace), rozpoznávání pojmenovaných entit, zpracování anafor, logická analýza, temporální či prostorová analýza apod.
- **Povrchový přístup** – Přistupuje k textu jen jako k množině klíčových slov a nepokouší se o žádné zevrubné rozpoznání jeho smyslu. Používá zpravidla jen některé jednodušší metody zpracování přirozeného jazyka, např. morfologickou analýzu a povrchovou syntaktickou analýzu. Soustřeďuje se spíše na statistické hodnocení výskytu klíčových slov či aplikaci různých vzorů (např. ve formě regulárních výrazů).

V praxi se mohou oba přístupy kombinovat. Vždy je třeba přihlédnout k cílovému použití QA systému a zohlednit, nakolik daná metoda přispívá ke zkvalitnění jeho výsledků, nebo naopak ke zvýšení výpočetních nároků.

2.2 Historie a současný stav

2.2.1 Evaluační fóra

Nejen problematikou zodpovídání otázek, ale i jinými oblastmi vyhledávání informací v textu se již tradičně zabývá americká konference **TREC**[1] (*Text Retrieval Conference*), která probíhá každoročně od roku 1992 a je financována organizací *National Institute of Standards and Technology* a americkým ministerstvem obrany.

Činnost konference je podle oblastí zájmu rozdělena do několika kategorií, v rámci nichž probíhá srovnávání a hodnocení (tzv. evaluace) systémů zúčastněných univerzit a jiných vědeckých pracovišť. Kategorie *Question Answering Track* existuje od roku 1999, za tu dobu však prošla různými obměnami. Z počátku se zabývala pouze faktografickými otázkami, v roce 2003 byly navíc přidány otázky definiční a seznamové (o typech otázek pojednává oddíl 2.3.1). Byly testovány především otázky vztahující se k nějaké osobě, organizaci či předmětu. V roce 2005 byly zahrnuty také události a začal se klást důraz na časové hledisko otázky. Testovaným zdrojem informací pro zúčastněné systémy byl většinou korpus tvořený texty novinářského charakteru. V roce 2007 byly nově přidány texty z internetových blogů, které jsou oproti zpravodajským textům méně formální a hůře strukturo-

[1] http://trec.nist.gov

vané. V roce 2008 byla kategorie QA ve své obecné podobě zrušena a nahradila ji kategorie *Blog Track*, specializující se čistě na extrakci informací z blogů. [3, str. 1-2, 17]

Obdobou amerického TRECu je evropská konference **CLEF**[1] (*Cross Language Evaluation Forum*), která se zabývá získáváním informací v různých evropských jazycích a dělí se rovněž do několika kategorií podle oblastí zájmu. Na zodpovídání otázek se zaměřuje kategorie *QA@CLEF*, fungující od roku 2003. V rámci této kategorie jsou stejně jako u TRECu pravidelně hodnoceny jednotlivé systémy a neustále se zvyšují nároky na ně kladené. V roce 2007 bylo například zavedeno shlukování testovacích otázek podle témat a jejich svázání pomocí koreference. V roce 2008 se mimo klasického odpovídání na otázky účastníci zabývali třemi vedlejšími tématy: automatickou validací získaných odpovědí, zodpovídáním otázek podle transkriptů mluveného slova a zjednoznačňováním významu slov pro odpovídače. Jako zdroj informací se používají zpravodajské texty a encyklopedie Wikipedia. [4, str. 2]

Zvláště pro východoasijské jazyky má velký význam japonské fórum **NTCIR**[2] (*NII Test Collection for IR Systems*), pořádané od roku 1997. Zodpovídáním otázek se zabývá kategorie *Question Answering Challenge*, která funguje od roku 2002. Kategorie se zaměřuje na japonštinu, při čemž ze začátku to byly hlavně faktografické otázky, od roku 2006 se pak pozornost obrací na otázky vyžadující delší odpověď (viz [5]). Zároveň probíhá hodnocení v kategorii *Cross-Lingual Question Answering*, která klade důraz na systémy, jejichž vyhledávání funguje napříč jazyky (otázka zadaná v jednom jazyce, odpověď získaná z textu v jiném jazyce). Touto oblastí se zabývají i předchozí dvě konference, japonské fórum však na rozdíl od nich pracuje mimo angličtiny hlavně s čínštinou a japonštinou, což v některých ohledech vyžaduje specifický přístup (viz [6]).

2.2.2 Existující systémy

Mimo jiné na základě poznatků vytvořených v rámci výše popsaných lingvistických konferencí byly vyvinuty nejrůznější QA systémy.

Při míře dnešního využití internetu se zvláště užitečné ukazují otevřené systémy hledající odpovědi na celém webu. Jak již bylo zmíněno v úvodu, je možné na tento druh QA systémů nazírat jako na další etapu vývoje standardních webových vyhledávačů, jakými jsou **Google**[3], **Yahoo**[4] či **Seznam**[5]. Některé takové vyhledávače se snaží kromě běžných vyhledávacích dotazů zpracovávat i normální otázky (např. Google dovede zodpovědět geografické otázky typu „*What is the capital of Poland?*“), tyto jejich schopnosti jsou však, pokud jde o složitost otázek a pokrytí odpovědí, stále dosti omezené. [7, str. 63]

Určitou alternativou k běžným vyhledávačům je například velmi oblíbený server **Ask.com**[6] (dříve nazývaný AskJeeves), který disponuje obsáhlou znalostní bází, naplněnou především informacemi z důvěryhodných a dobře strukturovaných serverů (z encyklopedií, serverů veřejných institucí, online kolekcí otázek a odpovědí atd.). Snahou tohoto vyhledávače je hlavně umožnit uživateli zadávat zcela přirozené otázky a navádět ho případně i na otázky související.

---

[1] http://www.clef-campaign.org

[2] http://research.nii.ac.jp/ntcir

[3] http://www.google.com

[4] http://www.yahoo.com

[5] http://www.seznam.cz

[6] http://www.ask.com

AnswerBus[1] při hledání odpovědi na zadanou otázku volí dva až tři z pěti webových vyhledávačů a s jejich pomocí extrahuje příslušné informace z webu. Výsledkem je pak jediná věta, doplněná odkazem na zdroj. Hodnocení vět provádí jednak na základě výskytu klíčových slov otázky ve větě a jednak s využitím informace o typu otázky, rozpoznání anafor a extrakce pojmenovaných entit. [8]

Systém **START**[2] využívá stejně jako Ask.com při odpovídání znalostní bázi (nazývanou Omnibase), která představuje jakési abstraktní rozhraní pro přístup k různě strukturovaným zdrojům na internetu. Informační zdroje jsou ručně či automaticky opatřeny tzv. anotacemi, což jsou jednoduché věty nebo fráze popisující obsah daného segmentu. Ty jsou pak uloženy do znalostní báze ve formě ternárních výrazů „*OBJEKT - VLASTNOST - HODNOTA*“, které umožňují nalezení odpovědi nezávisle na původní syntaxi otázky a zdrojového textu. Zvláštností tohoto systému je, že jako odpověď prezentuje někdy i multimediální data (fotografie, mapy apod.). [9]

Systém **QuALiM**[3] pro svou činnost nezavádí žádnou znalostní bázi, ale spoléhá na soustavu vzorů a na informace ze speciálních lexikálních slovníků (FrameNet, PropBank, VerbNet), s jejichž pomocí transformuje otázku na předpokládanou odpověď, například:

„what is the population of Czech Republic?“ – „the population of Czech Republic is NUMBER“

Odpovědi vyhledává na webu, ale prezentuje je v kontextu získaném z Wikipedie. [10, 11]

2.3 Obvyklé fáze

Ačkoliv se různé QA systémy použitými metodami navzájem více či méně liší, u všech systémů založených na vyhledávacích strojích je obvykle možné princip činnosti rozdělit do čtyř kroků, které budou jednotlivě popsány v tomto oddíle.

2.3.1 Zpracování otázky a její klasifikace

Pro správné nalezení příslušné odpovědi je nejprve zapotřebí rozpoznat, na co se vůbec uživatel ptá. Typický QA systém musí rozlišovat mezi různými typy otázek a očekávaných odpovědí, aby dovedl zvolit správnou strategii hledání odpovědi a co nejvíce toto hledání usnadnil.

2.3.1.1 Typy otázek

Pro účely výzkumu v oblasti Question Answering je běžnou praxí otázky rozdělovat podle toho, jakou podobu má mít hledaná odpověď. Například **faktografická** (faktoidní) otázka očekává jako odpověď typicky jednoduchou jmennou frázi či pojmenovanou entitu, **seznamová** otázka vyžaduje celý seznam takových frází či entit, **definiční** otázka očekává obsáhlejší text popisující dané téma apod. Příklady různých takových typů otázek jsou uvedeny v tabulce 2.1. Pro řešení jednotlivých typů je třeba použít často dosti odlišné techniky vyhledávání, proto bývají evaluační fóra (viz oddíl 2.2.1) i příslušné vědecké práce omezeny jen na jeden či několik málo typů.

[1] http://www.answerbus.com

[2] http://start.csail.mit.edu

[3] http://demos.inf.ed.ac.uk:8080/qualim

Formát odpovědi	Příklad otázky
Fakt	*Ve kterém městě se narodil Adolf Hitler?*
Seznam	*Jaké přísady patří do rajské omáčky?*
Definice	*Kdo to byl hrabě Drákula?*
Důvod/příčina	*Proč zanikla Říše římská?*
Vztah	*Jaký je vztah mezi Byronem a Řeckem?*
Ano/ne	*Existovaly tanky už v 19. století?*
…	

Tabulka 2.1 Typy otázek podle formátu odpovědi.

Problémem ovšem je, jakým způsobem rozpoznání toho či onoho typu otázky implementovat v QA systému. Přitom je žádoucí na základě dané otázky rozpoznat nejen formát odpovědi, ale především **očekávaný typ odpovědi** (expected answer type, EAT). Typem odpovědi přitom rozumíme určitou sémantickou kategorii, ať už obecnou (např. osoba, popis, místo) či naopak velmi konkrétní (např. rakouský prezident, druh motýla), do níž má spadat entita představující odpověď.

Tázací slovo	Příklad otázky
KDO	*Kdo vládl v Číně kolem roku 1200?*
KTERÝ	*Ve kterém městě se narodila Božena Němcová?*
KOLIK	*Kolik obyvatel má Kalifornie?*
JAK	*Jak vznikly úvahy o evoluční teorii?*
PROČ	*Proč vypukla první světová válka?*
…	

Tabulka 2.2 Typy otázek podle tázacího slova.

Běžně používanou metodou je triviální dělení otázek podle tázacího slova umístěného zpravidla na počátku či poblíž počátku věty (viz tabulka 2.2). Jedná se o velmi jednoduchou kategorizaci, díky níž získá systém přibližnou představu o očekávaném typu odpovědi (např. *kolik – počet*, *kdo – osoba*). V žádném případě ovšem nejde o spolehlivé mapování, jelikož použití jednotlivých tázacích slov se může různě navzájem prolínat a k určení typu odpovědi je většinou zapotřebí i zbytek otázky. Například následující dvě věty začínají stejným tázacím slovem a jsou po syntaktické stránce velmi podobné, přesto první očekává jako odpověď fakt, zatímco druhá vysvětlení postupu:

Jak se jmenoval první turecký sultán? – Jak se šije pravá skotská sukně?

2.3.1.2 Fokus a téma

V souvislosti s hledáním typu odpovědi se obvykle hovoří o tzv. **fokusu otázky** (popsán např. v [12]). Fokus je část otázky, která velmi dobře vypovídá o tom, jaký je očekávaný typ odpovědi. Nejtypičtěji nacházíme fokus v otázkách začínajících na JAKÝ/KTERÝ:

Z jakého <u>materiálu</u> se vyrábějí pneumatiky? *Který <u>český vynálezce</u> žil déle než 80 let?*

První otázka se ptá na materiál, druhá na českého vynálezce. V uvedených příkladech je fokus ve formě jmenné fráze, což je také nejobvyklejší. Uvědomme si, že taková jmenná fráze může být i dosti rozvitá – například v otázce *„Který <u>teplokrevný živočich žijící v subtropických oblastech</u> má za potravu mravence?“* je fokus tvořen šesti slovy. Zásadní je přitom především hlava této fráze (*živočich*), zatímco ostatní slova tvoří jen tzv. modifikátory fokusu [13]. Fokus ale mívá i jinou podobu – v následujících větách může být například za fokus označeno přídavné jméno, respektive příslovce:

Jak rychlá existují dopravní letadla? *Jak rychle létají dopravní letadla?*

Systém QuALiM dokonce v jistých případech rozpoznává jako fokus i slovesa [11].

Kromě fokusu je možné identifikovat tzv. **téma otázky**, označující objekt či událost, které se otázka týká. Například ve větě *„Jaká je výška Lysé hory?“* je tématem otázky *Lysá hora* a fokus *výška* určuje vlastnost tohoto objektu, o níž se zajímáme. Téma otázky je na rozdíl od fokusu pojmem poměrně neurčitým a nejsou výjimkou věty, v nichž lze rozpoznat různá témata. Například v otázce *„Který francouzský prezident jednal za druhé světové války se Stalinem?“* může být tématem *druhá světová válka*, *Stalin* nebo i *Francie*.

Zatímco fokus otázky má zásadní vliv na způsob hledání odpovědi, téma otázky, jakožto důraz na jeden z pojmů ve zbytku otázky, pouze může za určitých okolností hledání usnadnit. Velká část QA systémů (např. [8, 9]) žádný z pojmů otázky za téma neoznačuje a jejich výsledky tím nejsou nijak limitovány. Na druhou stranu například systém QuALiM téma identifikuje a snaží se při hledání nalézt takové textové pasáže v korpusu, které toto téma obsahují v nadpise [11]. V rámci evaluačního fóra TREC (od roku 2004) a CLEF (od roku 2007) se testují také otázky uspořádané do shluků podle tématu a svázané pomocí koreference [3, 4]. Například:

Ot1: *Who is George W. Bush?* Ot2: *When was he born?* Ot3: *Who is his wife?*

V případě takovýchto otázek je rozpoznání tématu nezbytné, jinak by nebylo možné zodpovědět druhou a třetí otázku, obsahující jen anafory (*he* a *his*) zastupující objekt uvedený v první větě (*George W. Bush*). Kromě práce s tématem a rozpoznání anafor má toto shlukování smysl v tom, že zavádí do vývoje QA systému problematiku udržování kontextu mezi kladenými otázkami, což je nezbytné, pokud má uživatel se systémem komunikovat formou dialogu.

2.3.1.3 Klasifikace podle typu odpovědi

Vraťme se ale k problematice rozpoznání očekávaného typu odpovědi. V předchozím textu jsme uvedli, že k jeho rozpoznání může přispět tázací slovo a fokus otázky. Nabízí se tedy možnost otázku zpracovat s využitím tradičních lingvistických metod (např. morfologickou a syntaktickou analýzou), extrahovat z ní tázací slovo a fokus a srovnat je se slovníkem, který mapuje příslušné výrazy na typy odpovědi (viz např. tabulka 2.3).

Tázací slovo	Fokus	Typ odpovědi	Tázací slovo	Fokus otázky	Typ odpovědi
kdo	-	OSOBA	který	země	MÍSTO
který	panovník	OSOBA	co	-	PŘEDMĚT
který	vojevůdce	OSOBA	kde	-	MÍSTO
který	zbraň	PŘEDMĚT	kdy	-	ČAS

Tabulka 2.3 Mapování tázacího slova a fokusu na typ odpovědi.

Pokrytí velkého množství různých fokusů je v praxi velmi obtížné, proto se využívá pomocných zdrojů, mezi něž patří například sémantický lexikon Wordnet [18]. Typický systém pracující tímto způsobem (např. [19]) vyhledá ve Wordnetu fokus otázky, postupuje po jeho hyperonymech (nadpojmech) a jakmile se dostane na pojem, který již zná, přiřadí podle něj k otázce příslušný typ odpovědi. Například:

turecký sultán $\rightarrow$ *sultán* $\rightarrow$ *panovník* $\rightarrow$ *OSOBA*

Problémem tohoto způsobu rozpoznání typu odpovědi je, že prakticky zanedbává jiné části otázky než tázací slovo a fokus. Přitom mohou existovat věty, v nichž fokus není vůbec přítomen a samotné tázací slovo neposkytuje dostatek informací. Například u obou následujících otázek člověk snadno rozpozná, že se pravděpodobně ptají na název knihy, přestože druhá otázka neobsahuje žádný fokus:

Jaký román napsala Božena Němcová? *Co napsala Božena Němcová?*

Kromě toho je vůbec otázkou, jakým způsobem tázací slovo a fokus ve větě hledat. Uvažme, že existuje spousta možností, jak položit víceméně tutéž otázku (nemusí jít ani o tázací větu):

Ve kterém roce byla založena Karlova univerzita? *Karlova univerzita byla založena ve kterém roce?*
Můžeš mi říct rok založení Karlovy univerzity? *Uveď rok založení Karlovy univerzity.*

Alternativou k přímé extrakci a využití tázacího slova a fokusu je nahlížení na problém rozpoznání typu odpovědi jako na obecnou **klasifikaci textu**, známou úlohu z oblasti zpracování přirozeného jazyka. Textem je v tomto případě otázka a smyslem klasifikace je přiřazení otázky do jedné či více tříd odpovědních typů. Formálně řečeno odpovídá klasifikace otázky přiřazení logické hodnoty (ano/ne) ke každé dvojici $(q_j, t_i) \in Q \times T$, kde Q je doména otázek a $T = \{t_1, t_2, ..., t_{|T|}\}$ je množina předdefinovaných typů. [14]

Ze všeho nejdříve je zapotřebí definovat množinu odpovědních typů. Tato množina by se měla nějakým způsobem odvíjet od domény QA systému a okruhu otázek, na jejichž odpovídání se systém zaměřuje. V rámci evaluačních fór probíhaly ve velké míře experimenty se systémy založenými na víceúrovňové taxonomii odpovědí. Li a Roth [15] vytvořili například dvouvrstvou taxonomii, tvořenou z 6 obecných tříd a 50 konkrétnějších, Suzuki et al. [16] použili při řešení celkem 150 typů odpovědi, strukturovaných do čtyř vrstev, Hickl et al. [17] klasifikovali otázky do třívrstvé hierarchie, čítající přibližně 300 typů. Autoři taxonomie typů někdy vycházejí z předpokladu, že úspěšný systém musí pracovat s co nejpřesnější taxonomií, tedy s co největším počtem typů. Ukázalo se však (viz např. [19, str. 150]), že kvalita výsledků nemusí růst s počtem typů. Zdá se být rozumné zohledňovat při tvorbě taxonomie schopnosti modulu, který bude poté pro ten či onen typ extrahovat z textu příslušné kandidátní odpovědi (takto postupuje třeba [17]).

Co se týče realizace samotné klasifikace, existuje několik přístupů. Možným řešením je ručně sestrojit **vzory** otázek například ve formě regulárních výrazů, nebo definovat určitá **pravidla** vyžadující přítomnost jistých klíčových slov v otázce apod.

