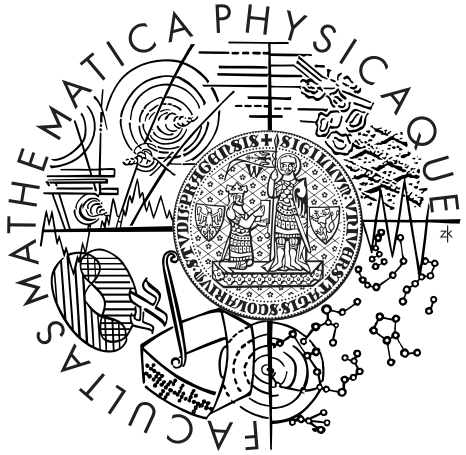
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

**BAKALÁŘSKÁ PRÁCE**




Jan Skalický

# Predikce vývoje ceny ropy na základě textových zpravodajských informací

Ústav formální a aplikované lingvistiky

Vedoucí bakalářské práce: RNDr. Ondřej Bojar Ph.D.
Studijní program: Informatika (B1801)
Studijní obor: programování

Praha 2012

Na tomto místě bych rád poděkoval Ondřeji Bojarovi za vedení bakalářské práce a také za jeho trpělivost, ochotu a inspiraci. Rovněž patří můj dík rodině a slečně za podporu, bez které bych nemohl práci dokončit.

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V ..... dne .....  Podpis autora

Název práce: Predikce vývoje ceny ropy na základě textových zpravodajských informací

Autor: Jan Skalický

Katedra: Ústav formální a aplikované lingvistiky

Vedoucí bakalářské práce: RNDr. Ondřej Bojar Ph.D., ÚFAL

Abstrakt: Pro předpověď vývoje ceny ropy existuje celá řada algoritmů. V této práci přinášíme nový pohled na tuto problematiku a představujeme náš projekt COPF. Pomocí klasifikátoru maximální entropie se snažíme předpovídat z textových informací dostupných na Internetu. Opíráme se o znalosti expertů v daném oboru. V rámci práce jsme testovali a vylepšovali úspěšnost systému COPF. Zjistili jsme, že tento přístup má mnoho problémů, které se ale dají řešit. V současném stavu naše úspěšnost sice překonala baseline, ovšem pro další vývoj je nutné získat více zdrojů dat. Naše metoda nebyla nikdy považována za nosnou, spíše může sloužit k vylepšení úspěšnosti předpovědí numerických algoritmů a v každém případě je zajímavá z hlediska možnosti dolování informací z textu.

Klíčová slova: předpověď, cena ropy, strojové učení, klasifikace textů, maxent

Title: Crude Oil Price Forecast based on Text News

Author: Jan Skalický

Department: Institute of Formal and Applied Linguistics

Supervisor: RNDr. Ondřej Bojar Ph.D., ÚFAL

Abstract: For crude oil price forecast, there is a whole range of algorithms. In this thesis we bring out a new perspective on this issue and introduce our project COPF. Using a maximum entropy classifier, we try to predict the change in crude oil price from text information available on the Internet. We are taking advantage of the knowledge of experts in the field. As a part of the thesis, we tested and improved COPF precision. We have found out that this approach poses a lot of interesting problems. In the current state, the precision of our prediction surpassed the baseline but for further development, it is necessary to obtain more data sources. Our algorithm has never been regarded as a self-standing method but it may nicely complement numerical algorithms.

Keywords: forecast, crude oil price, machine learning, text classification, maxent

# Obsah

**1 Úvod** **8**
1.1 Související práce . . . . . 8
1.2 Očekávané problémy . . . . . 8
1.3 Struktura bakalářské práce . . . . . 9

**2 Data** **10**
2.1 Články pro předpovědi . . . . . 10
2.2 Historie ceny ropy . . . . . 11

**3 Časová platnost informací v textech** **12**
3.1 Použité nástroje . . . . . 12
3.2 TimexTag . . . . . 12
3.3 Konverze formátů . . . . . 13
3.4 Možná vylepšení identifikace a normalizace časových údajů . . . 13

**4 Klasifikátor maximální entropie** **14**

**5 Treex** **15**
5.1 Exportní formát . . . . . 15

**6 Postup predikce** **17**
6.1 První fáze: filtrace relevantních textů . . . . . 17
6.2 Druhá fáze: predikce . . . . . 17

**7 Vytváření atributů** **18**

**8 Evaluace atributů** **19**
8.1 Atributy pro filtraci . . . . . 19
8.1.1 Ruční anotace . . . . . 19
8.2 Atributy pro odhad ceny . . . . . 20

**9 Experimenty** **23**
9.1 Charakteristika vstupních dat . . . . . 23
9.2 Testování a analýza evaluace . . . . . 25
9.3 Vyhodnocení predikce s defaultem . . . . . 31
9.4 Rozptyl . . . . . 32