Příklad vzoru:
```
((co)|(jaká událost) [proběhnout]|[uskutečnit se]) v|na [MÍSTO] roku [ČÍSLO]
```

Příklad pravidla:
```
(ZACINA_NA(co) || ZACINA_NA(jaký)) && OBSAHUJE(událost) && OBSAHUJE(roku)
```

Ruční specifikování vzorů či pravidel je výhodné v tom, že nevyžaduje existenci žádné rozsáhlé trénovací množiny otázek doplněných o typ odpovědi. Člověku stačí existence pár příkladů otázek a na jejich podkladě dovede sestavit obecné vzory či pravidla, díky nimž systém úspěšně klasifikuje otázky daného formátu. Na druhou stranu je tato metoda málo flexibilní – při změně řešené domény či jazyka je třeba vzory/pravidla zásadně upravit. Problémy mohou vzniknout už při pouhém odhalení nových případů otázek, kdy dojde ke zjištění, že některé vzory jsou nepřesné. V neposlední řadě pak proti této metodě hovoří časová náročnost ruční práce, která je spojená s definováním vzorů/pravidel pro dostatečné pokrytí různých otázek.

Uvedené problémy řeší druhý přístup ke klasifikaci otázky, kterým je **statistická klasifikace** s využitím **strojového učení**. Klasifikovaný text bývá obecně vyjádřen vektorem $\vec{x} = (x_1, x_2, ..., x_n)$, kde x_i jsou pro text specifické příznaky – například přítomnost různých slov, bigramů, pojmenovaných entit apod. U klasifikace otázek pro účely QA systémů je na rozdíl od kategorizace dokumentů zapotřebí obecně více druhů příznaků, včetně různých strukturálních či sémantických údajů [16]. Li a Roth [15] používají jako příznaky například slova, morfologické značky, fráze, pojmenované entity, sémanticky významná slova, n-gramy a speciální relační příznaky založené na vztazích mezi některými příznaky.

Mezi typické metody statistické klasifikace otázek s využitím strojového učení patří například algoritmus *k* nejbližších sousedů, naivní Bayesův klasifikátor, logistická regrese, rozhodovací stromy, podpůrné vektory (Support Vector Machines, SVM) nebo metoda Sparse Network of Winnows (SNoW). Tyto metody jsou v kontextu klasifikace otázek různě srovnávány (viz např. [14]), přesto nebylo dosud zjištěno, že by některá z nich výrazně vynikala nad ostatní. Konkrétní popis technik je bohužel mimo rozsah této práce.

Metodu SNoW využili například Li a Roth [15], jejichž klasifikátor učený trénovací množinou 5500 otázek dosahoval v rámci konference TREC 10 úspěšnosti klasifikace otázky 95 %. Suzuki et al. [16] sestrojili klasifikátor SVM, který po vytrénování sadou 5011 otázek rozpoznal správně 88 % otázek.

Jak již bylo naznačeno výše, disponuje statistická klasifikace otázek oproti klasifikaci ručně definovanými vzory či pravidly výhodu v podobě flexibilnějšího použití a lepší škálovatelnosti. Navíc není třeba investovat velké množství času do konstrukce příslušných vzorů/pravidel. Dovede si také poradit s otázkami, jejichž klasifikace je nejednoznačná a dá se vyjádřit pouze relativní mírou příslušnosti k danému typu odpovědi. Na druhou stranu je pro co nejkvalitnější statistickou klasifikaci zapotřebí obsáhlá trénovací sada otázek, postihující v dostatečné míře různé otázky.

Hybridní klasifikace, kombinující statistickou klasifikaci s množinou předdefinovaných vzorů či pravidel, může za určitých podmínek těžit z výhod obou přístupů. V rámci konference TREC 2006 a 2007 si například vedl při odpovídání faktografických otázek nejlépe systém PowerAnswer (viz [21]), který používal jak přesná pravidla postihující běžné otázky, tak statistickou klasifikaci pro řešení nejednoznačných otázek.

2.3.1.4 Problematické otázky

Na závěr oddílu o zpracování otázky je vhodné poznamenat, že toto zpracování může být v praxi ještě ztíženo, zadá-li uživatel otázku v nějaké nestandardní podobě. Ignatova et al. [22] například rozlišují pět případů otázek, které mohou pro QA systém představovat problém, přestože jsou zejména v internetovém prostředí zcela běžné:

Typ problému	**Příklad otázky**
Nesprávný pravopis	*Ze kterého statu pochází Umberto Eco?*
Internetový slang	*How r plants used 4 medicine?* (specifikum angličtiny)
Chybná syntaxe	*Jaké budou od ledna pravidla silničního provozu?*
Dotaz klíčovými slovy	*nejrozšířenější nemoc, Afrika*
Nejednoznačnost	*Kolik obyvatel má Dolní Lhota?*

Tabulka 2.4 Problematické podoby otázek.

Je tedy na zvážení, nakolik má být daný systém odolný vůči různě položeným otázkám. Není přitom třeba zdůrazňovat, že jednoduchost používání je jedno z hlavních kritérií úspěšného softwarového systému.

2.3.2 Získání textových segmentů

Po zpracování otázky a stanovení očekávaného typu odpovědi následuje obvykle proces, jehož úkolem je získat ze zdrojového korpusu textové segmenty, které by mohly obsahovat odpověď.

2.3.2.1 Vyhledávací stroj

Pro účely získání nadějného textu ze zdrojového korpusu slouží většinou externí **vyhledávací stroj**, který má dopředu naindexované příslušné textové dokumenty a dovede v nich efektivně hledat požadovaný text. Zdrojem může být přitom ohromná kolekce dynamicky se vyvíjejících dokumentů (např. u webového vyhledávače), nebo naopak relativně malé množství dokumentů umístěných na jediném počítači.

Ačkoliv se různé vyhledávače liší strukturou uloženého indexu i procesem hledání, jejich komunikační rozhraní bývají zpravidla založena na podobném principu. Na vstupu obvykle očekávají vyhledávací dotaz tvořený posloupností slov, doplněných o případné modifikátory (uvozovky, operátory, speciální symboly…) upřesňující požadavky na hledání. V tabulce 2.5 jsou uvedeny ukázkové dotazy pro webový vyhledávač Google.

Příklad dotazu	Požadavek na hledané dokumenty
`velká říjnová revoluce`	co nejvíce z těchto tří slov
`velká „říjnová revoluce“`	druhé a třetí slovo se musí nacházet v nalezených dokumentech za sebou
`velká říjnová +revoluce`	třetí slovo musí být v nalezených dokumentech povinně
`velká OR říjnová revoluce`	výskyt prvního a druhého slova má stejnou váhu jako výskyt jen jednoho z nich

Tabulka 2.5 Ukázka syntaxe dotazů pro Google.

Základní indexovanou jednotkou pro hledání bývá slovo, zatímco ostatní elementy textu (interpunkce apod.) jsou ignorovány. Stejně tak je při hledání obvykle ignorována i velikost písmen a výskyt funkčních slov (předložky, spojky apod.).

Pro úspěšné využití vyhledávacího stroje v QA systému je nezbytné rozhodnout zejména způsob generování vyhledávacího dotazu a způsob indexace zdrojového textu (tedy i podobu vraceného výsledku).

2.3.2.2 Generování dotazu

Smyslem generování dotazu je na základě předzpracované vstupní otázky (viz oddíl 2.3.1) vytvořit takový dotaz pro vyhledávací stroj, který zajistí vrácení co nejrelevantnějších textových segmentů – tedy s co největší šancí, že se v těchto segmentech bude nacházet hledaná odpověď.

Která slova otázky?

Nejjednodušším přístupem je položit jako dotaz vyhledávači všechna slova otázky, s případným vyloučením funkčních slov nebo obecně slov, která se ve zdrojovém korpusu vyskytují příliš často a

nemají z hlediska hledání určující charakter. Například z otázky *„Ve kterém roce se narodil Petr Chelčický?“* takto vznikne dotaz *„roce narodil Petr Chelčický“*, což je intuitivně správné, neboť to zcela postačuje pro vyhledání například následující pasáže s odpovědí:

*… Petr Chelčický patřil mezi naše přední náboženské myslitele. Narodil se přibližně v roce **1390** v Chelčicích u Vodňan a pocházel z tzv. nižší venkovské šlechty…*

Uvažme, že otázka může obsahovat i slova, která se v textu s odpovědí buď vyskytují jen někdy, nebo dokonce vůbec. Moldovan et al. [12] například upozorňují na skutečnost, že není jednoznačné, zda do vyhledávacího dotazu zahrnout fokus otázky. Srovnejme například následující dvě věty:

Ve kterém století probíhala Stoletá válka? *Proti které zemi bojovala Francie ve Stoleté válce?*

Zatímco u první z otázek je vhodné fokus *století* mezi klíčová slova umístit, jelikož se bude pravděpodobně nacházet v odpovědní entitě (např. *14. století*), u druhé by slovo *zemi* ztěží hledání usnadnilo; naopak může dojít k tomu, že budou upřednostněny textové segmenty s tímto slovem na úkor žádaných segmentů s odpovědí.

Systém Lasso [12] řeší problematiku umisťování slov do dotazu flexibilně podle úspěšnosti toho kterého dotazu. Vyhledávací dotaz je nastaven tak, aby povinně vyžadoval přítomnost VŠECH klíčových slov v nalezených segmentech. Systém začíná od velmi specifického dotazu, obsahujícího skoro všechna slova otázky, a v případě, že tento dotaz nevrátí dostatečný počet segmentů, některá slova se z něj odeberou a hledání probíhá znovu. To se opakuje, dokud je hledání neúspěšné, při čemž volba slov odebíraných z dotazu je řízena různými heuristikami (slovo v uvozovkách, pojmenovaná entita, jednoduchá jmenná fráze, podstatné jméno sloveso, fokus…).

Při výběru slov pro vyhledávací dotaz se mohou uplatnit i metody strojového učení. Monz [25] navrhl například systém, který extrahuje z jednotlivých slov otázky až 18 různých příznaků a pomocí naučeného rozhodovacího stromu stanovuje váhy jednotlivých slov. Tyto váhy vyjadřují užitečnost slov z hlediska vyhledávacího dotazu a jsou následně použity pro optimalizaci hledání.

Expanze slov otázky

Málokdy je situace natolik ideální, aby pro hledání postačovala slova uvedená v otázce. Vhodné přidání nových slov do dotazu bývá nejen nástrojem pro zlepšení pokrytí hledání, ale většinou i nutností, bez níž by nebyla odpověď vůbec nalezena.

Naskýtá se například otázka, jakým způsobem řešit problematiku ohýbání slov, tedy existence téhož slova v různých tvarech vyjadřujících příslušné mluvnické kategorie (pád, číslo, osobu…). V případě otázky o Petru Chelčickém je například žádoucí, aby byly nalezeny nejen výskyty slova *roce*, ale i *rok*, *roku*, *rokem* atd. Tento problém je přitom kritický hlavně pro flektivní jazyky, mezi něž patří čeština, nicméně nebývá opomíjen ani u angličtiny (viz např. [23, str. 32]), typického morfologicky chudého jazyka.

Jednou z možností je s využitím morfologického slovníku každé slovo v dotaze rozvinout o všechny jeho tvary, např: *„(rok roku rokem…) (narodit narodil narodí…) (Petr Petr Petrovi…) (Chelčický Chelčickému Chelčického…)“*. Častěji bývá však problém řešen pomocí tzv. **lemmatizátoru**, převádějícího jednotlivá slova na jejich základní tvar neboli lemma (např. na substantivum v prvním pádě jednotného čísla, sloveso v infinitivu apod.). Veškerá slova zdrojového korpusu jsou pak indexována svým lemmatem (vyhledávač tedy nerozlišuje například mezi *Petr* a *Petrovi*) a stejně

tak jsou na lemmata transformována slova dotazu (např. *„rok narodit Petr Chelčický“*). Výhoda lemmatizace oproti expanzi dotazu spočívá v menší velikosti indexu slov (nejsou indexovány nelemmatické tvary) a v rychlejším hledání díky kratšímu dotazu. Namísto lemmatizátoru bývá používán také **stemmer**, což je jednodušší modul, který pouze odtrhává koncovky slov a ponechává jejich kmen (anglicky *stem*).

Lemmatizace nebo stemming je běžnou součástí QA systémů (např. [12, 17]), zvyšující ve větší či menší míře pokrytí hledání. Zároveň však nevyhnutelně dochází ke snížení přesnosti výsledků, což je důvod, proč některé systémy (zvláště v kontextu morfologicky nenáročné angličtiny) tuto techniku nevyužívají [25, str. 50].

Kromě morfologického aspektu je nezbytné u slov řešit také aspekt sémantický. Například odpověď na otázku *„Proti které zemi bojovala Francie ve Stoleté válce?“* může být v textu formulována jako *„Ve Stoleté válce se střetla Francie s Anglií.“*. V tomto případě tedy nebylo vůbec použito slova *bojovat*, nýbrž významově podobného *střetnout se*. Řešením problému je uskutečnění rozvoje jednotlivých klíčových slov o jejich **synonyma**. Vyhledávací dotaz pro uvedenou otázku a jeho příslušná expanze o synonyma mohou vypadat například takto:

```
„země bojovat Francie stoletý válka“ ->
„(země stát) (bojovat střetnout) (Francie) (stoletý) (válka válčení)“
```

Naskýtá se samozřejmě otázka, jak hodně jednotlivé pojmy expandovat, aby došlo k rozumnému nárůstu pokrytí, ale nesnížila se drasticky přesnost. Synonyma nemusí být významově zcela shodná, může stačit, pokud je jejich význam podobný. Přichází tedy v úvahu různá míra sémantické expanze pojmů, dávající různé výsledky hledání. Jako zdroj informací o synonymech používají QA systémy (např. [26, 27]) velmi často sémantický lexikon Wordnet, nebo speciální thesaurus. Pomocí Wordnetu je možné navíc slova v dotaze expandovat nejen o synonyma, ale také o jinak související pojmy, například o **troponyma** (*bojovat – šermovat, boxovat, útočit*) či různé **odvozeniny** (*Francie – francouzský, Francouz*).

Velkým problémem sémantického rozvoje slov a obecně celého procesu odpovídání na otázky je **polysémie** neboli mnohovýznamovost slov. Slovo *stát* může například označovat jak státní útvar, tak sloveso vyjadřující stání. Víceznačná slova v dotaze nutně vedou ke snížení přesnosti, neboť často svádějí vyhledávač na textové segmenty obsahující slovo v jiném významu, než je ten zamýšlený. Do určité míry však výsledky korigují ostatní slova v dotaze, která svou přítomností zjednoznačňují význam daného slova – například slovo *stát* se jeví jinak v kontextu dotazu *„Francie stát obyvatelé“* než v *„dům stát les“*. Avšak v případě, že systém uskutečňuje sémantický rozvoj slov, může dojít ke značnému zkreslení – typicky například přidáním synonym špatného významu některého ze slov (třeba *„dům <stát země republika> les“*). Řešením je ještě před rozvojem zjednoznačnit význam slov nějakou statistickou metodou či pomocí ručních pravidel.

Synonyma či jiná slova vhodná k začlenění do dotazu je možné získat také statisticky přímo ze zdrojového korpusu. Yang et al. [28] pokládají například vyhledávači jako dotaz nejprve samotná slova otázky a z navrácených textových segmentů extrahují pojmy, které se v nich vyskytují dostatečně často poblíž dotazových slov. O tyto pojmy pak rozvíjejí původní dotaz a teprve potom provádějí hlavní hledání.

Přidání slov podle typu odpovědi

V předchozím textu byly popsány možnosti, jak rozvinout dotaz o slova, která mají nějaký vztah ke slovům z otázky. Existují však také snahy přidat do dotazu slova vycházející z očekávaného typu

odpovědi [25, str. 122]. Prager et al. [29] uskutečňují například tzv. **prediktivní anotaci**, spočívající v tom, že jsou ve zdrojovém korpusu dopředu rozpoznány různé typy entit a příslušné textové segmenty jsou v indexu reprezentovány nejen slovy, ale i výskytem těchto typů. Součástí dotazu je pak požadavek na typ entity, vyplývající z očekávaného typu odpovědi. Vyhledávací dotaz pro otázku *„Ve kterém roce se narodil Petr Chelčický?“* by tak mohl vypadat například jako *„narodit Petr Chelčický TYPE=ROK“*. Nevýhodou tohoto přístupu je především složitost zpracování a indexování potenciálně velkého zdrojového korpusu.

Pro některé typy odpovědí existují **charakteristická slova**, která velmi často tvoří danou odpověď, nebo se vyskytují poblíž. Typickým případem jsou otázky žádající hodnotu určité veličiny – např. *„Jaká je rychlost raketoplánu?“* nebo *„Jak rychle létá raketoplán?“*. Při řešení tohoto druhu otázek může zásadně pomoci do vyhledávacího dotazu začlenit příslušné jednotky dané veličiny (např. *„kilometrů v hodině“*, *km/h*, ms^{-1} apod.), protože budou velmi pravděpodobně součástí odpovědi. Tímto způsobem expandují vyhledávací dotaz například Pinchak a Bergsma [30], kteří jednotky pro různé veličiny získávají automaticky přímo z webu.

Zvyšování přesnosti

Vyhledávání pomocí slov otázky a přidaných slov dosahuje obecně dobrého pokrytí, výsledný dotaz však může být poměrně nepřesný a vracející velké množství nesouvisejících segmentů. Opačných vlastností lze dosáhnout, pokud bude dotaz tvořen čistě předpokládaným tvarem odpovědi, odhadnutým podle otázky, například:

Ve kterém roce vypukla Třicetiletá válka? → *Třicetiletá válka vypukla roku ČÍSLO.*

Textové segmenty obsahující uvedenou větu budou velmi pravděpodobně relevantní a navíc v tomto konkrétním příkladě už je šablonou jednoznačně definován způsob extrakce odpovědní entity *(ČÍSLO)*, takže je výrazně usnadněna extrakční fáze QA systému (viz oddíl 2.3.3). Na druhou stranu může být naivní očekávat, že se ve zdrojovém korpusu vyskytuje právě takováto formulace odpovědi. Tento přístup nachází uplatnění buď v kombinaci se standardním přístupem – vyhledávači jsou nejprve položeny takto specifické dotazy a v případě neúspěchu je hledáno klíčovými slovy – nebo je možné uplatnit jej tam, kde je zdrojový korpus dostatečně velký a redundantní.

Dostatečně obsáhlý a redundantní se z hlediska této metody ukazuje zvláště web v anglickém jazyce. Toho využívá například systém QuALiM [10], který pomocí různých syntaktických vzorů transformuje vstupní otázku na výrok s odpovědí a ten pak hledá na webu vyhledávačem Google. Podobně postupují Agichtein et al. [24], kteří se zaměřují na několik typů otázek a s využitím strojového učení generují různé transformace otázky na efektivní dotaz obsahující fráze deklarativního charakteru (např. *„what is“* → *„refers to“*, *„named after“* atd.).

2.3.3 Extrakce kandidátních odpovědí

Z textových segmentů získaných vyhledávačem je dále zapotřebí extrahovat konkrétní entity, které mohou potenciálně představovat odpověď. Na realizaci této fáze má zásadní vliv očekávaný typ odpovědi – jiným způsobem bude extrahován jednoduchý fakt než například definice pojmu nebo vysvětlení příčiny.