**10 Vyhodnocení na nových datech** **34**

**11 Závěr** **36**
11.1 Cíle . . . . . 36
11.2 Vyhlídky do budoucna . . . . . 36

**Seznam použité literatury** **38**

**Seznam tabulek** **41**

**Seznam obrázků** **42**

**Seznam použitých zkratek** **43**

**Přílohy** **44**

**A Uživatelská dokumentace** **45**
A.1 Úvod . . . 45
A.2 Instalace . . . 45
A.3 Příprava . . . 46
A.3.1 Přiložená data na CD . . . 47
A.4 Použití . . . 48
A.5 Nástroje . . . 49
A.5.1 Nástroje pro filtraci . . . 49
A.5.2 Nástroje pro vizualizaci . . . 50

**B Programátorská dokumentace** **51**
B.1 Architektura systému . . . 51
B.1.1 Diagram aktivit . . . 51
B.2 Rozšíření systému . . . 53
B.2.1 Rozšíření funkčnosti . . . 53
B.2.2 Přidání zdrojů dat . . . 53
B.3 Externí komponenty . . . 54
B.3.1 Timextag . . . 54
B.3.2 Treex . . . 55
B.4 Externí knihovny . . . 55
B.4.1 OpenNLP MaxEnt . . . 55
B.4.2 JFreeChart . . . 55
B.4.3 HtmlCleaner . . . 56
B.5 Důležité datové struktury . . . 56
B.5.1 copf . . . 56
B.5.2 copf.downloadedtexts . . . 57
B.5.3 copf.downloaders . . . 57
B.5.4 copf.downloaders.pagedownloaders . . . 57
B.5.5 copf.downloaders.pricedownloaders . . . 57
B.5.6 copf.exportformatparser . . . 57
B.5.7 copf.featuremanagement . . . 58
B.5.8 copf.pricehistorymanagement . . . 58
B.5.9 copf.utilities.manualevaluation . . . 58
B.5.10 copf.utilities.textsconverters . . . 58
B.5.11 copf.visualization.featureevalvisualization . . . 59
B.6 Data . . . 59
B.6.1 TimexTag . . . 59
B.6.2 Treex . . . 60
B.6.3 Ceny ropy . . . 61
B.6.4 Stahování textů . . . 62
B.6.5 Atributy . . . 63
B.6.6 MaxEnt . . . 66

**C Obsah CD** **67**

**D Formáty souborů** **68**
D.1 Příklad SGML formátu . . . . . 68
D.2 Příklad výstupu komponenty TimexTag . . . . . 68

**E Sady atributů** **70**
E.1 Sada atributů pro 1. fázi experimentu . . . . . 70
E.2 Sada atributů pro 2. fázi experimentu . . . . . 71
E.3 Poslední sada atributů . . . . . 72

# 1. Úvod

Počítačové zpracování jazyka je velmi perspektivní obor pro budoucnost. Mnohokrát bychom rádi věděli, co si o dané věci myslí větší množina lidí (např. politici by rádi věděli, jak si vedou momentálně u voličů), jakými okruhy témat se právě teď zabývají světové noviny, rádi bychom si přeložili svůj text do jiného jazyka atd. A častokrát nám v těchto věcech může pomoci právě automatické zpracování jazyka.

Ještě žádanější, avšak méně přesnou skupinou úkolů jsou predikce rozličných jevů podle momentálních podmínek a jejich historie. Nejlepším příkladem je pro nás předpověď počasí, která už není ničím novým. Avšak je tu mnoho dalších oblastí a velká část z nich spadá pod vývoj finančního trhu (ceny akcií atd.).

Jednou velmi specifickou oblastí je predikce vývoje ceny ropy. Pro tento problém existuje celá řada typů algoritmů (příklady v literatuře [1], [2] a [3]). Většina z nich je založena pouze na číselných datech. Právě proto jsme se rozhodli vytvořit algoritmus využívající jiného principu predikce. Budeme vycházet z novinových článků zabývajících se ropou a vývojem její ceny do budoucna a pro predikci použijeme klasifikátor MaxEnt (popsáno v kapitole 4).

## 1.1 Související práce

V průběhu řešení jsme sice zjistili, že už existuje algoritmus předpovídající mimo jiné pomocí data miningu (dolování dat) a rozpoznávání různých vzorů v textu, avšak ten je založen na jiných principech a všímá si jiného druhu dat. Tento algoritmus je součástí celé metodologie nazvané TEI@I (více viz [1]). Metodologie TEI@I kombinuje více přístupů pro dosažení lepších výsledků. Přístup založený právě na data miningu prohledává Internet a z internetových stránek dělá abstrakty a dále je filtruje. Potom vyhledává a testuje různé vzory (odpovídají událostem, případně podmínkám, jaké by mohly výslednou cenu ropy ovlivnit). Hlavně se snaží vyzkoumat, které z těchto vzorů se vážou na neobvyklé výkyvy v ceně ropy. Příkladem takové události by mohla být válka v Libyi či Iráku. Naproti tomu se chceme v našem algoritmu opřít pouze o důvěryhodné zdroje, v rámci kterých si budeme všímat názoru expertů na budoucí vývoj.

## 1.2 Očekávané problémy

Náš přístup je svým způsobem jedinečný, avšak zřejmě se bude potýkat s mnoha problémy: důvěryhodností zdrojů, příliš velkou potřebou sesbíraných dat atd.

Z těchto důvodů na jednu stranu neočekáváme příliš povzbudivé výsledky, ale na druhou stranu tím můžeme začít vývoj nového algoritmu, který by mohl pomoci ostatním algoritmům k přesnějším výsledkům.

## 1.3 Struktura bakalářské práce

Naše práce začala sháněním důvěryhodných zdrojů na Internetu, stažením dat ve formě textů z těchto zdrojů a jejich úpravou (popsáno v 2.1). Dále jsme použili systém pro automatické zjištění, jakým časovým obdobím se texty zabývají (kapitola 3). Stáhli jsme z nalezeného zdroje na Internetu historii cen ropy potřebnou při učení klasifikátoru a evaluaci naší predikce (popsáno v 2.2). Získali jsme implementaci klasifikátoru MaxEnt (kapitola 4) a mohli jsme začít vymýšlet atributy, jimiž se MaxEnt bude řídit (kapitola 7) a pomocí nich predikovat (kapitola 6). Pro efektivnější vymýšlení a ohodnocování atributů jsme texty nechali oanotovat pomocí komponenty Treex (kapitola 5). Nakonec jsme vyhodnotili naše atributy a upravili je podle nalezených chyb (kapitola 8 a 9). Bohužel jsme zjistili, že pro účinnou evaluaci atributů je potřeba mnohem větší množina vstupních textů, ale přesto jsme provedli finální vyhodnocení atributů (kapitola 10), na kterém se některé rozdíly projevily. V tomto směru je nutná další práce, kdy bude potřeba najít více věrohodných zdrojů a na nich otestovat vymyšlené atributy.

# 2. Data

## 2.1 Články pro předpovědi

Data pro projekt jsme získali z Internetu. Zdrojem těchto dat mohly být internetové stránky, RSS kanály apod. Pro naše potřeby bylo nutné, aby tyto zdroje splňovaly několik základních předpokladů. Daný zdroj se samozřejmě musel týkat daného tématu, tzn. ceny ropy. Musel být psán v anglickém jazyce, protože struktura různých jazyků je velmi rozdílná a některé komponenty projektu jsou uzpůsobené pouze pro anglický jazyk (např. použitý systém pro identifikaci časových údajů). Texty (články) ze zdroje musely být dlouhodobě vydávané a muselo jich být relativně velké množství, aby se vyplatilo investovat úsilí do získání těchto textů a jejich následné úpravy do požadovaného formátu. Bylo potřeba, aby zdroje byly alespoň minimálně strukturované, aby se dalo zjistit datum jejich vzniku, téma a daný text bez HTML tagů. A ne na posledním místě bylo nutné, aby daný zdroj byl důvěryhodný, jinak by pro tento projekt neměl smysl.

Pro účely získání textů v dané podobě bylo nutné vytvořit ke každému zdroji, který splňoval všechny potřebné podmínky, vlastní program (skript) pro stahování. Tyto programy jsme nejdříve vytvářeli v Perlu, později v Jave, kvůli větší přehlednosti druhého a nedostatečné znalosti prvého jazyka. Při vytváření vznikla poměrně velká sada funkcí, která značně urychluje přidávání dalších programů pro stahování z nových zdrojů. Ty ovšem nikdy nemohou plně vyřešit chyby ve struktuře textů, které práci značně zpomalují. Toto nebylo hlavní náplní práce, proto v tomto směru bylo uděláno jen nutné minimum pro získání dat.

Pro experiment (kapitola 9) se nám podařilo stáhnout 5612 textů.

Stažené texty bylo nutné dále upravovat a čistit, např. kvůli špatnému formátování. Dobrým příkladem je nesprávné odsazení okolo teček, které způsobilo, že parsery špatně rozpoznaly konec věty a nebyly schopny správně fungovat. K tomuto účelu, opravě formátování, bylo opět vyvinuto několik malých programů (skriptů), které byly namátkově ručně kontrolovány. Domníváme se, že tyto opravy textů by opět vystačily na více než jednu bakalářskou práci, proto se v tomto směru udělalo opět jen nutné minimum pro použitelnost v rámci komponent projektu.

Dalším úkolem bylo vytvořit převodníky reprezentací textů. Každá externí komponenta potřebovala vstup v jiném formátu a výstup byl opět v jiném, proto bylo nutné vytvořit malé programy pro jejich převod. Texty byly stahovány do XML souborů, kde k nim byly přiloženy informace o zdroji, datu vytvoření atd. Formát XML byl zvolen pro svou flexibilitu, a protože se v průběhu práce

ukázalo, že je potřeba ukládat nové informace, bylo toto rozhodnutí opodstatněné. Další specifické formáty jsme museli vytvářet a zpracovávat pro rozpoznání časových určení (TimexTag, kapitola 3) a lingvistickou analýzu (Treex, kapitola 5).