2.3.3.1 Faktografické otázky

V souvislosti s faktografickými otázkami má význam především koncept tzv. **pojmenované entity**. Pojmenovanou entitou rozumíme výrazy jednoznačně odkazující na nějaký objekt, událost, časový okamžik apod. [31, str. 6] Uvažme například následující text:

„Konec svého života strávil v Holandsku. Zde vydává soubor svých 43 spisů pod názvem Opera didactica omnia. Vznikají zde i menší díla a dílo Jedno potřebné bývá považováno za jeho závěť lidstvu. Zemřel 15. listopadu 1670 a byl pohřben v kostelíku v Naardenu.“

V uvedené pasáži se vyskytuje hned několik druhů pojmenovaných entit, ať už jde o místní názvy (*Holandsko*, *Naarden*), názvy literárních děl (*Opera didactica magna*, *Jedno potřebné*) nebo například časový údaj (*15. listopadu 1670*). Každá z těchto entit může potenciálně představovat odpověď na nějakou faktografickou otázku, je tedy zapotřebí, aby QA systém dovedl pro očekávaný typ odpovědi (viz oddíl 2.3.1) extrahovat z textového segmentu příslušné pojmenované entity.

Rozpoznání pojmenovaných entit v textu je samostatnou oblastí zpracování přirozeného jazyka (viz např. [31]), při čemž pro rozpoznání se užívají buď ručně sestavené **vzory**, postihující obsah či kontext entity, nebo **statistické metody** a strojové učení. Některé přístupy si vystačí s povrchovou strukturou textu, jiné vyžadují například morfologickou či syntaktickou analýzu.

Různé QA systémy se liší propracovaností rozpoznávání i počtem rozpoznatelných kategorií pojmenovaných entit. Zatímco starší systémy prezentované v rámci konference TREC pracovaly obvykle s nižším počtem kategorií, modernější systémy dovedou často s využitím externích značkovačů entit identifikovat velké množství různých typů. Například systém Lasso [12] z roku 1999 používá pro extrakci pojmenovaných entit gramatická pravidla a různé heuristiky, kterými identifikuje pouze jména osob, organizací, výrobků, míst, data a peněžní částky. Chaucer-2 [17] z roku 2007 dokáže externím modulem označit více než 300 různých typů pojmenovaných entit a k tomu užívá ještě 500 webových lexikonů a zeměpisných slovníků, jimiž zvyšuje pokrytí identifikovaných výrazů. K extrakci pojmenovaných entit přistupuje však až poté co selžou ručně sestrojené vzory pro rozpoznání odpovědi.

Některé externí značkovače pojmenovaných entit dosahují dobrých výsledků, ale pracují s příliš malým počtem příliš širokých kategorií (např. jen osoba, místo, číslo, rok…), jenž pro potřeby QA systémů nedostačuje [31, str. 18]. Tento problém řešili například Han et al. [32], kteří k podporovaným kategoriím zavedli navíc podkategorie a pro jejich rozpoznání sestavili poloautomaticky vlastní slovník, tvořený pojmy nalezenými ve zdrojovém korpusu.

2.3.3.2 Složitější otázky

Pro extrakci jiných typů odpovědí než jednoduchých faktů neexistuje tak přímočaré univerzální řešení, jakým je rozpoznání pojmenovaných entit.

Jedním z přístupů, nastíněným již na konci oddílu 2.3.2.2, je transformace otázky na předpokládaný tvar odpovědi. Triviálně toho lze dosáhnout například jednoduchým převodem **syntaxe** otázky na syntaxi oznamovací věty:

Kdo je to Nicolas Sarkozy? → Nicolas Sarkozy je …
Proč vypukla druhá světová válka? → Druhá světová válka vypukla, protože …
Jaký je vztah mezi Českem a Slovenskem? → Vztah mezi Českem a Slovenskem je …

Takováto transformace je samozřejmě velmi naivní a disponuje obvykle nízkým pokrytím hledání. Pro některé typy otázek však po vhodném rozpracování (přidáním různých parafrází apod.) může generovat obstojné výsledky. Příkladem je třeba již popisovaný systém QuALiM [10], který pro transformaci otázky na odpověď zavádí velké množství syntaktických vzorů, například následující (zjednodušeno):

`1:When 2:did 3:NP 4:V 5:NP` → `3 4 5 in DATE`

(*např. When did Amtrak begin operations?* → *Amtrak began operations in DATE*)

Syntaxi zdrojového textu užívají pro extrakci odpovědí také Ferret et al. [13], jejichž systém nejprve pomocí různých vzorů rozpoznává syntaktický model otázky (např. `What-do-NP-VB`) a podle něj vytváří vzor odpovědi, kterým je schopen řešit například otázku *„What do Knight Ridder publish?“ („Co vydává Knight Ridder?“*), což je v podstatě faktografická otázka, ale kvůli neznalosti typu odpovědi ji nelze řešit pojmenovanou entitou.

Od roku 2003 se věnovala v rámci konference TREC pozornost definičním otázkám (typu *„Kdo je to Nicolas Sarkozy?“*). Není-li pro dané téma otázky (*Nicolas Sarkozy*) k dispozici encyklopedické heslo, které by přímo nabízelo příslušnou definici tohoto pojmu, ukazuje se rovněž vhodné využít syntaktických vzorů a extrahovat „definici“ pojmu ze zdrojového textu. Například Han et al. [32] vybírají ze zdrojového textu věty obsahující téma otázky a následně v nich hledají především jmenné a slovesné fráze modifikující toto téma (např. *„…prezident Nicolas Sarkozy…“*).

Mnohé otázky však syntaktickým přístupem řešit skoro nelze. Náročně řešitelné jsou kupříkladu otázky typu PROČ (např. *„Proč vypukla Druhá světová válka?“*) nebo otázky typu JAK+SLOVESO (např. *„Jak se vyrábí zubní pasta?“*). Uvažme, že například na první zmíněnou otázku lze teoreticky odpovědět jak krátkým souslovím (např. *„...kvůli nespokojenosti některých mocností…“*), tak podrobnějším vysvětlením, tvořeným více větami. Přestože je možné například jednoduše detekovat určitá klíčová slova (*kvůli*, *protože*, *důvod*, *příčina*…) a extrahovat jmenné fráze poblíž, ztěží bychom tímto způsobem dosáhli kvalitních výsledků. Pro zodpovězení vysvětlovacích otázek tohoto typu je zapotřebí hlubší analýza textu, která umožní odhalit příslušné vztahy, jež jsou předmětem otázky. Tento postup ukazují například Verberne et al. [33], kteří se zaměřují na otázky typu PROČ a řeší je tzv. **analýzou promluvy**, založenou na korpusu, v němž jsou ručně označkovány vztahy jako příčina, účel, okolnost, podmínka apod.

Je třeba podotknout, že výzkum v oblasti zodpovídání složitějších otázek je v podstatě stále na počátku. Extrakce některých typů odpovědí může vyžadovat hloubkovou analýzu textu nebo zapojení externích znalostí a v jistých případech může být dokonce zapotřebí extrahovat více informací z různých míst a nějakým způsobem tyto informace spojit.

2.3.4 Výběr správných entit

Nejsou-li odpovědní entity extrahovány ze zdrojového textu přímo pomocí vzorů vzniklých transformací otázky, nebo nějakou hloubkovou metodou zajišťující jejich relevanci, musí být tyto entity dále analyzovány s cílem rozhodnout, která nebo které z nich tvoří nejpravděpodobnější odpověď. Typickým výstupem extrakční etapy je množina kandidátních entit (např. ve formě frází či pojmenovaných entit) nalezených v textových segmentech jen na základě jednoduchých kritérií – nejčastěji podle očekávaného typu odpovědi. Musí tedy následovat **hodnocení** entit a jejich kontextu, které rozhodne, nakolik jednotlivé entity skutečně vyhovují položené otázce.

Hodnocení entit může být stejně jako ostatní fáze realizováno buď hloubkovými lingvistickými metodami, nebo povrchově například jen na lexikální úrovni.

2.3.4.1 Hloubkové metody

Hloubkové metody produkují obecně kvalitnější výsledky, ale jsou náročné na výpočetní výkon a dostupné lingvistické zdroje [35, str. 4]. Výhodnou pozici má v tomto smyslu angličtina, na níž se upírá v oblasti výzkumu zpracování přirozeného jazyka největší pozornost, takže pro ni existuje spousta lingvistických zdrojů (korpusů, lexikonů apod.) a nástrojů. Díky tomu mohly v uplynulých letech vzniknout kvalitní QA systémy analyzující text například až na sémantické úrovni.

Vhodným příkladem je třeba již zmíněný systém PowerAnswer (viz [20, 21]), který v rámci konferencí TREC 2006 a 2007 dosahoval nejlepších výsledků pro faktografické otázky. Vstupní otázku i kontext odpovědní entity převádí na logické vyjádření a hodnocení provádí tzv. **abduktivním odvozováním** (inferencí) s využitím znalostí uložených ve Wordnetu. Například otázku *„Which company created the Internet Browser Mosaic?“* reprezentuje logickým formalismem jako:

```
-(exists e1 x2 x3 x5 x6 (_organization_at(x2) & company_nn(x2) &
create_vb(e1,x2,x6) & internet_nn(x3) & browser_nn(x4) &
mosaic_nn(x5) & nn_nnc(x6,x3,x4,x5)))
```

Podobným stylem vyjadřuje i větu s kandidátní odpovědí a přidává různé logické axiomy představující znalosti okolního světa (world axioms) nebo jazykové jevy (NLP axioms). Z výsledné soustavy výroků se pak pokouší dokázat správnost odpovědi. [34]

2.3.4.2 Povrchové metody

Povrchový přístup k hodnocení entit se realizuje snadněji než hloubkový, nevyžaduje propracované lingvistické zdroje a je flexibilnější, pokud jde například o přechod na jiný jazyk. Z těchto důvodů se jeví také vhodnější pro použití v rámci této práce.

Při povrchovém hodnocení se obvykle posuzují dva aspekty – kontext entity a počet výskytů entity v nalezených textových segmentech. **Kontext** se neanalyzuje po sémantické stránce, ale hodnotí se především existence různých slov poblíž odpovědi, neboť je žádoucí, aby se zde nacházela klíčová slova otázky a jiné výrazy, které mohly tvořit již vyhledávací dotaz (viz oddíl 2.3.2.2). Důležitost **počtu výskytů** kandidátní odpovědi zase vychází z přesvědčení, že častěji nalezená odpověď je správnější. Pro hodnocení kontextu i počtu výskytů existují různé výpočetní metriky, přehledně shrnuté a srovnávané například v práci [35]. Následuje popis základních metrik, které bývají v praxi různě modifikovány a kombinovány.

Podíl klíčových slov (keyword overlap) – Hodnotí entitu podle počtu klíčových slov nalezených v okolí. Označíme-li množinu klíčových slov jako K a množinu slov z pevně definovaného okolí entity jako P, může být hodnocení vypočteno například tzv. **Jaccardovým koeficientem**, vyjadřujícím míru překrytí dvou množin [23, str. 59]:

$$J(K,P) = \frac{|K \cap P|}{|K \cup P|} \tag{2.1}$$

Jde tedy o počet poblíž nalezených klíčových slov, normalizovaný jak délkou otázky, tak délkou pasáže s odpovědí.

Vzdálenost od klíčových slov – Vychází z předpokladu, že kandidátní odpovědi jsou tím relevantnější, čím blíže se u nich nacházejí klíčová slova. Toto hodnocení může být počítáno různými způsoby, například jako převrácený průměr vzdáleností (d_k) jednotlivých klíčových slov od odpovědní entity:

$$E = \frac{|K|}{\sum_{k \in K} d_k} \tag{2.2}$$

Obě výše uvedené metriky je možné různým způsobem modifikovat. Kromě klíčových slov jako takových mohou být při hodnocení kontextu zohledněny také bigramy slov, nejdelší možné posloupnosti slov, interpunkce, jednoduché syntaktické jevy apod. Při počítání klíčových slov nebo měření vzdáleností je možné také přiřadit k různým slovům různé váhy podle toho, jak důležitou hraje slovo v otázce roli, nebo podle toho, jaká je jeho obecná výpovědní hodnota – ta může být odhadnuta například pomocí invertované dokumentové frekvence [23, str. 116]:

$$IDF(w) = \frac{N}{df(w)} \tag{2.3}$$

kde N je počet dokumentů ve zdrojovém korpusu a df je počet dokumentů, v nichž se vyskytuje slovo w.

Hodnocení výskytů slov v kontextu entity může být také doplněno o syntaktickou analýzu a srovnání stavby vět kontextu se stavbou otázky. To činí například systém Palantir [20] a systém KUQA [32], které při hodnocení užívají jak povrchovou vzdálenost od klíčových slov, tak syntaktickou podobnost otázky a odpovědi.

IR skóre – Hodnotí entitu čistě podle skóre textového segmentu, v němž byla entita nalezena. Toto skóre určuje při hledání vyhledávací stroj a vychází z počtu výskytů jednotlivých klíčových slov v segmentu. Každé kandidátní odpovědi z téhož segmentu je tedy přiřazeno stejné hodnocení, což je naivní přístup, nacházející uplatnění buď v kombinaci s jinými metrikami, nebo u systémů, které pracují s velmi malými segmenty. Na hodnocení segmentů klade důraz například systém AnwerBus [8], který za segmenty považuje jednotlivé věty.

Frekvence odpovědi – Vyjadřuje počet nalezených výskytů odpovědní entity. Těží z redundance zdrojového korpusu (např. webu) a předpokládá, že čím vícekrát byla nalezena určitá odpověď, tím je správnější. S tím je však spojen problém srovnávání dvou různých podob též odpovědi. Uvažme, že jsou-li hledanou odpovědí například *Spojené státy americké*, je zapotřebí při počítání frekvence zahrnout i entity jako *„Spojené státy“*, *„USA“*, *„U.S.A.“* atd. Pro tyto účely se používají jednak synonyma slov a celých pojmenovaných entit (extrahovaná z různých slovníků, encyklopedií apod.) a jednak různé míry podobnosti řetězců – například **Levenshteinova vzdálenost** [22, str. 56], vyjadřující počet znakových změn, které je třeba učinit pro transformaci jednoho řetězce na druhý.

Triangulace entit – Je založena na vzájemném posilování jednotlivých kandidátních entit podle toho, nakolik jsou sémanticky blízké. Triangulaci navrhli Roussinov a Robles [36], kteří pro hodnocení sémantické podobnosti dvou entit (*sim(a,b)*) použili jednoduše vzájemné překrytí slov, počítané obdobně jako výše popsaný Jaccardův koeficient. Idea je jakýmsi vylepšením počítání frekvence odpovědi, oproti které se vyznačuje větší flexibilitou a vhodností pro hodnocení delších nefaktografických odpovědí. Hodnocení entity *a* posílené ostatními nalezenými entitami se vypočítá jako:

$$s^t(a) = \sum_{a_i \in O, a_i \neq a} s(a_i) \cdot sim(a, a_i) \tag{2.4}$$

kde *O* je množina všech kandidátních entit, *sim(a,b)* vyjadřuje podobnost dvou entit a s*(a)* je původní hodnocení entity *a* (stanovené jinými metrikami). [36]

Kookurence otázky a odpovědi – Hodnotí odpověď podle toho, jak často se ve zdrojovém korpusu vyskytuje společně s klíčovými slovy otázky. Tento způsob využití redundance navrhli Magnini et al. [37], kteří nechávají odpověď a klíčová slova vyhledat na webu a pro hodnocení jejich společného výskytu (kookurence) zkoušejí počítat například tzv. **bodovou vzájemnou informaci** (Pointwise Mutual Information, PMI), z níž vyplývá následující vzorec pro hodnocení odpovědi:

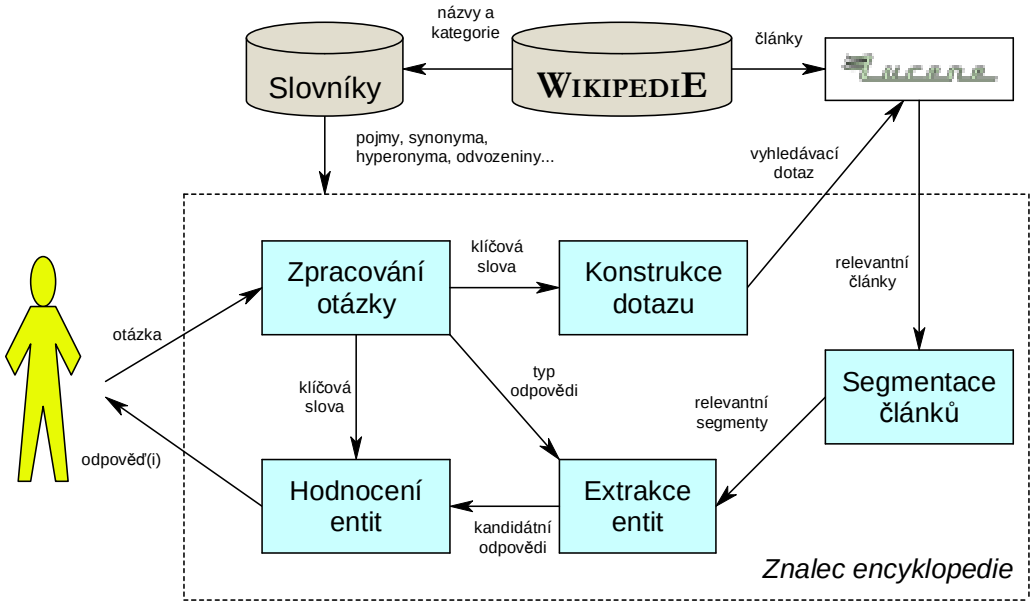
$$s(A) = \frac{hits(Q \text{ NEAR } A)}{hits(Q) \cdot hits(A)} \cdot MAX \tag{2.5}$$

kde *hits(Q)*, resp. *hits(A)* je počet nalezených dokumentů obsahujících klíčová slova otázky, resp. odpovědi, *hits(Q NEAR A)* je počet dokumentů obsahujících obojí poblíž a *MAX* je maximální počet dokumentů, který je vyhledávač schopen vrátit. [37]

3 Realizace Znalce encyklopedie

Cílem této práce je sestrojit s využitím výše nastíněných poznatků **český** odpovědní systém, který bude jako zdroj informací využívat encyklopedii **Wikipedie**[1]. Systém ponese pracovní název Znalec encyklopedie a po implementační stránce bude založen na platformě **Java**[2].

Tato kapitola popisuje různé aspekty vývoje Znalce encyklopedie, především způsob zpracování zdrojového textového korpusu a řešení jednotlivých fází činnosti nezbytných pro nalezení odpovědi (viz obrázek 3.1). Systém bude primárně zaměřen na klasické faktografické otázky víceméně z otevřené domény a k textu bude přistupovat jen s využitím povrchových metod, nenáročných na dostupnost kvalitních lingvistických zdrojů.



Obrázek 3.1 Schéma činnosti řešeného systému.

3.1 Lingvistické zdroje

Navzdory povrchovému přístupu potřebuje program ke své činnosti alespoň základní lingvistické informace, které čerpá z externích slovníkových zdrojů. Pro integraci veškerých slovníků užívá program knihovnu **`fsa`** [38], která dovede slovník v textové podobě převést na konečný automat, umožňující efektivní vyhledání daného pojmu. Samotné hledání je pak realizováno knihovnou **`morfologik.fsa`** [39], která je na rozdíl od `fsa` navržena přímo pro Javu.

3.1.1 Morfologický slovník

Morfologický slovník je nejdůležitějším lingvistickým zdrojem. Mapuje slovní tvary na potenciální lemmata a gramatické kategorie, vyjádřené jednoduchými značkami. Program používá slovník, který

[1] http://cs.wikipedia.org

[2] http://java.sun.com

je produktem českého morfologického analyzátoru **ajka** [40]. Záznamy pro slovní tvar „*svetru*“ vypadají například následovně:

```
svetru#svetr#k1gInSc2   svetru#svetr#k1gInSc3   svetru#svetr#k1gInSc6
```

Za lemma tohoto tvaru je považováno slovo *svetr*, jedná se o substantivum (`k1`), rodu mužského neživotného (`gI`), v jednotném čísle (`nS`) a ve 2., 3. nebo 6. pádě (`c2`/`c3`/`c6`).

Morfologický slovník umožňuje vyhledávání slov v textu podle lemmat, tedy nezávisle na ohýbání slov – nezbytná vlastnost systému pracujícího s flektivním jazykem. Informace o slovním druhu je užitečná v různých etapách činnosti systému (pro odlišení funkčních slov od významových, hledání fokusu otázky, rozeznání důležitých slov v textu apod.). Ostatní gramatické značky jsou využívány v menší míře (např. pro zpracování kategorií).

Program se snaží řešit i situaci, kdy narazí na tvar, který se nevyskytuje ve slovníku – má k dispozici ručně předdefinovaný seznam koncovek, pomocí nichž odhaduje lemmata neznámých slov, typicky vlastních jmen.

3.1.2 Slovník synonym

Slovník synonym poskytuje k příslušným slovům pojmy stejného nebo velmi podobného významu. Nachází uplatnění při expanzi vyhledávacího dotazu (viz oddíl 3.4.1) a při analýze výskytů klíčových slov (viz oddíl 3.6), kdy je akceptován nejen výskyt daného slova, ale i kteréhokoliv jeho synonyma.

Program užívá slovník, který vznikl sloučením volně šiřitelného **thesauru** [41] pro Open Office a synonym extrahovaných z českého **Wordnetu** [42]. Následuje ukázka formátu slovníku:

```
jízda##řízení|vyjížďka|jezdectvo|
jízdenka##los|vstupenka|lístek|
jízdní kolo##kolo|bicykl|
```

S používáním slovníku je bohužel spjat problém polysémie – například u slova *jízda* jsou tři synonyma, ale ta mají vztah ke třem odlišným významům slova. Při používání slovníku tak nutně dochází ke zkreslení výsledků.

3.1.3 Derivační slovník

Derivační slovník slouží jako zdroj slov, které jsou s daným výrazem morfologicky příbuzné, ale nikoliv na úrovni flexe (jako morfologický slovník), nýbrž procesem odvozování založeným například na přítomnosti odlišné předpony či přípony.

Použitý slovník, vycházející z práce [43], obsahuje nejen příslušné odvozeniny, ale také číselné kódy udávající typ slovotvorného procesu (např. 51 – ženské přivlastňování), kterým odvozenina vznikla:

```
Pavlína##Pavlínin|51
diplomat##diplomatův|58|diplomatka|340|diplomatický|334
zazpívat##zazpíván|705|zazpívání|316|zazpívaný|706|zazpívající|707
```

Derivační slovník tedy mapuje slovo na odlišné výrazy (třeba i jiný slovní druh), ty však bývají sémanticky více či méně blízké původnímu slovu, takže jsou vhodné pro podobné účely jako synonyma.

3.1.4 Wikipedie jako slovník pojmů

Vzhledem k tomu, že je Znalec encyklopedie primárně zaměřen na faktografické otázky, jejichž odpovědi mají obvykle tvar pojmenované entity (viz str. 14), je třeba tyto pojmenované entity nějakým způsobem v textu rozpoznávat.

Při vývoji nebyl k dispozici žádný český rozpoznávač pojmenovaných entit, bylo tedy nutné vyvinout vlastní. Pro tyto účely se ukázalo vhodné využít jako slovník pojmenovaných entit samotnou Wikipedii. Jak uvádí například Yu et al. [44], představuje Wikipedie vlastně jednoduchou **ontologii**, definující různé pojmy (jeden pojem – jeden článek) a vztahy mezi nimi (pomocí kategorií, odkazů apod.). Přestože nejde o ontologii formální a přesnou, její rozsah a neustálá aktualizace přispěvateli z ní činí vhodný lingvistický zdroj.

Slovník názvů

Pro co nejefektivnější vyhledávání pojmenovaných entit v textu jsme se rozhodli vytvořit konečný automat obsahující názvy článků Wikipedie. Za relevantní jsou přitom považovány jen články hlavního jmenného prostoru bez výčtových článků. Aby bylo možné příslušné pojmy v textu rozeznat i v jiném než základním tvaru (např. *Česká republika – Českou republikou* apod.), uskutečňuje program lemmatizaci jednotlivých slov názvu. Do slovníku jsou pak pro daný název uloženy všechny možné kombinace lemmat. Pokud dále název obsahuje rozlišovač v závorce, je uložen s tímto rozlišovačem i bez něj. Například pro článek *„Lysá hora (1323 m)“* tak ve slovníku existují tyto čtyři záznamy:

```
lysat hora 1323 m     ##ULysá hora (1323 m)
lysat hora            ##ULysá hora (1323 m)
lysý hora 1323 m      ##ULysá hora (1323 m)
lysý hora             ##ULysá hora (1323 m)
```

Z výrazů před křížkem je sestaven konečný automat, umožňující rychlé nalezení (či nenalezení) příslušného názvu článku, uvedeného za křížkem. Je tedy možné snadno podle skupiny lemmatizovaných slov v textu rozpoznat pojmenované entity, aniž by bylo třeba uskutečňovat přesnou lemmatizaci či syntaktickou analýzu. Například:

> *Podle pověsti se na Lysé hoře slétaly čarodějnice z celého Slezska a k smrti utancovaly každého zabloudilce, který se připletl k jejich rejům.*

Ke každé entitě náleží **definiční článek**, podle jehož kategorií může systém určit typ entity (viz dále). Vlivem lemmatizace a kvůli existence stejnojmenných článků s rozlišovačem však může být pro entitu nalezeno více různých článků. Systém tedy názvy vrácené konečným automatem ještě srovnává s daným souslovím a určuje, zda se skutečně liší jen mluvnickým pádem a číslem, což z nich činí tentýž pojem, jen v jiném tvaru.

Určitým neduhem názvů článků na Wikipedii je fakt, že všechny povinně začínají velkým písmenem. To zvyšuje nejednoznačnost rozpoznávání entit, jelikož sousloví, která jsou v textu zřetelně uvozená malým, respektive velkým písmenem, mohou být nesprávně přiřazena i k pojmům začínajícím na opačné písmeno (např. zvíře *ježek* × značka piva *Ježek*). Tento problém se systém snaží redukovat určením velikosti počátečního písmene z obsahu článku. Při vytváření slovníku názvů hledá program výskyty názvu daného článku v samotném textu a v případě, že nalezne jednoznačný výskyt

názvu s malým či velkým písmenem (mimo počátek věty), přidává tuto informaci do slovníku (značka U/L/?).

Hyperonymní kategorie

U klasických faktografických otázek vyjadřuje očekávaný typ odpovědi velmi často **hyperonymum** (nadpojem) hledané entity. Například pro otázku *„Ve kterém městě se narodila Božena Němcová?“* hledáme jako odpověď entitu, jejíž hyperonymum je *město*. Tento druh sémantického vztahu lze „vyčíst“ z kategorií Wikipedie. Při jejich bližším zkoumání totiž vychází najevo, že kategorie daného článku velmi často odkazují právě na hyperonyma pojmu, který je článkem popsán. Například článek *Vídeň* patří do následujících kategorií:

Světové dědictví (Rakousko), Vídeň, Města v Rakousku
Rakouské spolkové země, Hlavní města v Evropě

Pro rozpoznávání pojmenovaných entit mají význam podtržené kategorie, které budeme nazývat **hyperonymní kategorie**. Vyjadřují vztah typu *„POJEM je OBECNĚJŠÍ_POJEM“* (například *„Vídeň je město v Rakousku“* nebo *„Vídeň je rakouská spolková země“*). Jiné kategorie mohou zachycovat holonymní vztahy – *„POJEM je_součástí POJEM“* (např. *„Vídeň je součástí Světového dědictví“*) – nebo prosté asociace typu *„POJEM souvisí_s POJEM“*.

Hyperonymní kategorie je do značné míry možné od ostatních odlišit podle toho, že mívají název ve tvaru **množného čísla** (*rakouské země, hlavní města*…). Znalec encyklopedie tedy při indexaci u každého článku analyzuje názvy kategorií a pomocí jednoduchých heuristik vybírá ty kategorie, jejichž hlavní substantivum je v množném čísle. Ty pak indexuje pro daný článek ve speciálním poli HYPERNYM (viz str. 24). Do tohoto pole neumisťuje jen přímé hyperonymní kategorie, ale tranzitivně i všechny jejich hyperonymní nadkategorie, čímž pro daný pojem získává přehled nadpojmů s různou úrovní abstrakce (např. *Vídeň* → *Hlavní města v Evropě* → *Hlavní města* → *Města* → *Obce*).

V nejrůznějších případech však rozpoznání hyperonymní kategorie selhává. Některé kategorie sice nesou název v podobě množného čísla, tento název však ve skutečnosti představuje jednotlivý pojem (např. *Karlovy Vary*, *Simpsonovi* apod.). Pro tyto účely byl sestaven soubor nastavení, který uvádí seznam výjimek (cca 500). Existuje řada kategorií, u jejichž názvu nelze stanovit, zda se jedná o jednotné či množné číslo (např. *Historie*, *Povodí* apod.). V takovém případě se systém pokouší plurál určit z koncového *-s* v anglickém názvu kategorie. Pokud ani pak neuspěje, kategorie je považována za nehyperonymní. Jsou však také případy, kdy je jistá kategorie A z povahy hyperonymní a její nadkategorie B rovněž, avšak vztah mezi nimi hyperonymní není. Tento problém se vyskytuje hlavně u jmen národů či rodů – například *Svatý Václav* je řazen do kategorie *Přemyslovci* a ti jsou v kategorii *Šlechtické rody*, nelze však tvrdit, že Svatý Václav je šlechtický rod. Byl tedy zaveden další soubor nastavení, který vylučuje z indexačního procesu jen některé konkrétní vazby mezi kategoriemi.

Synonyma podle přesměrování

Přesměrování (redirecty) jsou mikročlánky Wikipedie, které mapují určitý pojem na jiný existující článek. Mohou být tvořeny z různých technických i uživatelských důvodů, ale jejich nejčastějším smyslem je pokrýt pro daný článek rozličné alternativní názvy – tedy v podstatě **synonyma** daného pojmu. Například na článek *Česko* vedou přesměrování jako *Česká republika*, *ČR*, *Český*, *České země* apod.

Znalec encyklopedie ukládá při indexaci do speciálního pole `REDIRECT` názvy všech přesměrování, která vedou na daný článek, a umožňuje tak jednak hledání článků podle synonym a jednak extrakci synonym pro konkrétní pojem.

3.2 Indexace zdrojových dat

Data české jazykové mutace Wikipedie byla stažena ve formátu XML z příslušného webového úložiště[1]. Jako vyhledávací stroj byl zvolen **Lucene**[2], open-source systém vyvinutý pro použití v jazyce Java. Jeho indexační mechanismus je založen na použití tzv. **polí** (fields), které umožňují odděleně indexovat různé informace o daném článku – je tedy možné rozlišovat například mezi hlavním textem, nadpisem, textem z tabulek apod. U každého pole se lze také rozhodnout, zda do něj bude uložen text v nezměněné podobě, nebo zda bude „tokenizován“ na jednotlivá slova, což umožňuje vyhledávání podle slov, ale vede k zanedbání interpunkce apod.

Pro účely této práce byly jednotlivé články při indexaci rozděleny do následujících polí:

`NAMESPACE` – Jmenný prostor neboli typ článku (normální/kategorie/šablona).

`TITLE` – Název článku.

`TEXT` – Text článku.

`REDIRECT` / **`REDIRECT_UNTOKENIZED`** – Tokenizované/netokenizované znění přesměrování, která vedou na daný článek.

`HYPERNYM` – Přimé i nepřímé hyperonymní kategorie článku.

Proces indexace se skládá z následujících kroků:

1) Jednoduchým XML analyzátorem (SAXParser[3]) je uskutečněn **první průchod** zdrojovým souborem. Při něm jsou do paměti uloženy jen údaje o přesměrování a o příslušnosti kategorií do nadkategorií.
2) Přesměrování jsou převedena na **zpětná přesměrování** – ke každému článku je vygenerována množina přesměrování, která na něj vedou.
3) Iterativním algoritmem je pro každou kategorii sestavena množina přímých i nepřímých **hyperonymních nadkategorií**.
4) Při **druhém průchodu** XML souborem je vytvářen samotný index článků. Na základě dat extrahovaných z XML souboru a údajů získaných v kroku 2 a 3 jsou pro každý článek naplněna jednotlivá indexační pole (viz výše). Volitelně je také vygenerován slovník názvů, popsaný v oddíle 3.1.4 .
5) Výsledný index je optimalizován pro efektivní použití (zajišťuje přímo Lucene).

[1] http://download.wikimedia.org

[2] http://lucene.apache.org

[3] javax.xml.parsers.SAXParser

3.3 Zpracování otázky

Smyslem zpracování otázky je rozpoznat uživatelovy požadavky na hledanou odpověď. Otázka je nejprve rozdělena na jednotlivá slova a jsou identifikovány pojmenované entity, čímž dojde k pomyslnému seskupení některých slov. Pro detekci pojmenovaných entit používá program mechanismus popsaný v oddíle 3.5.1.

Znalec encyklopedie klade důraz na **tázací slovo** a případný **fokus otázky**, pomocí nichž odhaduje očekávaný typ odpovědi. Tabulka 3.1 ilustruje základní modely otázek, které program rozpoznává – výrazy v hranaté závorce představují lemmata slov, lomená závorka určitý slovní druh, `<NE>` pojmenovanou entitu a `<F>` označuje fokus otázky. Je ovšem třeba poznamenat, že rozpoznání toho kterého modelu otázky ještě neznamená, že program dovede otázku daného typu zodpovědět.

Vzor otázky	Očekávaná odpověď	Vzor otázky	Očekávaná odpověď
`kdo [být] <NE>?`	definice osoby	`[čí] … <F> …?`	vlastnictví F
`co [být] <NE>?`	definice věci	`<prep> [čí] … <F> …?`	vlastnictví F
`kdo [být] … <F> …?`	fokus – osoba	`[kolik] … <F> …?`	počet F
`co [být] … <F> …?`	fokus – věc	`<prep> [kolik] … <F> …?`	počet F
`[kdo] …?`	osoba	`kde/odkud/kam/kudy …?`	místo
`[co] …?`	věc	`kdy/odkdy/dokdy …?`	čas
`<prep> [kdo] …?`	osoba	`jak <adj>/<adv> ...?`	adj/adv fokus
`<prep> [co] …?`	věc	`jak ...?`	způsob
`[jaký|c1] [být] <NE>?`	vlastnosti	`proč ...?`	důvod, příčina
`[jaký]/[který] … <F> …?`	fokus	`<verb> ...?`	ano/ne
`<prep> [jaký]/[který] … <F> …?`	fokus		

Tabulka 3.1 Rozpoznávané modely otázek.

Při hledání fokusu otázky vychází program z předpokladu, že fokus je první **jmenná fráze** otázky odpovídající mluvnicky tázacímu slovu a případné předložce. Vzhledem k tomu, že program nepoužívá žádný externí syntaktický analyzátor, detekuje jmennou frázi vlastním algoritmem, který za jmennou frázi považuje jednoduše posloupnost mluvnicky shodných substantiv a adjektiv. Například v otázce *„Ve kterém anglickém městě se narodil William Shakespeare?“* je fokus tvořen jedním substantivem (**hlava fokusu**) a jedním adjektivem (modifikátor fokusu). Obě tato slova jsou navíc mluvnicky shodná se slovem *kterém* a vyhovují i předložce *ve* – program má k dispozici ručně sestavený slovník mapující předložky na pády (např. *ve* → 4, 6).

Jako další krok zpracování otázky program prochází jednotlivá slova a všechna užitečná slova (tedy nikoliv zájmena, předložky nebo spojky) kromě hlavy fokusu označuje za tzv. **klíčová slova**. Ta budou mít zásadní význam při hledání textu s odpovědí (viz oddíl 3.4) i při následném hodnocení kandidátních entit (viz oddíl 3.6). Zatímco tázací slovo a fokus vyjadřují očekávaný typ odpovědi, jednotlivá klíčová slova definují především podmínky a pojmy, jichž se odpověď týká. Ve výše uvedené otázce by například klíčovými slovy byly *anglickém*, *narodil* a *William Shakespeare* – pojmenovaná entita je považována za jediné „slovo“. Navíc jsou pojmenované entity v jednoduchých znalostních otázkách zpravidla dosti důležitá, takže je systém označuje za tzv. **nezbytná slova**, jejichž přítomnost v kontextu hledané odpovědi bude povinná. Pokud se v otázce nenachází žádná pojmenovaná entita, pak je za nezbytné označeno poslední klíčové slovo; výraz na konci otázky totiž mívá rovněž velmi často výsadní postavení.

Součástí systému je ručně předdefinované nastavení *questionToFocus*, které k některým druhům otázek přiřazuje tzv. **implicitní fokus**. Jedná se o soustavu jednoduchých pravidel pro klasifikaci

otázky, jejichž podstata byla nastíněna už na straně 9. Mapují otázku podle typu a volitelně podle fokusu či slovesa na implicitní fokus, který skutečně popisuje typ hledané odpovědi. Pravidla vypadají například takto:

```
TYP = místo → FOKUS = sídlo/stát/kontinent/stavba/místo…
TYP = počet && FOKUS = litr/barel/galon/pinta… → FOKUS = objem
TYP = počet && SLOVESO = vážit → FOKUS = hmotnost
TYP = způsob && SLOVESO = jmenovat/nazývat → FOKUS = PODMĚT
```

K dispozici je také nastavení *interrogativeToPreposition*, které mapuje některá tázací slova na **implicitní předložky**. Například:

```
kde → blízko (2.p.), kolem (2.p.), mezi (7.p.) …
dokdy → do (2.p.), po (4.p.)
```

Implicitní nebo explicitní předložky pak mohou pomoci při hodnocení kandidátních entit (viz oddíl 3.6) – naivně lze předpokládat, že se budou nacházet před správnou odpovědí.