## 2.2 Historie ceny ropy

Pro potřeby MaxEnt klasifikátoru jsme potřebovali získat historii cen ropy, abychom mohli testovat model a atributy, kterými jsme ohodnocovali získané texty. K tomuto účelu byl vytvořen program pro stahování této informace z Internetu. Vzhledem k tomu, že jsme chtěli udělat naši predikci univerzální a nechtěli jsme se omezovat na nějaký region či typ ropy, potřebovali jsme jakési průměrné hodnoty. Problémem bylo, že neexistuje žádná globální cena, ale existuje více klasifikací ropy (podle různých poměrů ve složení). Přesto existovaly stránky počítající jakýsi průměr cen a vydávající ho za světovou cenu ropy. Bohužel v průběhu prací na projektu tento zdroj zanikl a bylo nutné hledat nový. Nakonec jsme získali podrobná data pro dvě hlavní klasifikace ropy (Brent a WTI) a jejich průměr považujeme pro naše účely v projektu za globální cenu. Tuto historii získáváme v týdenních intervalech, což je pro nás dostatečně přesné. Zdroj poskytuje i denní interval, ale v tuto chvíli by nás evaluace po dnech spíše zdržovala, než aby přinášela užitek.

Cena ropy je vyjádřena standardně v dolarech na barel.

# 3. Časová platnost informací v textech

## 3.1 Použité nástroje

Pro zjištění časových období, kterými se texty zabývají, jsme využili komponentu TimexTag[1]. Jde o tzv. TIMEX2 systém. TIMEX2 je schéma pro detekci a normalizaci časových výrazů. TIMEX2 systém je implementace tohoto schématu. Pro zprovoznění tohoto systému bylo nutné stáhnout ještě další dvě komponenty, LIBSVM a Charniakův parser.

LIBSVM[2] je knihovna pro klasifikaci pomocí SVM (support vector machine). Jde o metodu strojového učení pomocí podpůrných vektorů. U této komponenty jsme měli problém s kompatibilitou nové verze s TimexTagem, ale to jsme nakonec vyřešili stažením starší verze LIBSVM.

Charniakův parser[3] je parser (tzn. syntaktický analyzátor) inspirovaný principem maximální entropie.

## 3.2 TimexTag

Způsob práce TimexTagu popisuje těchto šest kroků. Nejdříve se použije statické vyhodnocení pro nalezení a normalizaci plně kvalifikovaných výrazů, odpovídajících nějakému vzoru z velmi malé předem definované množiny. Poté přichází na řadu klasifikátor SVM, který se snaží identifikovat všechny časové výrazy. Pokud najde některé, které byly nalezeny i v prvém kroku, vypustí je z prvně nalezené množiny a zanechá je pouze v druhé. Po nalezení rozdělí tyto výrazy do šesti kategorií: bod, trvání, obecný bod, obecné trvání, opakování a nespecifikované. V každé z těchto kategorií dochází k porovnání s vlastní množinou vzorů. Výrazy nevyhovující ani jednomu ze vzorů se přesouvají do kategorie „nespecifikované“. Naopak u těch, které mají nějaký vyhovující vzor, dochází k prvotní normalizaci. Výrazy v kategorii „bod“ jsou podle referenčního data rozděleny do tří tříd: před, při, po. Nakonec jsou výrazy ve všech kategoriích, kromě „nespecifikované“, normalizovány do formátu, jenž je rozšířením standardu ISO 8601.

Pro identifikaci tagů klasického textu komponenta TimexTag dosahuje přesnosti (precision) 0,912 a úplnosti (recall) 0,786. Pro normalizaci to je přesnost

[1]K 21. březnu 2012 dostupné na: `http://ilps.science.uva.nl/resources/timextag`
[2]K 21. březnu 2012 dostupné na: `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`
[3]K 21. březnu 2012 dostupné na: `ftp://ftp.cs.brown.edu/pub/nlparser/`

0,850 a úplnost 0,732 (hodnoty převzaty z [12]). Toto jsou pro nás více než dostačující hodnoty.

## 3.3 Konverze formátů

Této komponentě jsme museli převést texty do SGML formátu (příklad viz příloha D.1). Výstupem byl XML soubor s množinou TIMEX2 tagů (příklad viz příloha D.2). Tyto tagy jsme potřebovali převést na časové období. Zde jsme prostě přečetli všechny tagy, u kterých došlo ke konečné normalizaci, našli jsme nejstarší a nejmladší časový údaj a udělali jsme z nich požadované časové období.

Po konečné normalizaci nebyly pro některé texty nalezeny žádné tagy. Těmto textům jsme nastavili čas, kterým se zabývají, na čas vydání článku (s malým rozmezím), což nám zaručilo, že se nikdy nevyskytnou v množině článků, podle kterých se předpovídá, ale zároveň se mohou zúčastnit trénování modelu. Pokud v nich nebyl žádný časový údaj, předpokládali jsme, že s velkou pravděpodobností budou mluvit o vývoji v přítomnosti a jejím blízkém okolí.

## 3.4 Možná vylepšení identifikace a normalizace časových údajů

Tato část projektu by se jistě dala optimalizovat, nicméně pro naše potřeby bylo toto určení dostatečné. Právě snaha lépe specifikovat časové období by byla v této fázi projektu spíše na škodu, protože čím přesnější a užší období, tím méně textů by se účastnilo předpovědi a tím hůře by se evaluovala kvalita atributů pro klasifikátor. Jinou možností vyhodnocení TIMEX2 tagů by bylo rozdělit text na více částí podle tagů a každou z částí považovat za samostatný text.

V budoucnosti by bylo zajímavé změřit závislost správnosti určení časového období pomocí TIMEX2 systému na správnosti odhadu výsledné ceny ropy. Zde bychom mohli vyzkoušet i další implementace TIMEX2 systému (např. TARSQI Toolkit).

# 4. Klasifikátor maximální entropie

Maximum entropy classifier (klasifikátor založený na principu maximální entropie), dále jen klasifikátor MaxEnt, je hlavní komponentou této bakalářské práce. Pro naše účely jsme ho získali z projektu Apache OpenNLP[1]. Vzhledem k jeho důležitosti je nutné alespoň nastínit, co se za tímto pojmem skrývá.

Klasifikátor je algoritmus, který je při vhodné množině znalostí schopen úspěšně rozdělovat vstupní data s hodnotami atributů (příznaků) do výstupních předem zvolených skupin (tříd). Tato vstupní data mohou být z mnoha heterogenních informačních zdrojů. Pro účely klasifikace se vstupní data ohodnotí vzhledem ke konečné množině atributů, které předem definuje experimentátor z informací důležitých pro klasifikaci. Každý atribut odpovídá jednomu omezení pro model klasifikátoru.

Princip maximální entropie je předpoklad, který říká, že pravděpodobnostní rozdělení nejlépe vystihující současný stav poznání je to s největší entropií. Říká tedy, že při hledání pravděpodobnostního rozdělení máme zvolit to, které nám nechá největší míru nejistoty v souladu s omezeními.

MaxEnt klasifikátor je příkladem metody tzv. supervised learning (učení s učitelem). Klasifikátor odhaduje data díky tzv. learning sample (trénovacím datům). Trénovací data sestávají ze dvojic vstupních objektů, vektorů příznaků (atributů) a požadovaného výstupu, který by měl klasifikátor vydat, kdyby byl daný objekt testován. Proces odhadování dat se potom skládá ze dvou částí, natrénování klasifikátoru z trénovacích dat a pak samotného výběru pomocí algoritmu klasifikátoru. V případě MaxEntu se vybírá model s největší entropií splňující daná omezení.

Přítomnost atributů ve vstupu se dá definovat dvěma způsoby podle druhu MaxEnt klasifikátoru, u prvního buď atribut v daném vstupu je, či není, u druhého se přítomnost atributu vyjadřuje reálným číslem. V našem projektu atributy momentálně měříme prvním způsobem. Druhý způsob má několik výhod: např. možnost vyjádřit četnost atd., proto bude v budoucnosti nutné přejít na způsob druhý. To nebude složité. Pro atributy, které budeme chtít stále měřit prvním způsobem, vyjádříme přítomnost např. číslem 1 a nepřítomnost číslem 0.

[1] K 21. březnu 2012 dostupné na: `http://opennlp.apache.org`

# 5. Treex

Treex[1] je modulární software pro zpracování přirozeného jazyka implementovaný pro Linux. Jednou z komplexních aplikací platformy Treex je strojový překlad z anglického jazyka do českého jazyka. V našem případě jsme ho použili pro automatickou analýzu textů na analytické a tektogramatické rovině (více viz [8], [9], [10] a [11]).

V tuto chvíli je nutné si nadefinovat několik termínů. Obyčejná věta je rozdělena na slova. Analogicky věta na analytické rovině je rozdělena na a-lemmata a věta na tektogramatické rovině je rozdělena na t-lemmata. Termínem token označujeme slovo, a-lemma nebo t-lemma.

## 5.1 Exportní formát

Výstupem naší analýzy Treex je pro jeden text jeden soubor. Tento soubor je buď v Treex formátu, nebo v tzv. Exportním formátu. Formát Treex je aplikací PML (Prague Markup Language), jazyku založeném na XML (více viz [9]). Pro nás je ovšem nepraktický pro svou velikost, a proto jsme se rozhodli pro Exportní formát. Ten sice neuchovává všechny atributy, ale pro nás je bohatě postačující a hlavně je několikanásobně menší než XML. V souboru v Exportním formátu je každá věta uložena na jeden řádek s částmi oddělenými tabulátorem v pořadí: ID věty (obsahující typ zdroje), analytická rovina, tektogramatická rovina, čísla, odkazy z t-lemmat na odpovídající a-lemmata s lexikální hodnotou a odkazy t-lemmat na jejich pomocná a-lemmata. Tokeny ve větě jsou odděleny mezerou a atributy tokenů svislítkem. Tento formát nám umožňuje snadné porovnávání atributů. Zde je příklad věty „Truly Exxon is not alone.“ (každý tabulátor je pro přehlednost nahrazeny dvěma novými řádky, mezera jedním):

```
DownloadedTextsInFiles/oil_prize_net/
   HowExxonpaidzerotaxesin200920100415.parsed.treex.gz:s7

Truly|truly|RB|1|4|Adv
,|,|,|2|4|AuxX
Exxon|Exxon|NNP|3|4|Sb
is|be|VBZ|4|0|Pred
not|not|RB|5|4|Neg
alone|alone|RB|6|5|Adv
.|.|.|7|0|AuxK
```