3.4 Získávání segmentů s potenciální odpovědí

3.4.1 Položení dotazu vyhledávači

Pokud se v otázce vyskytuje pojmenovaná entita, je možné její definiční článek přímo zařadit do seznamu článků ke zpracování. Například u otázky *„Jaké je hlavní město Polska?“* není pochyb o tom, že se odpověď bude pravděpodobně nacházet v článku *Polsko*. Navíc u definičních otázek (typu *„Co to je Polsko?“*) je řešení triviální – systém prezentuje uživateli první odstavec článku *Polsko* a končí svou činnost. V ostatních případech přichází na řadu konstrukce vyhledávacího dotazu pro systém Lucene a snaha o nalezení co nejrelevantnějších článků.

Jádro dotazu tvoří jednotlivá **klíčová slova** otázky, rozvinutá o synonyma z thesauru i Wikipedie a o odvozeniny z derivačního slovníku. Vzhledem k tomu, že Lucene podporuje zvyšování významu libovolných slov dotazu (tzv. boosting), nabízí se hodnotit klíčová slova podle toho, jak důležitou hrají v otázce roli, což by mělo pomoci sestavit optimální vyhledávací dotaz. Znalec encyklopedie tuto myšlenku aplikuje tím způsobem, že zvyšuje význam pojmenovaným entitám začínajícím na velké písmeno – přiřazuje jim koeficient 1.5 (nárůst 50%).

Pokud jde o **fokus otázky**, na straně 12 bylo již vysvětleno, že jeho zahrnutí do vyhledávacího dotazu může mít v jistých případech smysl a v jiných nikoliv. Program ve své aktuální implementaci fokus v dotaze používá a vychází z předpokladu, že při dostatečném počtu zkoumaných výsledků hledání nemusí příliš vadit, že je správný článek vlivem nerelevantního fokusu ve výsledcích hlouběji. Do budoucna je však na zvážení, zda by nebylo vhodné nějakým způsobem odvíjet využití fokusu od charakteru otázky, nebo implementovat prediktivní anotaci článků (viz str. 14).

Hledaná odpověď se může nacházet buď v textu některého článku Wikipedie, nebo ji může představovat samotný pojem, jenž tvoří název článku. V takovém případě pomůže hledat fokus otázky v **hyperonymních kategoriích**. Například při zodpovídání otázky *„Který prezident byl zadržován v Mauthausenu?“* vede hledání článku, jenž popisuje prezidenta a obsahuje slovo *Mauthausen*, ihned ke správné odpovědi – *Antonín Novotný*. Program se však nemůže spoléhat na to, že každý článek obsahuje všechny možné informace související s daným pojmem, takže tento postup funguje spíše

jako doplněk ke standardnímu postupu, usnadňující někdy nalezení článku. Poznamenejme, že hyperonymní kategorie se podobají prediktivní anotaci, ovšem omezené na název článku.

Při experimentování s dotazy se ukázalo, že konvenční způsob generování dotazu může pro některé otázky vést k optimálním výsledkům, pro jiné však může být příliš volný a vracet před relevantním článkem spoustu jiných článků. Jeví se užitečné dotaz někdy zpřísnit vyžadováním povinné přítomnosti některých klíčových slov v nalezených článcích – Lucene tuto možnost podporuje pomocí operátoru +. Protože ale není jasné, u kterých otázek tvořit jak přísné dotazy, byla zvolena flexibilní implementace, trochu podobná přístupu systému Lasso [12]. Znalec encyklopedie začíná přísným dotazem, vyžadujícím přítomnost všech klíčových slov, a pokud jím nenajde dostatek článků, zkouší volnější dotaz, vyžadující jen pojmenované entity uvozené velkým písmenem. Posledním stupněm je pak nejvolnější dotaz, který netrvá na přítomnosti žádného slova.

Následuje příklad vyhledávacího dotazu, který vznikne při položení otázky *„Ve kterém anglickém městě se narodil William Shakespeare?“*:

```
+namespace:main ( "městě" OR "Město" OR "Měšťan" OR "Královské město" OR
"Poddanské město" OR "Kamerální město" OR "střed" OR "obec" OR "střed it" OR
"středek" OR "samosprávná obec" ) ( "anglickém" OR "anglicky" OR "anglický"
OR "anglickost" )  ( "narodil" OR "narodivší" OR "narozený" OR "narození" OR
"narodit" OR "narozen" )  +( "William Shakespeare" OR "William Shakespeare"
OR "Shakespeare" OR "Shakespear" )^1.5
```

Jde o středně přísnou variantu, vyžadující přítomnost entity *William Shakespeare*. Z dotazu je zejména patrné, jak jsou jednotlivá klíčová slova expandována o synonyma a odvozeniny. Poznamenejme ale, že například *Poddanské město* nedává jako synonymum města příliš smysl – přesměrování Wikipedie bohužel nejsou pro synonyma stoprocentním zdrojem.

3.4.2 Extrakce segmentů

Z článků nalezených pomocí systému Lucene je vybráno nejvýše 50 článků, v nichž se potenciálně nachází odpověď. Jednotlivé články však mohou být příliš dlouhé na to, aby se celé analyzovaly výpočetně náročnými metodami popsanými v následujícím oddíle. Proto se články dále dělí na **segmenty** a až v nich jsou hledány odpovědi.

Text článku získaný z indexu systému Lucene obsahuje veškeré formátování Wikipedie a spoustu elementů, které jsou pro extrakci odpovědi nepodstatné. Znalec encyklopedie nejprve pomocí ručně předdefinovaných regulárních výrazů z textu vyfiltruje obrázkové galerie, matematické vzorce, reference, šablony, tabulky, soubory, kategorie, interwiki a veškeré HTML značky.

Kromě prostého textu tak zůstávají prakticky jen nadpisy, seznamy, wikiodkazy a značky pro zvýraznění kurzívou nebo tučným fontem. Pro zpracování těchto prvků byla využita knihovna **java-wikipedia-parser** [45], poskytující jednoduchý analyzátor wikitextu. Při analýze článku program detekuje hranice odstavců a každý odstavec ukládá jako samostatný segment, který obsahuje čistý text a má přidruženou hierarchii nadpisů, například:

```
Al Pacino – Kariéra – Od roku 2000 po současnost (offset: 17)
```
V novém tisíciletí se Al Pacino blýskl ve filmu Insomnie, kde si zahrál veterána losangelské policie, thilleru o CIA Test (The Recruit), v adaptaci klasické divadelní hry Williama Shakespeara, Kupec benátský (The Merchant of Venice) a v dramatu Maximální limit (Two for the Money), kde ztvárnil postavu mocného šéfa obrovské národní korporace, která dává tipy na sportovní sázky.

Takto reprezentovaný segment je dostatečně ucelený a kompaktní, což jej činí vhodným pro extrakci faktografické odpovědi. Wikiodkazy a značky pro zvýraznění kurzívou jsou sice z textu vyfiltrovány, ale zůstávají uchované v paměti pro potřeby automatické extrakce entit (viz další oddíl).

Vzniklé segmenty jsou ohodnoceny podle výskytu klíčových slov a pro další zpracování je ponecháno maximálně 50 nejlépe hodnocených segmentů.

3.5 Hledání odpovědi v segmentech

V jednotlivých segmentech je dále zapotřebí vyhledat kandidátní entity, které mohou potenciálně představovat odpověď na danou otázku. Vzhledem k tomu, že je program zaměřen ve své aktuální implementaci na faktografické otázky, jedná se především o různé pojmenované entity a krátká sousloví.

3.5.1 Automatická extrakce entit

Rozpoznání entity v textu

Při hledání kandidátních entit v textovém segmentu postupuje systém po jednotlivých slovech a entitu rozpoznává čtyřmi metodami (seřazené podle priority):

a) Podle hranic **wikiodkazu** nebo **kurzívy**

Wikiodkazy a názvy zabalené do kurzívy představují výhodné prvky textů na Wikipedii. Explicitně vyznačují důležité pojmy článku a sdružují slova patřící k sobě. Systém je tedy používá jako jednoznačné vodítko pro stanovení hranic entity. Navíc v případě wikiodkazu již není třeba konečným automatem odhadovat definiční článek entity, neboť je přímo dán odkazem.

b) Podle hranic **uvozovek**

Na některých místech Wikipedie jsou použity pro vyznačení názvů i obyčejné uvozovky, proto jsou při extrakci entit také zohledněny.

c) **Víceslovná pojmenovaná entita**

Bez přítomnosti explicitních hranic jsou entity v textu rozeznávány podle existence stejnojmenného článku na Wikipedii (viz oddíl 3.1.4). Začíná-li aktuálně zpracovávané slovo na velké písmeno, zkouší systém sestavit pojem až z pěti slov uvozených tímto slovem (od nejdelšího po dvouslovný). Při shodě sousloví s názvem některého článku je úspěšně rozpoznána pojmenovaná entita. Zkoušeny jsou také dvojice slov, jejichž první slovo nezačíná velkým písmenem, ale je substantivem či adjektivem.

d) **Jednoslovná entita**

Jednotlivá slova jsou extrahována jako kandidátní entity v případě, že na Wikipedii existuje stejnojmenný článek, nebo jde stoprocentně o název – slovo začíná velkým písmenem a nenachází se na počátku věty.

Ověření typu entity

Článek na Wikipedii odpovídající názvem dané textové entitě považuje Znalec encyklopedie za tzv. **definiční článek** entity. Jak již bylo uvedeno na straně 22, může být k entitě nalezeno i více možných definičních článků. Systém se snaží tyto definiční články redukovat podle wikiodkazů nacházejících

se v kontextu entity. Jsou-li například pro textovou entitu *Mars* identifikovány definiční články *Mars (planeta)* a *Mars (mytologie)*, prohledá systém wikiodkazy článku, v němž byla entita *Mars* nalezena, a pokusí se nalézt odkaz, který by ukazoval na jeden z těchto definičních článků. Tímto způsobem dosahuje v některých případech zjednoznačnění významu entity.

> *Áres byl ctěn v Řecku mnohem méně než jeho římský protějšek [[Mars (mytologie)|Mars]].*
> *Obě tyto postavy však nejsou zcela totožné a Mars rozhodně byl bohem významnějším.*

Definiční článek (příp. definiční články) slouží jako zásadní zdroj informací o typu entity. Program srovnává hyperonymní kategorie (viz str. 23) definičního článku s fokusem otázky, představujícím očekávaný typ odpovědi, čímž rozpoznává, zda se jedná o relevantní typ entity. Pokud není nalezena žádná shoda (nebo nenáleží k entitě žádný definiční článek), program hledá fokus také uvnitř textové entity nebo v jejím bezprostředním okolí. Pokud je třeba z textu extrahovat například názvy hor, umožňuje tento postup i bez použití hyperonymních kategorií rozpoznat například entity *hora Říp* či *Lysá hora* podle přítomnosti fokusu *hora*.

3.5.2 Extrakce entit pomocí vzorů

Některé faktografické odpovědi nelze extrahovat pomocí Wikipedie jako slovníku ani není možné spoléhat na přítomnost fokusu v blízkosti entity.

Pro detekci různých **časových údajů** a **číselných veličin** používá program soustavu ručně definovaných vzorů vyjadřujících podobu hledané entity či její bezprostřední kontext. Program podporuje jednak klasické **regulární výrazy**, s jejichž pomocí lze přesně stanovit různé variace podoby entity, a jednak zavádí vlastní formát **slovních vzorů**, které dovedou do určité míry pracovat s morfologií slov – zejména klást požadavek na lemma či slovní druh.

O vzorech, které mají být použity pro extrakci odpovědi, rozhoduje soubor nastavení *focusToEntity*, který mapuje fokus otázky na konkrétní typ hledané entity, například:

plocha/rozloha/povrch → `area` *délka/dlouho* → `length, duration`

Vzory pro entitu typu `duration` (časové trvání) pak vypadají například takto:

```
@ *%|<k4> [milisekunda]|[sekunda]|[vteřina]|[minuta]|[hodina]|[den]|[týden]|[měsíc]|[rok]
[^\w-]((?:- ?)?\d(?:[ \d\.,]|tis\.|mil\.|mld\.)*? ?(?:ms|s|min|hod))[^\wěščřžýáíéúůďťňó]
```

Zavináčem je uvozen slovní vzor s jednoduchou syntaxí, který vyjadřuje požadavek na to, aby první slovo entity bylo buď číslo, nebo slovem psaná číslovka, a za ním má následovat libovolně vyskloňovaná časová jednotka. Druhý vzor je již klasickým regulárním výrazem, který nepracuje s ohýbáním slov, ale pomocí flexibilní syntaxe pokrývá případy odvíjející se od prosté struktury textu.

3.6 Ohodnocení kandidátních entit

Jednotlivé kandidátní entity extrahované z textových segmentů je třeba ohodnotit a vybrat z nich entitu, která nejpravděpodobněji představuje hledanou odpověď.

Znalec encyklopedie klade při hodnocení důraz na výskyt klíčových slov v pevně definovaném okolí kandidátní entity. Ve své aktuální implementaci tedy hodnotí jen kontext entity, nikoliv počet nálezů. Vychází z předpokladu, že Wikipedie je korpus s relativně vysokou spolehlivostí, ale nízkou redundancí.

Při hodnocení odpovědní entity se posuzují čtyři **kontextové oblasti** – název článku, v němž se entita nachází, nadpisy příslušející k segmentu, textové okolí o velikosti 20 slov a textové okolí o velikosti 5 slov. Definujeme-li funkci *in(e, A)*, která vrací 1 při výskytu textové entity *e* v oblasti *A* a v opačném případě vrací 0, pak se hodnocení výskytu **klíčových slov** (respektive entit, jelikož může jít i o pojmenované entity, chovající se jako jedno slovo) vypočte jako:

$$o_K = \sum_{k \in K} in(k, \mathrm{TITLE}) + in(k, \mathrm{HEADINGS}) + verb(k) \cdot in(k, \mathrm{TEXT}) + 2 \cdot verb(k) \cdot in(k, \mathrm{NEAR}) \qquad (3.1)$$

kde *K* je množina klíčových slov a funkce *verb* vrací hodnotu 2 pro sloveso, jinak hodnotu 1. Z neformálních experimentů se zdálo, že zvýhodnění slovesa vede k lepším výsledkům.

Jednotlivé klíčové entity se nemusejí v textu nacházet v doslovném znění. Samozřejmostí je srovnávání lemmat, díky němuž nezáleží na konkrétní flexi. Za výskyt určité klíčové entity se považuje také výskyt kteréhokoliv jeho **synonyma**. V oddíle 3.4.1 bylo popsáno využití synonym a odvozenin slov při generování vyhledávacího dotazu – stejná slova se tedy uplatňují i v této fázi.

V omezené míře se program snaží řešit také **koreferenci** entit. Pokud je víceslovná entita nalezena v názvu článku nebo v nadpise, považuje se v daném kontextu již za „známou“ a v samotném textu postačuje výskyt jejího hlavního substantiva. Například:

Karel IV. – Císař
*Na Hod boží velikonoční 5. dubna 1355 byl **Karel** (spolu s Annou Svídnickou) v Římě korunován římským císařem. Jeho oficiální titul byl latinský a zněl takto: Karolus Quartus divina favente clemencia Romanorum imperator semper augustus et Boemie rex.*

Kromě samotných klíčových entit, hodnotí program také výskyt **bigramů**. Je-li *B* množina všech bigramů vzniklých ze sousedících klíčových entit (první a druhá, druhá a třetí apod.), pak se hodnocení výskytu bigramů vypočítá:

$$o_B = \sum_{b \in B} 4 \cdot in(b, \mathrm{TEXT}) + 4 \cdot in(b, \mathrm{NEAR}) \qquad (3.2)$$

Entity bigramu se mohou vyskytovat v libovolném pořadí a je tolerován i případ, kdy je mezi nimi jedno slovo navíc.

Posledním hodnoceným aspektem je přítomnost předložky z otázky (viz oddíl 3.3) před entitou. Pokud se některá doporučená předložka nachází do vzdálenosti čtyř slov nalevo od kandidátní entity, rovná se bonus za předložku (např. o_p) hodnotě 4. Zatímco hodnocení klíčových entit a bigramů je poměrně univerzální míra, výskyt příslušné předložky úzce souvisí s formulací otázky a snaha o jeho hodnocení působí naivně. Z experimentálních důvodů však program toto hodnocení uskutečňuje.

Celkové ohodnocení kandidátní entity využívá normalizace počtem klíčových slov a vypočte se jako:

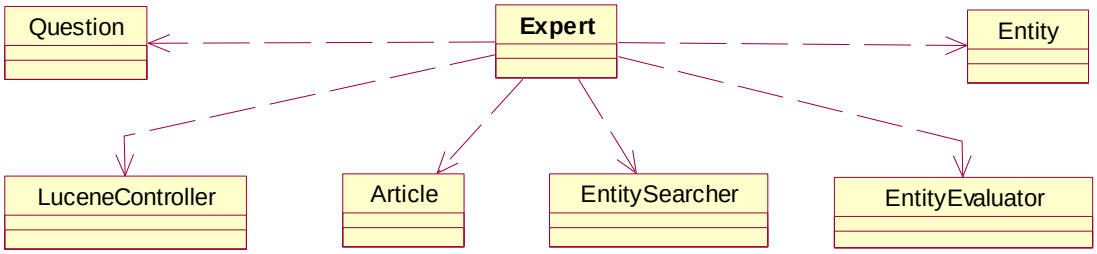
$$s(e) = round\left(10 \cdot \frac{o_K + o_B + o_p}{|K|}\right) \qquad (3.3)$$

Na jeho základě dochází k seřazení kandidátních entit a nejlépe ohodnocenou entitu program považuje za správnou odpověď. Program prezentuje uživateli i další nalezené entity nad určitý práh hodnocení (viz oddíl 3.8), jelikož se na kvalitu nejlépe hodnocené entity nelze vždy spoléhat.

3.7 Architektura systému

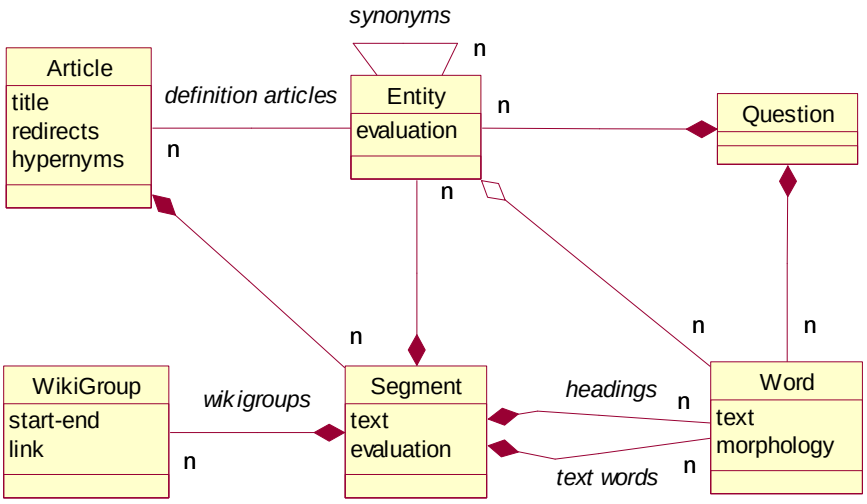
Architektura systému byla navržena objektově orientovaným stylem typickým pro jazyk Java a to s ohledem na maximální možnou modularitu systému a jeho potenciální modifikaci v budoucnu.