[1] K 25. březnu 2012 dostupné na: `http://ufal.ms.mff.cuni.cz/treex/`

```
truly|MANN|1|3|complex|adv:|adv.denot.grad.neg|-|neg0|-|-|-|-|-|-|
   pos|-|-|-|-|-|-|-|-|0|-|-
Exxon|ACT|2|3|complex|n:subj|n.denot|sg
   |-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|0|-|-
be|PRED|3|0|complex|v:fin|v|-|neg1|sim|ind|decl|-|-|-|-|disp0|-|it0
   |-|-|res0|-|-|1|-|-
alone|MANN|4|3|complex|adv:|adv.denot.grad.neg|-|neg0|-|-|-|-|-|-|
   pos|-|-|-|-|-|-|-|-|0|-|-

0-0 2-1 3-2 5-3

1-2 4-2
```

# 6. Postup predikce

Ve chvíli, kdy máme stažené texty a historii ceny ropy, kdy jsme pro všechny texty zjistili období, kterými se zabývají (TimexTag, kapitola 3), a provedli jejich automatickou analýzu (Treex, kapitola 5), můžeme postoupit k samotné predikci.

Proces predikce se skládá ze dvou fází. V první fázi filtrujeme texty, které se zabývají požadovanou tematikou. Druhou fází je samotná predikce.

## 6.1 První fáze: filtrace relevantních textů

V této fázi jsme nejdříve museli vytvořit množinu atributů pro klasifikátor a ručně ohodnotit větší množství textů. Na textech jsme hodnotili, zda jsou pro naše potřeby relevantní. Na základě tohoto vzorku jsme natrénovali klasifikátor a spustili jsme ho pro všechny texty. Podle výsledků klasifikátoru jsme určovali, zda daný článek prošel či neprošel filtrací. V našem projektu jsme dále používali pouze ty texty, které prošly filtrací. Atributy a ruční anotaci rozebíráme podrobně v kapitolách 7 a 8.

## 6.2 Druhá fáze: predikce

V rámci samotné predikce jsme postupovali takto. Nejdříve jsme vytvořili atributy pro klasifikátor. Poté jsme vzali texty, které prošly filtrací, a našli jsme v nich tyto atributy. Podle data, které jsme chtěli predikovat, jsme rozdělili období na trénovací a to, z kterého vezmeme texty k predikci (rozdělení viz 8.1). Pro každý text z trénovacího období jsme zjistili reálnou změnu ceny (z historie ceny ropy) a poté jsme na těchto datech natrénovali klasifikátor. V rámci této fáze jsme neprováděli žádnou manuální anotaci, za „správnou“ odpověď při trénování bereme právě zjištěnou reálnou změnu ceny.

Dále jsme vzali texty z období pro predikci a na atributech každého z nich jsme spustili klasifikátor MaxEnt (kapitola 4). Pro každý text jsme dostali jeden výsledek z předem nadefinované množiny. Celkový výsledek byl ten nejčastější výsledek (tj. predikovaný výsledek pro největší počet textů). Atributy a množinu výsledků rozebíráme podrobně v kapitolách 7 a 8.

# 7. Vytváření atributů

Inspiraci pro atributy jsme získávali v průběhu manuálního ohodnocování textů pro filtraci. Nejdřív jsme si mohli uvědomit, že množina atributů velmi úzce souvisí s množinou výsledků. Toto se muselo brát v potaz při vymýšlení vhodných atributů. Dále bylo nutné opravdu pro každý prvek z množiny řešení vymyslet atributy a nespoléhat na to, že absencí některých atributů dojde klasifikátor právě požadovaného řešení. Pokud by měl některý prvek z množiny řešení výrazně více atributů, které by na něj ukazovaly, byly by výsledky nevyrovnané. Toto bylo těžké zvláště pro prvek „stag“ (stagnace), kde se nám to nezdařilo úplně. Dále jsme se snažili ke slovům v atributech nalézt synonyma. V budoucnu by se to dalo řešit slovníkem synonym, pomocí kterého by se generovaly synonymní atributy.

Atributy jsme rozdělili na několik druhů. Prvním druhem je množina tokenů v textu, kterou jsme využívali hlavně ve fázi filtrování článků. Druhým je množina tokenů ve větě, která je praktičtější zase pro fázi odhadu, kvůli potřebě vzájemné závislosti tokenů v atributu. Další druhy atributů můžeme do projektu lehce přidat (např. posloupnost tokenů ve větě atd.).

Tokeny v atributech mohou být tří typů. Normální slova (v základním tvaru, podstatná jména v prvním pádu jednotného čísla, slovesa v infinitivu atd.), a-lemmata nebo t-lemmata (více viz 5). Jakého druhu jsou tokeny specifikujeme speciální trojicí písmen na jeho začátku. Pokud si vybereme token z analytické nebo tektogramatické roviny, můžeme mu specifikovat libovolné parametry dané roviny (více viz 5). Pro ostatní prostě nevyplníme prostor pro daný parametr. Tuto možnost přidávání různých typů parametrů jsme přidali z několika důvodů. Hlavním důvodem bylo, že jsme díky anotaci získali slova v základním tvaru, což nám velmi usnadnilo práci a razantně snížilo počet atributů. Dalším důvodem a výhodou byla snadná specifikace času (např. jestli jde o budoucnost či minulost, což je pro nás zvlášť důležité), nebo zjištění, zda je v dané větě zápor, který by mohl způsobit zásadní chybu v modelu klasifikátoru atd. Tektogramatickou rovinu jsme použili také pro určení času (budoucnosti, přítomnosti nebo minulosti) u atributů (samozřejmě pouze u sloves).

# 8. Evaluace atributů

Pro bakalářskou práci bylo nutné vytvořit dva druhy evaluací pro atributy článků. První je pro atributy filtrace (pomáhá nám se zbavit nepotřebných článků). Druhá je pro atributy odhadu vývoje ceny ropy z článků.

## 8.1 Atributy pro filtraci

Evaluace atributů, pomocí kterých filtrujeme pro nás nezajímavé články, se od té druhé liší v dimenzi řešení. Zatímco pro druhou evaluaci se počet výsledků může postupem času a s rozrůstajícím se projektem měnit, výsledkem první je buď ano (yes), nebo ne (no). Také pro nás není důležité mít v této evaluaci optimální procento úspěšnosti, ale stačí nám dostatečně dobré, protože filtrování pro nás není klíčové. Z těchto dvou důvodů a z důvodu rychlosti evaluace máme pro tuto evaluaci mnohem menší a hrubší množinu atributů.

Samotná evaluace atributů pro filtraci je náročná tím, že výsledky musíme kontrolovat na manuálně ohodnocených datech, jejichž získání je časově náročné. Proto jsme tuto evaluaci prováděli jen zběžně.

### 8.1.1 Ruční anotace

Pro testování atributů jsme museli ručně ohodnotit u vzorku textů, zda se zabývají daným tématem. Pro anotaci jsme si vytvořili jednoduchý program, který nás upozorňoval na důležitá slova v textu a usnadňoval nám práci.

Texty hodnotil pouze autor práce. Ohodnoceno bylo 282 textů, z toho 134 se jich zabývalo cenou ropy a 148 se jí nezabývalo. Po filtraci zůstalo pro experiment ze stažených 5612 textů dohromady 3419 textů.

V procesu anotace mohlo vzniknout určité procento chyb z důvodů subjektivity. I u takto přesně vymezeného tématu existují texty, u kterých by různí lidé hodnotili různě. Příkladem je následující část textu, který prošel manuální evaluací:

**Niko Resources Ltd. has signed four Offshore Production Sharing Agreements (PSA's) with the President of the Islamic Republic of Pakistan... Some key features of the fiscal terms include no contract signing bonus, 85% of oil or gas production available for cost recovery, royalty of 0% on gross production for first 48 months following start of commercial production and increases to a maximum of 12.5% in the 7th year of production, royalty of 5.0% on gross production for months 49 – 60, maximum Royalty of 12.5%, contractor receives 80% of Profit Oil up to first 100 mmbbl; 90% of Profit**

**Gas up to 100 mmboe; higher Profit Oil/Gas percent for recovery from deeper horizons, and oil pricing is based on basket of Arabian/Persian Gulf crude oils plus or minus quality differential.**

Tento článek byl nakonec ohodnocen kladně (tzn. anotátor se domnívá, že se zabývá cenou ropy, resp., že jeho znalost může ovlivnit predikci ceny ropy).

## 8.2 Atributy pro odhad ceny

Evaluací atributů pro odhad ceny se bylo nutné zabývat mnohem podrobněji. Samotný úkol je mnohem složitější a vyžaduje mnohem větší a lépe vybranou množinu atributů než filtrace. Také není úplně jednoduché správně změřit, jak dobré pro nás jsou určité atributy, případně jak je dobrá celá množina atributů.

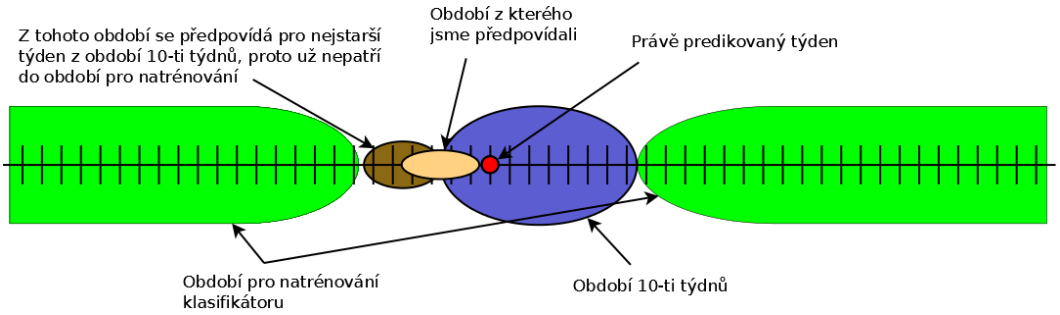
Množina řešení odhadu ceny je nadefinovaná v ohodnocovací funkci, která danému rozdílu začáteční a konečné ceny na daném intervalu přiřadí výsledek, zatím je tříprvková: nahoru (up), dolů (down) a stagnace (stag). Tuto množinu můžeme změnit, ale následně je vhodné přizpůsobit atributy změnám. Pro naši aktuální množinu jsme museli určit, co vlastně stagnace cen znamená, museli jsme ji nějak číselně vyjádřit. Nakonec jsme se rozhodli částečně intuitivně, částečně na základě evaluace. Hodnota byla v době psaní tohoto textu 0,5$ na barel.