Zjednodušený závislostní diagram na obrázku 3.2 znázorňuje třídy, které jsou nejdůležitější z hlediska vykonávané funkce. Jádro systému tvoří třída **`Expert`**, která definuje základní rámec procesu odpovídání a volá jednotlivé moduly vykonávající specifické funkce. Třída **`LuceneController`** slouží jako rozhraní pro přístup k vyhledávači Lucene a umožňuje získat z Wikipedie množinu článků na základě výskytu slov. Každý článek je zapouzdřen třídou **`Article`**, která dělí text na segmenty a zprostředkovává údaje o článku. Třída **`EntitySearcher`** nese odpovědnost za extrakci kandidátních entit z textových segmentů a rovněž za hledání pojmenovaných entit v otázce. Klíčovou fázi ohodnocení kandidátních entit pak zajišťuje třída **`EntityEvaluator`**.



Obrázek 3.2 Hlavní funkční třídy systému.

Způsob strukturování textu je patrný z diagramu na obrázku 3.3. Článek je rozdělen na jednotlivé segmenty, které pojímají jednak příslušný text a jeho slova a jednak veškeré nadpisy, které mají k segmentu vztah (viz oddíl 3.4.2). Kromě toho si segment ponechává informace o původním rozmístění wikiodkazů a kurzív (**`WikiGroup)`**. Každé dílčí slovo rozpoznané v textu je reprezentováno třídou **`Word`**, která má na starosti velké množství funkcí poskytujících údaje o formátu slova a jeho morfologii. Třída **`Entity`** pak umožňuje seskupit několik sousedních slov, což má význam jednak při detekci víceslovných pojmenovaných entit v otázce a jednak při extrakci odpovědních entit ze segmentu.



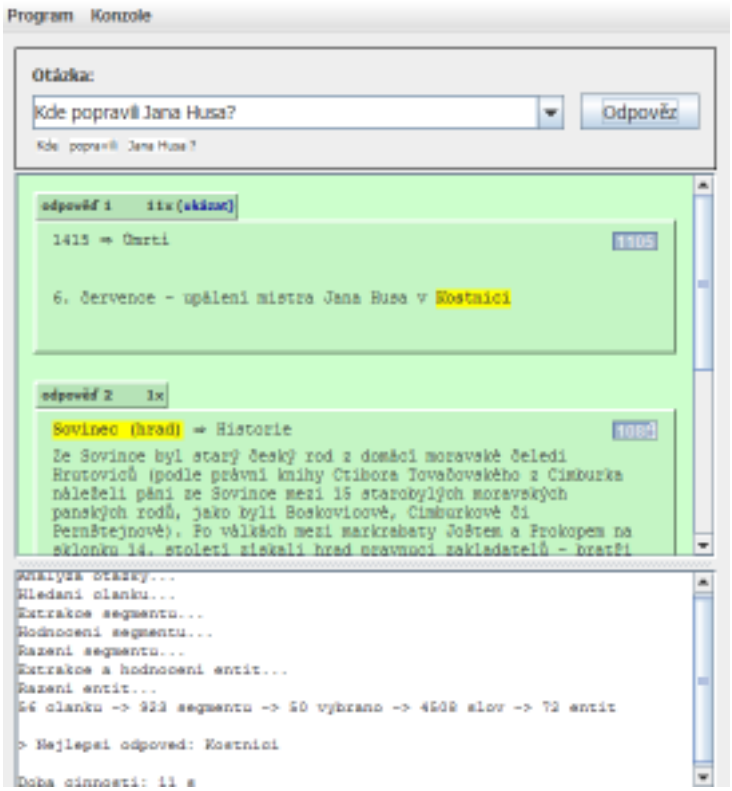
Obrázek 3.3 Hlavní datové třídy systému.

Existuje dále celá řada tříd, které vykonávají pomocnou funkci. Například třídy **`Dictionaries`** a **`Settings`** zprostředkovávají přístup k lingvistickým slovníkům a k nastavení specifického chování systému – zahrnuje vzory pro extrakci odpovědi, mapování fokusů apod. Třída **`TextProcessor`** řeší nejrůznější aspekty wikitextu, **`PhraseExtractor`** se snaží zastat úlohu syntaktického analyzátoru při extrakci fokusu otázky, **`SegmentScanner`** slouží čistě pro počítání výskytu slov v segmentu apod. Detailní popis všech tříd a jejich metod je možné nalézt v příslušné API dokumentaci, která je součástí přílohy této práce.

3.8 Uživatelské rozhraní

Pomocí knihovny **Java Swing** bylo vytvořeno jednoduché uživatelské rozhraní, jehož smyslem je přehledně prezentovat uživateli nalezené odpovědi (viz obrázek 3.4).

Systém vychází z předpokladu (vyřčeného například také v [10]), že je vhodnější odpovědi prezentovat i s kontextem a poskytnout tak uživateli možnost odpověď si sám ověřit. Výstup se tak de facto podobá výstupu klasického vyhledávacího stroje, ovšem s tím rozdílem, že jsou zřetelně vyznačeny odpovědní entity a jsou seskupeny relevantní pasáže podporující tutéž entitu. Kliknutím na některou pasáž je možné přímo prohlížet článek Wikipedie, v němž byla pasáž nalezena.



Obrázek 3.4 Hlavní okno uživatelského rozhraní.

V zápatí hlavního okna programu se navíc nachází textová konzole podávající zprávu o průběhu hledání. Systém umožňuje nastavit tři úrovně podrobnosti vypisovaných informací.

4 Hodnocení systému

V této kapitole bude popsáno vyhodnocení úspěšnosti Znalce encyklopedie na testovací sadě otázek. Cílem vyhodnocení je získat povědomí o schopnostech systému a především odhalit hlavní funkční nedostatky.

4.1 Testovací sada

Ideální testovací sada otázek pro ohodnocení Znalce encyklopedie by měla vyhovovat následujícím požadavkům:

1. **Rozsah** – Sada by měla být dostatečně veliká, aby získané výsledky byly statisticky průkazné.
2. **Rozložení otázek** – Podíl zastoupení jednotlivých typů otázek by měl odpovídat rozložení otázek od reálného uživatele.
3. **Schopnost odpovědět** – Otázky musí být zodpověditelné pomocí české Wikipedie.
4. **Nezávislost** – Konstrukce testovací sady nesmí být ovlivněna hodnocenými schopnostmi systému.

S ohledem na žádanou nezávislost bylo třeba vyhnout se tvoření testovací sady a použít již hotovou sadu. Na internetu je možné nalézt obsáhlé testovací sady anglických otázek (např. korpus systému AnswerBus[1]), jejich převod do češtiny by však byl přinejmenším problematický. Kromě jazykových aspektů by bylo třeba se potýkat s faktem, že mnohé z těchto otázek nemají encyklopedický charakter, nebo úzce souvisí s reáliemi anglofonních národů. Naopak témata, která zajímají potenciálního českého uživatele, by nemusela být vůbec zastoupena. Z těchto důvodů byla snaha nalézt sadu otázek v českém jazyce.

Problémovým se však ukázal požadavek na rozložení otázek. Potřeby uživatelů se liší podle charakteru vyhledávacího systému, takže opravdu správné rozložení otázek by bylo možné získat jen z obdobného systému. Takový však v českém prostředí není možné nalézt, takže bylo třeba od požadavku do určité míry ustoupit.

Sada otázek byla nakonec extrahována z vědomostní hry **Chcete být milionářem? 1.5 beta**[2]. Hra obsahuje celkem 2166 otázek znalostního charakteru. V drtivé většině jde o faktografické otázky, jejichž odpověď je možné vyjádřit krátkou textovou entitou, což koresponduje se zaměřením Znalce encyklopedie. Vzhledem k tomu, že hra nabízí uživateli na výběr čtyři odpovědi, bylo třeba některé otázky přeformulovat z doplňovací podoby (např. *„Hlavní město Číny je:“*) na klasickou tázací větu (např. *„Jaké je hlavní město Číny?“*). Byl však kladen důraz na to, aby při reformulaci nebyla otázka žádným jiným způsobem modifikována.

Velká část otázek musela být zcela vypuštěna, protože je nebylo možné zodpovědět pomocí české Wikipedie – buď se odchylovaly od encyklopedického charakteru (např. *„Doplňte: Mluviti stříbro…“*), nebo zkrátka nebyly kryté obsahem Wikipedie (např. *„Jak se jmenoval kůň dona Quijota?“*). Výslednou testovací sadu tvoří **1201 otázek**.

[1] http://www.answerbus.com/corpus/index.shtml

[2] http://www.dwn.cz/chcete-byt-milionarem

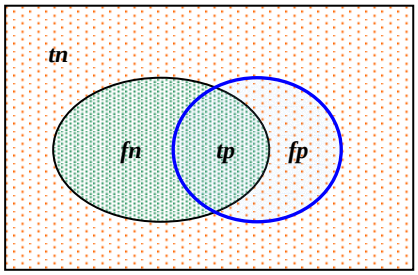
4.2 Metriky

Je třeba se rozhodnout, jakým způsobem vyjádřit úspěšnost hledání systému.

Mezi nejzákladnější metriky pro hodnocení vyhledávacích systémů patří **přesnost** a **pokrytí** (viz např. [23, str. 153]). Přesnost (precision, P) udává podíl správných odpovědí v množině prezentovaných odpovědí, pokrytí (recall, R) vyjadřuje naproti tomu podíl nalezených správných odpovědí vůči celkovému počtu správných odpovědí.

$$P = \frac{tp}{tp + fp}$$

$$R = \frac{tp}{tp + fn}$$



Obrázek 4.1 Přesnost a pokrytí vyhledávacího systému.

Na obrázku 4.1 jsou například správné odpovědi (respektive relevantní dokumenty apod.) znázorněné šrafovanou elipsou a tučná elipsa představuje množinu odpovědí prezentovaných systémem uživateli. Hodnocení přesnosti a pokrytí vychází z touhy po tom, aby se obě elipsy co nejvíce podobaly.

Je poměrně snadné docílit toho, aby měl systém buď vysokou přesnost (například zaměřením se na jednoduché případy), nebo vysoké pokrytí (prostým zobrazením všech možných odpovědí) – jde však o to uspět v obou veličinách zároveň. Jako metrika zohledňující zároveň přesnost i pokrytí se velmi často používá například **míra F** (viz [23, str. 153]):

$$F_\beta = \frac{\left(1 + \beta^2\right) \cdot P \cdot R}{\beta^2 \cdot P + R} \qquad \beta \in (0, \infty) \tag{4.1}$$

kde koeficient β vyjadřuje, nakolik je pokrytí důležitější než přesnost, respektive naopak.

Standardní vyhledávací systémy a stejně tak i Znalec encyklopedie zobrazují uživateli výsledky hledání seřazené podle kvality. Výše uvedené metriky však samy o sobě nezohledňují, jak hluboko se správná odpověď v seznamu nachází. Kromě toho nemá při hodnocení Znalce encyklopedie příliš význam používat údaj o pokrytí, jelikož testovací sada obsahuje skoro výlučně otázky, na něž existuje víceméně jedna odpověď – pokrytí se tedy bude rovnat vždy hodnotě 0 nebo 1. Pokud bychom hodnotili seznamové otázky (viz str. 6), bylo by navíc poměrně náročné určit, jaké všechny odpovědi potenciálně přicházejí v úvahu (zejména pak s ohledem na jejich dohledatelnost na Wikipedii).

Při hodnocení systému tedy využijeme jednoduchou metriku zvanou **průměrné převrácené pořadí** (mean reciprocal rank, MRR), užívanou mj. v některých ročnících evaluačního fóra TREC (viz např. [27]) nebo CLEF [4]. MRR je založena na přesnosti, do výpočtu však zahrnuje jen prezentované odpovědi, které jsou v seznamu výše než správná odpověď. Hlouběji umístěné odpovědi zanedbává, což koresponduje s chováním uživatele při procházení výsledků hledání – jakmile dojde na správnou odpověď, dalšími výsledky se již nenechá rozptylovat. Takováto přesnost se vypočte jedno-

duše jako převrácená hodnota pořadí správné odpovědi v seznamu výsledků a celková kvalita systému je pak vyjádřena průměrem přesností pro různé otázky:

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{Q}\frac{1}{rank_i} \tag{4.2}$$

kde Q je množina otázek a $rank_i$ je pořadí první správné odpovědi na otázku q_i v seznamu výsledků [1]. Pokud se správná odpověď ve výsledcích vůbec nenachází, pokládá se $rank_i = \infty$, čili $1/rank_i = 0$.

4.3 Celkové hodnocení

Hodnocení systému pomocí testovací sady 1201 otázek probíhalo poloautomatickým způsobem. Program načítal jednotlivé otázky z příslušného textového souboru a uskutečňoval hledání odpovědí. Získané výsledky poté procházel lidský uživatel a označoval správné odpovědi. Počet odpovědí zobrazených systémem byl omezen na **deset**, při čemž klasické odpovědní entity byly i s kontextem prezentovány v seznamu nejdříve, zbytek pak tvořily nejlépe hodnocené celistvé segmenty. Navíc bylo přijato omezení, že musí být zobrazeny alespoň dva segmenty, tudíž maximální počet odpovědních entit byl osm.

Aby bylo možné jakýmsi způsobem vyhodnotit relativní přínos systému, byl také uskutečněn testovací cyklus, v němž byly odpovědi hledány čistě pomocí vyhledávacího stroje. Program pouze extrahoval z každé otázky klíčová slova a zkonstruoval z nich vyhledávací dotaz pro Lucene. Nebyla při tom využita žádná expanze o synonyma, pouze lemmatizace slov. Systém vyhledal 50 nejlepších článků a z nich čistě podle výskytu klíčových slov vybral 10 nejlepších textových segmentů. Údaje o úspěšnosti nalezení odpovědí v takovýchto segmentech poslouží jako **referenční výsledky** pro srovnání s vlastními výsledky Znalce encyklopedie.

Výsledky hodnocení systému jsou znázorněny na obrázku 4.2. Tabulka a histogram zachycují rozložení správných odpovědí ve výsledkových seznamem jednotlivých testovacích otázek. Písmenem **H** jsou označeny referenční výsledky prostého hledání klíčových slov. U samotného odpovědního systému jsou pak rozlišeny případy, kdy jde o plnohodnotnou odpověď ve formě textové entity (**E**), a volnější případy, kdy je k dispozici textový segment (**S**) s odpovědí. Případy typu S zahrnují všechny případy typu E a dále výsledky, z nichž je nutné odpověď bez pomoci systému vyčíst. Pro zařazení do této kategorie ovšem nestačilo, aby se v segmentu nacházela entita shodující se s odpovědí – hodnotící uživatel dbal na to, aby obsah segmentu skutečně danou odpověď vyjadřoval.

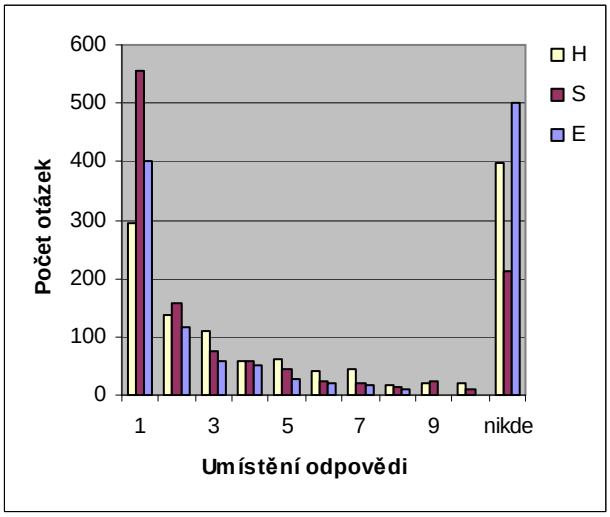
Z jednotlivých pořadí správných odpovědí byla vypočítána suma přesností vyjádřených převrácenou hodnotou pořadí – tzv. **reciprocal rank** (RR). Vydělením celkové sumy počtem otázek (viz vzorec 4.2) pak dostáváme průměrnou přesnost $\boldsymbol{MRR_e}$ **= 0,42** pro plnohodnotné odpovědi a $\boldsymbol{MRR_s}$ **= 0,58** pro relevantní segmenty. Referenční vyhledávač dosáhl přesnosti $MRR_h = 0{,}37$.

4.3.1 Diskuze o výsledcích

Relativně vysokou úspěšnost prostého hledání klíčových slov zapříčinila především všeobecná jednoduchost otázek testovací sady. Takováto úspěšnost závisí nezbytně na přibližném souladu mezi formulací otázky a formulací hledané informace v textu. Přestože Wikipedie jako zdrojový korpus nedosahuje zdaleka redundance informací webových dokumentů, ukázalo se, že se testovací otázky

obvykle ptají na natolik běžná fakta, že jejich nalezení na Wikipedii je možné často i bez generování synonym a jiných podpůrných technik.

Pořadí	Počet otázek			Přesnost (RR)		
	H	S	E	H	S	E
1	294	557	400	294,00	557,00	400,00
2	137	159	118	68,50	79,50	59,00
3	109	76	58	36,33	25,33	19,33
4	59	59	50	14,75	14,75	12,50
5	63	45	29	12,60	9,00	5,80
6	42	24	19	7,00	4,00	3,17
7	43	22	18	6,14	3,14	2,57
8	17	14	10	2,13	1,75	1,25
9	22	23	0	2,44	2,56	0,00
10	19	10	0	1,90	1,00	0,00
nikde	396	212	499	0,00	0,00	0,00
			MRR =	**0,37**	**0,58**	**0,42**



Obrázek 4.2 Statistika četnosti různého umístění správné odpovědi v seznamu výsledků.

Z celkových výsledků nicméně vyplývá, že Znalec encyklopedie dosahuje výrazně lepší přesnosti než referenční systém. Přesnost MRR_h referenčního systému má smysl srovnávat zejména s MRR_s, jelikož oba údaje vyjadřují schopnost prezentovat uživateli celé segmenty s odpovědí. Výsledky však dokládají, že i plnohodnotné odpovídání textovou entitou – MRR_e – dopadlo lépe než odpovídání obsáhlými segmenty u referenčního systému. Rozdíl mezi přesnostmi 0,37 a 0,42 není příliš velký a bylo by tedy možné pochybovat o jeho statistické významnosti. Po vzoru [46] proto využijeme **Studentův t-test**, rozšířenou metodu testování statistických hypotéz. Jelikož oba výsledky pocházejí z totožné testovací sady, přichází v úvahu zpárování přesností (resp. převrácených pořadí – *RR*) správných odpovědí u jednotlivých testovacích otázek. Pro každou otázku (s číslem *i*) vypočteme rozdíl přesnosti RR_h referenčního systému a RR_e vytvořeného systému:

$$Y_i = RR_{e,i} - RR_{h,i} \tag{4.3}$$

Z celé testovací sady tak získáme průměrný rozdíl přesností (který odpovídá rozdílu MRR):

$$\overline{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i \quad \cong 0{,}049 \tag{4.4}$$

kde $n = 1201$ je počet otázek testovací sady. Následuje výpočet rozptylu hodnot:

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(Y_i - \overline{Y}\right)^2 \quad \cong 0{,}276 \tag{4.5}$$

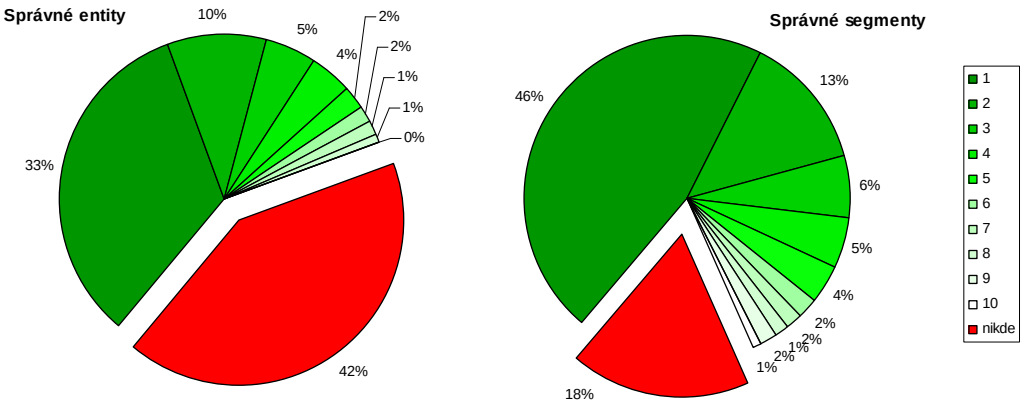
Samotný t-test, kterým se snažíme zamítnout hypotézu, že rozdíl v přesnostech je statistický nulový, se pak vypočte následovně:

$$t = \frac{\overline{Y}}{\sqrt{s^2 / n}} \cong 3{,}208 \qquad (4.6)$$

Čím vyšší je hodnota *t*, tím průkazněji je přesnost Znalce encyklopedie lepší než přesnost referenčního systému. Získaná hodnota t = 3,208 dle statistických tabulek[1] svědčí o tom, že v lepší výsledky oproti referenčnímu systému lze věřit s pravděpodobností $p > 99{,}9$ %.

Z grafu na obrázku 4.2 je patrný strmý pokles šance pro nalezení správné odpovědi s rostoucí hloubkou v seznamu výsledků. Ukazuje se jednak, že hodnocení odpovědních entit a segmentů dovede výsledky relativně dobře seřadit, a jednak, že větší počet prezentovaných odpovědí by neměl příliš vliv na kvalitu výstupu. Hluboko položené výsledky totiž málokdy představují správnou odpověď; je třeba poznamenat, že drobný nárůst správných segmentů v hloubce 9 (a nulová správnost entit tamtéž) je způsoben výše popsaným požadavkem na to, aby přinejmenším v hloubce 9 a 10 byly vždy segmentové odpovědi.

Poměry mezi užitečnostmi jednotlivých hloubek při hledání odpovědi jsou znázorněné v koláčových grafech na obrázku 4.3. Je evidentní, že velkou část otázek nedovede systém vůbec zodpovědět. Vystačíme-li si s prostými segmenty, jedná se o 18 % otázek, pokud však trváme na skutečných odpovědích, systém zcela selhává ve 42 % případů.



Obrázek 4.3 Podíl umístění správné odpovědi v otázkách testovací sady.

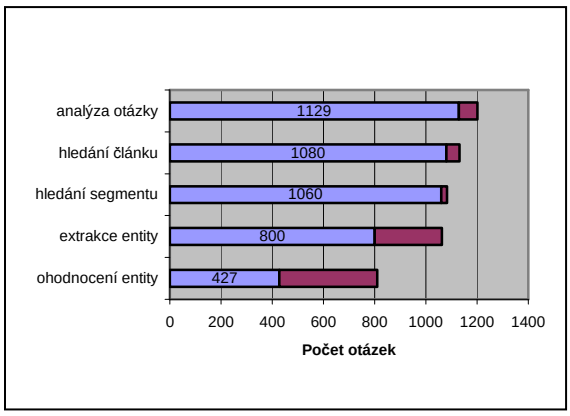
Jestliže bychom Znalce encyklopedie převedli na ryzí odpovídač, vracející právě jednu odpověď (nikoliv seznam), získali bychom správnou odpověď ve 33 % případů, zatímco u 13 % otázek by bylo třeba odpověď vyčíst z textového segmentu a zbývajících 54 % otázek by zůstalo nezodpovězeno.

4.4 Hodnocení fází

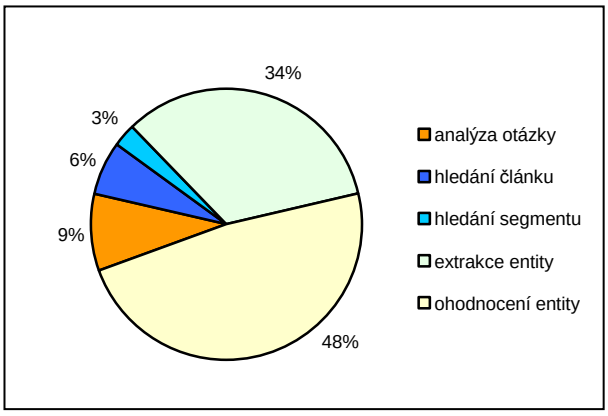
Graf na obrázku 4.4 ukazuje, kolik testovacích otázek z celkového počtu 1201 prošlo úspěšně jednotlivými fázemi činnosti systému. Za úspěch poslední fáze je považován takový stav, kdy je odpovědní entita v seznamu na prvním místě, nebo je výsledek velmi blízko tomuto ideálu. Omezíme-li se nao-

[1] http://en.wikipedia.org/wiki/Student's_t-distribution

pak na případy, které nekončí ideálním výsledkem, můžeme z obrázku 4.5 vyčíst, které fáze způsobovaly nejčastěji selhání systému (případně nedokonalost výsledků).



Obrázek 4.4 Úspěšnost jednotlivých fází činnosti.



Obrázek 4.5 Příčina selhání systému z hlediska fází.

4.4.1 Zpracování otázky

Ukazuje se, že analýza otázky selhává pouze v 6 % případů, přestože je implementována jen pomocí naivních pravidel kladoucích důraz na tázací slovo a fokus otázky (viz oddíl 3.3). Příčinou je relativní jednoduchost otázek v testovací sadě – kromě toho je třeba připomenout, že některé otázky byly přeformulovány do standardní tázací podoby a příležitostně byly také opravovány pravopisné chyby. Testovací sada je tak tvořena sice různě formulovanými otázkami, ale výlučně jde o formálně dokonalé otázky, které je v praxi ztěží možné od uživatele vyžadovat.

Selhání analýzy otázky spočívalo především v chybně rozpoznaném fokusu. Vyšlo najevo, že si program nedovede poradit s otázkami typu *„Jak se jmenuje jedna z prvních jarních květin?“*, kde rozpoznává jako fokus nesprávně výraz *jedna*. Potíže způsobovala také řada jiných nestandardně formulovaných otázek – například u otázky *„Jaký název mělo první nadzvukové dopravní letadlo?“* by se slušelo, aby fokusem bylo *letadlo*, nikoliv *název*.

Problémem jsou rovněž otázky, které žádají po systému kategorizaci nějaké entity, například: *„Jaký typ vesmírného tělesa je Slunce?“*. Fokus *typ* sám o sobě většinou neumožňuje správnou extrakci odpovědi, je třeba očekávaný typ odpovědi identifikovat korektněji.

V několika případech byla v otázce využita nezvyklá tázací slova (např. *kolikaranný*, *kolikaaktová*). Program také nezohledňuje případy, kdy je tázací slovo *co* umístěno za jiným slovním druhem než předložkou – například *„Výrobou čeho proslula Sušice?“*.

Velká část selhání byla zapříčiněna extrakcí nezbytných entit z otázky. Při návrhu mechanismů pro zpracování otázky byl kladen důraz na reálné otázky, vyjádřené co nejkompaktněji. V testovací sadě se však vyskytují i otázky, které obsahují redundantní pojmy. Například otázku *„Za kterou zemi hrají fotbalové kluby Rapid a Austria?“* by bylo možné snadno zodpovědět, avšak pojmenované entity *Rapid* a *Austria* jsou při analýze označeny za nezbytné, takže hledaná odpověď (*Rakousko*) musí mít ve svém kontextu oba pojmy, což je nepravděpodobné.

Zhruba 30 otázek testovací sady vyjadřuje požadavek na alternativní název určitého pojmu – například *„Jak se říká Prvnímu svátku vánočnímu?“* nebo *„Jak se jmenoval vlastním jménem papež Jan Pavel II.?“*. Systém bohužel není schopen tyto druhy otázek rozpoznat, přestože extrakce takových odpovědí z Wikipedie je možná. Paradoxně jsou potenciální odpovědi explicitně vyloučeny

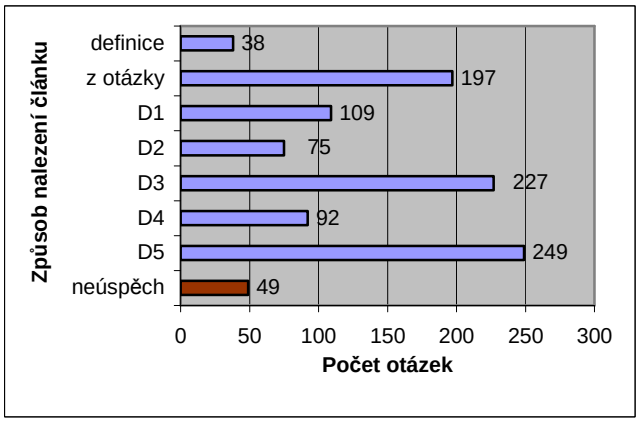
z výsledků, protože se systém vyhýbá pojmům uvedeným v otázce – a například *Karol Wojtyla* je chápán jako synonymum k *Jan Pavel II.*

4.4.2 Hledání článků a segmentů

Z grafu na obrázku 4.4 je evidentní, že proces hledání relevantních článků Wikipedie a následně hledání textových segmentů bývá neúspěšný jen v malém množství případů. Maximální počet 50 prozkoumávaných článků i segmentů je zřejmě v kontextu Wikipedie dostatečná hodnota pro nalezení požadované informace.

V oddílu 3.4 bylo popsáno, že systém užívá v podstatě sedmi různých metod pro získání relevantních článků: U definičních otázek jednoduše vyhledává článek, jehož název odpovídá definovanému pojmu, u ostatních typů otázek nejprve podobně extrahuje články pro některé pojmy z otázky a poté přichází na řadu až pět forem vyhledávacího dotazu pro systém Lucene (označme D1–D5), aplikovaných od nejpřísnějšího po nejvolnější.

Užitečnost jednotlivých přístupů při řešení testovací sady otázek je možné posoudit z grafu na obrázku 4.6. Histogram zachycuje pouze otázky, u nichž byly nakonec nalezeny správné odpovědi, a omezuje se čistě na mateřské články těchto odpovědí. Je především pozoruhodné, že celých 197 nedefiničních otázek (cca 20 %) je nakonec nejlépe zodpovězeno pomocí článku přímo uvedeného v otázce. Vypovídá to hlavně o jednoduchosti otázek testovací sady. Pokud jde o vyhledávací dotazy, výsledky hodnocení naznačují, že dotazy D1 a D2, vyžadující přítomnost fokusu v hyperonymních kategoriích článku, fungují relativně dobře. Dominantní úlohu sice mají dotazy D3–D5, je ale třeba si uvědomit, že dotazy D1 a D2 nejsou tak univerzální – mohou uspět jen tam, kde existuje pro odpověď přímo článek na Wikipedii a tento článek musí navíc mít korespondující hyperonymní kategorie. Kromě toho je počet vrácených článků u dotazů D1 a D2 omezen na 5+5, zatímco ostatní dotazy mohou úhrnně dosáhnout až celkového limitu 50 článků.



Obrázek 4.6 Úspěch jednotlivých metod pro nalezení článku.

Příčinou neúspěchu hledání článků nebo následně segmentů bývá obvykle selhání expanze slov o synonyma a odvozeniny. Znalec encyklopedie je bohužel v této souvislosti limitován kvalitou užitých lingvistických zdrojů. Nalezení článku selhalo například u otázky *„Jak se nazývají zanícení přívrženci knih?“*. Kontext hledané odpovědi je *„Osoba se zaujetím pro knihy je označována jako bibliofil nebo knihomil.“* – slovník synonym ale bohužel neuvádí *zaujetí* jako významově blízké slovo k *zanícení*. Podobných příkladů by se jistě dala poskytnout celá řada.

Některé otázky mohou být založeny na příliš nespecifických slovech, což může mít za následek množství nalezených článků výrazně přesahující zvolený limit. V testovací sadě to byla například otázka *„Jaký stát má nejmenší rozlohu?“*. Bylo vyzkoušeno, že ani dotaz omezující se na články států a obsahující čistě neexpandovaná slova *nejmenší* a *stát* by nebyl schopen nalézt článek *Vatikán* výše než na 100. pozici ve výsledkovém seznamu. Případy tohoto druhu je tedy nutné ošetřit již ve fázi analýzy otázky a odpověď hledat chytřeji.

4.4.3 Extrakce kandidátních entit

Dle očekávání je naprostá většina selhání systému (82 %) způsobena klíčovými fázemi činnosti – nefunkční extrakcí kandidátních entit nebo jejich nesprávným ohodnocením.

Z hodnocení testovací sadou vyplývá, že pokud systém úspěšně zpracuje otázku a získá relevantní segmenty, pak je zhruba **76%** šance, že se mu podaří extrahovat příslušnou odpovědní entitu. Na extrakční fázi má zásadní vliv očekávaný typ odpovědi, vyjádřený fokusem otázky. V testovací sadě se vyskytlo odhadem 400 různých fokusů otázky – nejčastější z nich je možné vidět v tabulce 4.1. Tyto výrazy byly použity v celých 40 % otázek, není se tedy čemu divit, že existují i odpovědní systémy, které si vystačí s malým počtem rozeznávaných typů (viz oddíl 2.3.1).

člověk	*stát*	*barva*	*jednotka*	*jazyk*	*prvek*
místo	*země*	*řeka*	*čas*	*národnost*	*zvíře*
rok	*autor*	*délka*	*zkratka*	*skupina*	*král*
město	*jméno*	*hora*	*značka*	*ostrov*	*sport*

Tabulka 4.1 Výběr 24 nejčastějších fokusů otázky.

Jak bylo vysvětleno v oddíle 3.5.2, program se při extrakci odpovědí specializuje na krátké textové výrazy, většinou pojmenované entity. K jejich rozeznání v textu užívá v případě časových údajů a různých číselných veličin soustavu slovních vzorů a regulárních výrazů, v ostatních případech těží ze struktury wikitextu a z údajů o slovních druzích. Už z podstaty tedy není schopen zodpovědět nefaktografické otázky ani žádným způsobem odpověď generovat agregací více zdrojů. Pokud se tedy v testovací sadě vyskytla komplexnější otázka, selhaly schopnosti systému právě ve fázi extrakce.

Největším zdrojem problémů extrakční fáze je neschopnost ověřit **sémantický typ entity**. Segment s odpovědí může být nalezen a mohou být rozpoznány hranice textové entity, která představuje odpověď, pokud však není zjištěna shoda s očekávaným typem odpovědi, systém entitu zahodí a pokračuje v hledání.

Ověření typu entity z velké části spoléhá na Wikipedii coby ontologii – hlava fokusu nebo její synonymum se musí nacházet v hyperonymní nadkategorii definičního článku entity. Spolehlivost Wikipedie však v těchto směrech značně kolísá. Například při hledání odpovědi na otázku *„Který orgán produkuje hormon sekretin?“* nemohl systém přijmout pojem *dvanáctník*, protože se stejnojmenný článek nenachází v kategorii *Orgány*. Potíže mohou nastat také v důsledku neznalosti synonym – při hledání *„týmu NHL“* například systém nerozpoznal kategorii *Sportovní kluby*. Byly zjištěny také nedostatky v indexaci kategorií – systém neumí rozpoznat například žádný druh hmyzu, protože *hmyz* není formálně množné číslo a nebyl ani ručně zařazen mezi hyperonymní kategorie.

Řada fokusů otázek ani prakticky nemůže být kryta kategoriemi Wikipedie, protože se jedná o různé mnohovýznamové, abstraktní a relativní pojmy, například: *„Kterou vadou trpěl v pozdějším*

věku malíř Francisco Goya?“, *„Co je symbolem svatého Marka?“*, *„V jakém prostředí žije pižmoň?“*, *„Jak se jmenuje bratr Lisy Simpsonové?“*. Do určité míry v takovém případě pomáhá sekundární ověření typu pomocí kontextu – pokud se v bezprostředním okolí entity (do tří slov) nachází fokus otázky, pak je entita přijata. V testovací sadě pomohl tento způsob ověřování cca u 40 otázek, zejména u relativních pojmů (bratr/manžel/symbol…). U otázek typu *„Kdo je autorem/tvůrcem/hrdinou…?“* (rovněž cca 40) pomohlo systému nespoléhat jen na explicitní fokus, ale na základě tázacího slova *Kdo* extrahovat se sníženým hodnocením jména všech osob.

Pro zkvalitnění ověřování typu entity by bylo vhodné hyperonymní vztahy extrahovat nejen z kategorií, ale třeba i přímo z textu. Systém by mohl při indexaci projít textový korpus a pomocí předpřipravených vzorů najít příslušné vazby. Velice užitečná bývá v tomto smyslu už první věta každého článku Wikipedie, která většinou začíná tvarem „`<POJEM> je <NADPOJEM>`“. Kromě Wikipedie je však třeba zapojit také klasické lingvistické zdroje, především lexikon Wordnet [42], který ovšem zatím nedosahuje kvalit anglické verze.

Zásadním nedostatkem aktuální implementace systému je neschopnost extrahovat odpovědi obecného typu *věc*. Program si umí poradit s otázkou typu *„Co bylo oblíbenou zbraní maďarských bojovníků?“*, v níž je explicitně uveden fokus, nezvládá však například otázku *„Co bylo oblíbené u maďarských bojovníků?“*. Očekávaný typ odpovědi je příliš široký, takže by se dalo za kandidátní entitu považovat prakticky cokoliv. Otázky tohoto druhu (v testovací sadě 42) při extrakci bez výjimky selhávají. Možným řešením by bylo uskutečnit odhad očekávaného typu statistickou klasifikací otázky (viz oddíl 2.3.1), nebo detekovat syntaktickou podobnost v textu – například *„U maďarských bojovníků … byly populární … šavle.“*

Pokud jde o extrakci entit pomocí vzorů, k jejich využití došlo u 138 otázek a u 89 % z nich byla extrakce úspěšná. Ukázalo se však, že by bylo dobré sestavit vzory ještě pro několik málo speciálních entit – například pro chemické značky, písmena nebo příjmení.

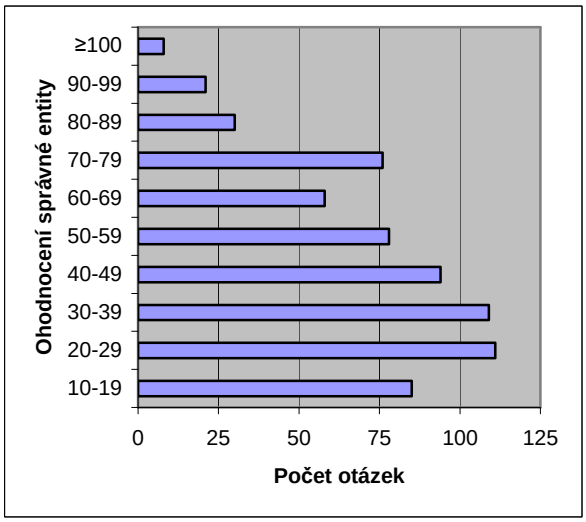
4.4.4 Ohodnocení entit

K největšímu selhání systému dochází ve fázi ohodnocení kandidátních entit. Je však třeba říci, že za toto selhání se považují i případy, kdy byla správná odpověď uživateli prezentována, ale až na druhém či hlubším místě. Z testovací sady vyplynulo, že pokud je správná entita extrahována z textu, pak cca v **50 %** případů ji hodnocení umístí na první pozici v seznamu výsledků, v 38 % se nachází na jiné prezentované pozici (2.–8.), v ostatních případech není pro své nízké hodnocení vůbec uživateli zobrazena.

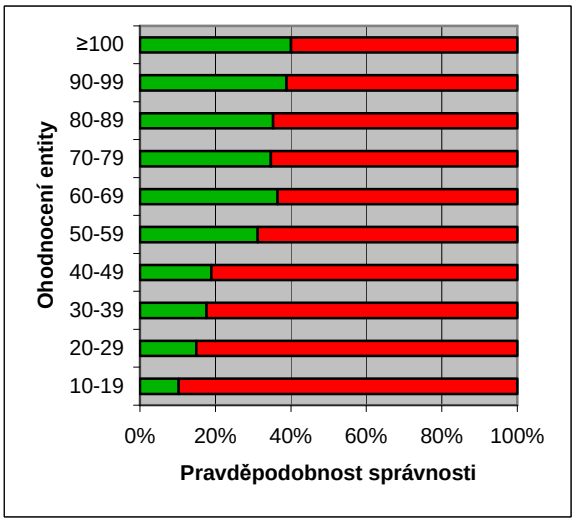
Ukazuje se, že ohodnocení, závislé v drtivé míře na výskytu klíčových slov (viz oddíl 3.6), se svou úspěšností velmi kolísá a nemá zřejmě smysl, aby se systém snažil nějakým způsobem na konkrétní hodnoty spoléhat. Obrázek 4.7 zachycuje rozložení ohodnocení u správných entit v testovací sadě. Je evidentní, že velká část otázek byla zodpovězena entitou s dosti malým ohodnocením, mnohdy na hranici prahu o velikosti 10. Relativně oproti ostatním nalezeným entitám bylo ohodnocení totiž dostačující.

Graf na obrázku 4.8 znázorňuje teoretickou **důvěru ve správnost** entity podle konkrétního ohodnocení. Vyplývá z něj, že zvyšující se hodnocení v absolutním pojetí roste s kvalitou entity jen do hodnoty 50, jakékoliv vyšší hodnoty poskytují víceméně stejnou míru důvěry – cca 35 %. Ohodnocení entit tedy není zdaleka ideální a může jako míra kvality odpovědi sloužit jen omezeně. Zamysleme se ale nad vlivem prahového ohodnocení pro kvalitu výstupu. Zkombinováním údajů z obou

grafů je možné zjistit, že průměrná míra důvěry v odpovědní entity s ohodnocením $e \geq 10$ je cca 19 %. Z hlediska uživatele je žádoucí, aby míra důvěry byla co nejvyšší, nabízí se tedy zvýšit vhodným způsobem práh a vyfiltrovat tak málo pravděpodobné odpovědi. Tímto však na druhou stranu nezbytně dojde ke snížení počtu zodpovězených otázek (a tedy i MRR). Jak by to prakticky mohlo vypadat znázorňuje obrázek 4.9, modelující vliv pěti různých prahů ohodnocení na výsledky systému.

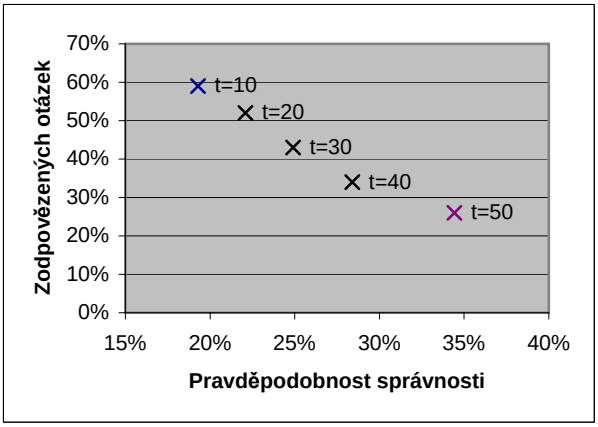


Obrázek 4.7 Výskyt ohodnocení správných entit.



Obrázek 4.8 Důvěra ve správnost entity podle ohodnocení.

Za příčinu nedokonalosti hodnocení entit lze považovat jednoduchost zpracování této fáze. Přestože bylo hodnocení výskytu klíčových slov řešeno poměrně důkladně (viz oddíl 3.6), stále jde přecejen o velmi povrchový přístup k textu. Je vůbec otázkou, nakolik lze čistě na lexikální a morfologické úrovni dosáhnout zlepšení. Bylo by určitě možné lépe zpracovat významnost jednotlivých klíčových slov v otázce a přiřadit jim váhy například na základě strojového učení. V úvahu přichází také optimalizace ohodnocení podpůrným hledáním – v oddíle 2.3.4 již bylo uvedeno, že někteří autoři validují odpovědi analýzou kookurence slov otázky a odpovědi na webu. Právě využití více zdrojů při hledání odpovědi nabízí velký prostor pro zlepšení.



Obrázek 4.9 Vliv prahu *t* na správnost entit a pokrytí otázek.

Při detekci klíčových slov má podobně jako v jiných fázích důležitý vliv slovník synonym a další lingvistické zdroje. Nemožnost rozpoznat alternativní znění klíčového slova v kontextu entity

vedlo ke snížení ohodnocení u řady otázek. Stejně tak ale systém selhává v opačném případě, kdy velké množství dostupných synonym zkresluje výsledky a nesprávné entity v seznamu odpovědí předběhnou korektní odpověď. Ukázalo se také, že se slova mnohdy sice nacházejí poblíž entity, ale syntaxe dokazuje, že k entitě nemají vztah. Je tedy klíčové jednak mít co nejkvalitnější lingvistické zdroje pro určení sémantické podobnosti slov a jednak zapojit syntaktický analyzátor, který systému výrazně rozšíří obzory.

5 Závěr

V rámci této práce byl navržen systém, který dovede s větším či menším úspěchem zodpovídat otázky formulované v přirozeném jazyce. Jako zdroj informací posloužila česká jazyková mutace internetové encyklopedie Wikipedie. Systém se zaměřuje na faktografické otázky, u nichž je odpověď vyjádřena většinou krátkou textovou entitou.

Za hlavní přínos práce lze považovat už snahu o využití české Wikipedie jako textového korpusu, která je v této oblasti netradiční. Projekt nastínil způsob využití některých strukturálních prvků Wikipedie pro lokalizaci odpovědních entit a navrhl také nahlížet na Wikipedii jako na slovník pojmů. Rozebrány byly některé aspekty související se zpracováním české otázky a její klasifikací podle očekávaného typu odpovědi. Práce také popsala způsob generování co nejefektivnějšího vyhledávacího dotazu a metriky pro hodnocení výskytu klíčových slov v kontextu nalezené entity.

Výsledný program dosáhl na sadě jednoduchých znalostních otázek přesnosti $MRR_e = 0{,}42$ při maximu stanoveném na osm prezentovaných entit. Tím lehce předstihl klasický vyhledávací stroj, u něhož navíc uživatel musí odpovědi vyčíst z velkých textových pasáží. Vzhledem k tomu, že program prezentuje odpovědní entity i s kontextem a volný prostor do maximálního počtu deseti výsledků vyplňuje textovými pasážemi, bylo možné uskutečnit také hodnocení programu coby vyhledávače pasáží. Zjištěná přesnost $MRR_s = 0{,}58$ je více než o 50 % lepší než přesnost referenčního vyhledávacího stroje.

Vzhledem k jednoduchosti testovacích otázek je však přesnost odpovědního systému nedostačující pro praktické využití. I při omezení se na pokryté druhy otázek systém nedovede v řadě případů extrahovat entitu žádaného typu a rovněž následné hodnocení entit selhává velmi často a z nejrůznějších důvodů.

Na druhou stranu je třeba říci, že systém není náročný na lingvistické zdroje – vystačuje si s nejednoznačnými morfologickými údaji, synonymy slov a samotnou Wikipedií. Dokazuje, že i při velmi povrchovém zpracování textu je možné sestrojit relativně obstojný QA systém.

Literatura

[1] Wikipedia, The Free Encyclopedia [online]. 2008. [cit. 2008-11-03] URL: <http://en.wikipedia.org>

[2] OGILVIE, P. Retrieval Using Structure for Question Answering. In *Proceedings of the First Twente Data Management Workshop (TDM 04)*. 2004.

[3] DANG, H. T.; KELLY, D.; LIN, J. Overview of the TREC 2007 Question Answering Track. In *Proceedings of the Sixteenth Text REtrieval Conference (TREC 2007)*. Gaithersburg, USA: NIST, 2007.

[4] FORNER, P., et al. *Overview of the CLEF 2008 Multilingual Question Answering Track.* Aarhus, Dánsko: 2008.

[5] FUKUMOTO, J., et al. An Overview of the 4th Question Answering Challenge (QAC-4) at NTCIR Workshop 6. In *Proceedings of the Sixth NTCIR Workshop Meeting*. Tokyo, Japan: National Institute of Informatics, 2007. s. 433-440. ISBN 978-4-86049-039-3.

[6] SASAKI, Y., et al. Overview of the NTCIR-6 Cross-Lingual Question Answering (CLQA) Task. In *Proceedings of the Sixth NTCIR Workshop Meeting*. Tokyo, Japan: National Institute of Informatics, 2007. s. 153-163. ISBN 978-4-86049-039-3.

[7] ROUSSINOV, D.; FAN, W.; FLORES, J. R. Beyond Keywords: Automated Question Answering on the Web. *Communications of the ACM*, 2008, vol. 51, no. 9, s. 60-65. ISSN 0001-0782.

[8] ZHENG, Z. AnswerBus Question Answering System. In *Proceedings of Human Language Technology Conference (HLT 2002).* San Diego, USA: 2002.

[9] KATZ, B., BORCHARDT, G. C., FELSHIN, S. Natural Language Annotations for Question Answering. In *Proceedings of the Nineteenth International Florida Artificial Intelligence Research Society Conference*. USA: AAAI Press, 2006. s. 303-306.

[10] KAISSER, M., BECKER, T. Question Answering by Searching Large Corpora with Linguistic Methods. In *Proceedings of the 13th Text Retrieval Conference (TREC 2004)*. Gaithersburg: 2004.

[11] KAISSER, M. The QuALiM Question Answering Demo: Supplementing Answers with Paragraphs drawn from Wikipedia. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, Demonstrations Session, ACL 2008*. Columbus, USA: 2008.

[12] MOLDOVAN, D., et al. Lasso: A tool for surfing the answer net. In *Proceedings of the Eighth Text Retrieval Conference (TREC-8)*. USA: NIST, 1999.

[13] FERRET, O., et al. Finding An Answer Based on the Recognition of the Question Focus. In *Proceedings of the Tenth Text Retrieval Conference (TREC 2001)*. USA: NIST, 2001.

[14] SUNDBLAD, H. A Re-examination of Question Classification. In *Proceedings of the 16th Nordic Conference of Computational Linguistics NODALIDA-2007*. Tartu: University of Tartu, 2007. s. 394-397.

[15] LI, X., ROTH, D. Learning question classifiers. In *Proceedings of the 19th international conference on Computational linguistics (COLING'02)*. Taipei: 2002.

[16] SUZUKI, J., et al. Question Classification using HDAG Kernel. In *Proceedings of the ACL 2003 workshop on Multilingual summarization and question answering*. Sapporo, Japonsko: ACL, 2003, vol. 12. s. 61-68.

[17] HICKL, A., et al. Question Answering with LCC's Chaucer-2 at TREC 2007. In *Proceedings of the 2007 Text Retrieval Conference (TREC 2007)*. Gaithersburg, USA: 2007.

[18] FELLBAUM, C. *Wordnet: An Electronic Lexical Database*. MIT Press, 1998. ISBN 978-0262061971.

[19] PRAGER, J., CHU-CARROLL, J., CZUBA, K. Statistical Answer-Type Identification in Open-Domain Question Answering. In *Proceedings of Human Language Technology Conference (HLT 2002)*. San Diego, USA: 2002.

[20] HARABAGIU, S., et al. Employing Two Question Answering Systems in TREC-2005. In *Proceedings of the 2005 Text Retrieval Conference (TREC 2005).* Gaithersburg, USA: 2005.

[21] MOLDOVAN, D., BOWDEN, M., TATU, M. A Temporally-Enhanced PowerAnswer in TREC 2006. In *Proceedings of the fifteenth text retrieval conference.* Gaithersburg, USA: 2006.

[22] IGNATOVA, K., BERNHARD, D., GUREVYCH, I. Generating High Quality Questions from Low Quality Questions. In *Workshop on the Question Generation Shared Task and Evaluation Challenge*. 2008.

[23] MANNING, C. D., RAGHAVAN, P., SCHÜTZE, H. *An Introduction to Information Retrieval*. New York, USA: Cambridge University Press, 2008. ISBN 978-0-521-86571-5.

[24] AGICHTEIN, E., LAWRENCE, S., GRAVANO, L. Learning to Find Answers to Questions on the Web. *ACM Transactions on Internet Technology (TOIT)*, 2004, vol. 4, issue 2, s. 129-162. ISSN 1533-5399.

[25] MONZ, C. *From Document Retrieval to Question Answering* (dizertační práce). Amsterdam: Universiteit Van Amsterdam, 2003. ISBN 90-5776-116-5.

[26] HARABAGIU, A., et al. Falcon: Boosting Knowledge for Answer Engines. In *Proceedings of the 9th Text Retrieval Conference (TREC-9)*. Gaithersburg: NIST, 2000. s. 479-488.

[27] HOVY, E., et al. Question Answering in Webclopedia. In *Proceedings of the Ninth Text Retrieval Conference (TREC-9).* Gaithersburg: NIST, 2000. s. 655-664.

[28] YANG, H., et al. Structured Use of External Knowledge for Event-based Open Domain Question Answering. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in IR.* Toronto, Canada: ACM Press, 2003. s. 33-40.

[29] PRAGER, J., et al. Question Answering by Predictive Annotation. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Reasearch and Development in IR.* Athény, Řecko: 2000. s. 184-191.

[30] PINCHAK, C., BERGSMA, S. Automatic Answer Typing for How-Questions. In *Procedings of the joint Human Language Technologies and the Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2007).* New York: 2007. s. 516-523.

[31] ŠEVČÍKOVÁ, M., ŽABOKRTSKÝ, Z., KRŮZA, O. *Zpracování pojmenovaných entit v českých textech*. Technická zpráva ÚFAL/CKL TR-2007-36. Praha: ÚFAL MFF UK, 2007. ISSN 1214-5521.

[32] HAN, K. S., et al. Korea University Question Answering System at TREC 2004. In *Proceedings of The 13th Text Retrieval Conference (TREC 2004).* Gaithersburg, USA: NIST, 2004.

[33] VERBERNE, S., et al. Discourse-based answering of why-questions. *Traitement Automatique des Langues 2006*, 2007, vol. 47, no. 2, s. 21-41.

[34] MOLDOVAN, D., CLARK, C., HARABAGIU, S. COGEX: A Logic Prover for Question Answering.

In *Proceedings of the HLT-NAACL 2003*. Edmonton, Canada: 2003. s. 87-93.

[35] LEE, C. W., LEE, Y. H., HSU W. L. Exploring Shallow Answer Ranking Features in Cross-Lingual and Monolingual Factoid Question Answering. *International Journal of Computational Linguistics & Chinese Language Processing*, 2008, vol. 13, no. 1.

[36] ROUSSINOV, D., ROBLES, J. Self-Learning Web Question Answering System. In *Proceedings of the 13th international WWW conference on Alternate track papers & posters*. New York, USA: 2004.

[37] MAGNINI, B., et al. Is It the Right Answer? Exploiting Web Redundancy for Answer Validation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, USA: 2002. s. 425-432.

[38] DACIUK, J. *Finite state utilities* [online]. [cit. 2008-12-28]. URL: <http://www.eti.pg.gda.pl/katedry/kiw/pracownicy/Jan.Daciuk/personal/fsa.html>

[39] WEISS, D. *Finite State Automaton in Java* [online]. 2004. [cit. 2008-12-28]. URL: <http://www.cs.put.poznan.pl/dweiss/xml/projects/fsa/index.xml?lang=en>

[40] SEDLÁČEK, R., SMRŽ, P. A New Czech Morphological Analyser ajka. In *Text, Speech and Dialogue, 4th International Conference, TSD 2001*. Berlin: Springer-Verlag, 2001. s. 100-107. ISBN 3-540-42557-8.

[41] Dictionaries - Czech. *OpenOffice.org Wiki* [online]. 2007. [cit. 2008-12-29]. URL: <http://wiki.services.openoffice.org/wiki/Dictionaries#Czech_.28Czech_Republic.29>

[42] PALA, K., SMRŽ, P. Building Czech Wordnet. *Romanian Journal of Information Science and Technology*. Budapest: 2004. s. 79-88.

[43] SEDLÁČEK, R. *Morfematický analyzátor češtiny*. Dizertační práce. Brno: Fakulta informatiky Masarykovy univerzity, 2004.

[44] YU, J., THOM, J. A., TAM, A. Ontology Evaluation Using Wikipedia Categories for Browsing. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*. Lisabon, Portugalsko: 2007. s. 223-232.

[45] Java-wikipedia-parser. *Google Code* [online]. 2008. [cit. 2009-01-02]. URL: <http://code.google.com/p/java-wikipedia-parser/>

[46] Greenwood, M. A. *Open-Domain Question Answering*. Dizertační práce. Velká Británie: Department of Computer Science, University of Sheffield, 2005.

Příloha

Součástí diplomové práce je **DVD**, které obsahuje:

- tuto technickou zprávu ve formátu *.doc a *.pdf
- zdrojový kód systému
- nastavení systému, definující chování v určitých situacích
- data české Wikipedie a index vytvořený systémem Lucene
- pomocné slovníky