Pro anulování škodlivého dopadu extrémů v historii cen na naše vyhodnocení jsme použili techniku křížové evaluace (cross-evaluation). Technika křížové evaluace se zakládá na rozdělení vstupů na několik částí a jejich postupného testování. V každé iteraci odložíme stranou malou část dat jako budoucí testovací množinu. Na zbylých datech natrénujeme klasifikátor. Nakonec tento klasifikátor vyhodnotíme na odložených datech. Tak nám vznikne skupina výsledků, která reprezentuje ohodnocení atributů. Z této skupiny pak můžeme zjistit, nejen jak jsou atributy úspěšné v průměru, ale i jak moc se od průměru mohou lišit nebo jak jsou na tom v různých extrémních případech.

V našich testech jsme vždy vybrali období, které odpovídalo deseti týdnům. Pak jsme pro každý tento týden vzali období třiceti dnů do minulosti posunuté zpět ještě o týden (týden byl naší predikční dobou) a z tohoto období jsme předpovídali pro daný týden. Klasifikátor jsme předtím natrénovali na celém zbylém období.

Pro období deseti týdnů jsme se rozhodli z důvodu časové optimalizace (nemusíme tolikrát trénovat klasifikátor), a také proto, že jsme chtěli měřit rozptyl predikce, což by na jednotlivých týdnech nemělo smysl (viz sekce 9.4).

Rozdělení období je vidět na obrázku 8.1.




Obrázek 8.1: Rozdělení období při jednom z cyklů evaluace

Nakonec zde uvádíme příklad evaluace jednoho článku. Nalezené tokeny atributů jsme pro přehlednost zvýraznili. Článek byl napsán 15.12.2009 a jeho zdrojem je internetová stránka `http://www.oilmarketer.co.uk/`.

**With about half an hour left in the floor trade session in New York on Tuesday, the price of January contracts for West Texas Intermediate crude oil had risen $1.14 to $70.65 per barrel on the New York Mercantile Exchange, after an upwardly revised estimate of 2010 consumption from the Organization of Petroleum Exporting Countires. OPEC upped its estimate of how much oil will be used next year by 70,000 barrels over its previous estimate on the heels of a similarly upgraded forecast on consumption issued last week by the Internationl Energy Agency. January contracts for Brent crude, meanwhile, added 23 cents to $72.12 per barrel on the ICE Futures Europe exchange in London. Nymex January gasoline futures were up 2 cents in afternoon trade to $1.85 per barrel while January heating oil futures also gained 2 cents to $1.93 per gallon. The retail prices of a gallon of gasoline in the United States, on the other hand, dropped 0.5 cent overnight to $2.598 gallon on average nationally.**

Reálný výsledek tohoto článku je „up“ a v predikci nám vyšlo „up“, tzn. predikce pro tento článek byla úspěšná. Také podle našeho úsudku článek hovoří o zvýšení ceny ropy do budoucnosti. Lze to poznat např. z toho, že budoucí smlouvy o prodeji ropy zdražují (January contracts for Brent crude, meanwhile, added 23 cents to $72.12 per barrel...). Zrovna tato část textu je zajímavá tím, že ji při momentální implementaci klasifikátor není schopen správně vyhodnotit. Pro správné vyhodnocení by bylo potřeba více provázat klasifikátor s komponentou pro určení času a ohodnocovat jednotlivé věty.

Článek je pro klasifikátor vyjádřen čtyřmi atributy. Tři atributy jsou typu množina tokenů ve větě a čtvrtý je atribut zdroje. Množiny tokenů ve větě jsou tyto: barrel up $; barrel cent up; price average. Vidíme také, že se nám podařilo vybrat článek, na jehož základě bychom mohli změnit množinu atributů. První dva nalezené atributy jsou sice z našeho pohledu korektní, ale třetí (price ave-

rage) považujeme spíše za atribut pro stagnaci. Toto by se dalo vyřešit vytvořením nového typu atributů, ve kterém bychom kromě množiny povinných tokenů kontrolovali ještě zakázané tokeny (v tomto případě drop). Dále je vidět, že bychom mohli přidat atribut typu množina tokenů ve větě pro tyto tokeny: barrel add cent.

# 9. Experimenty

## 9.1 Charakteristika vstupních dat

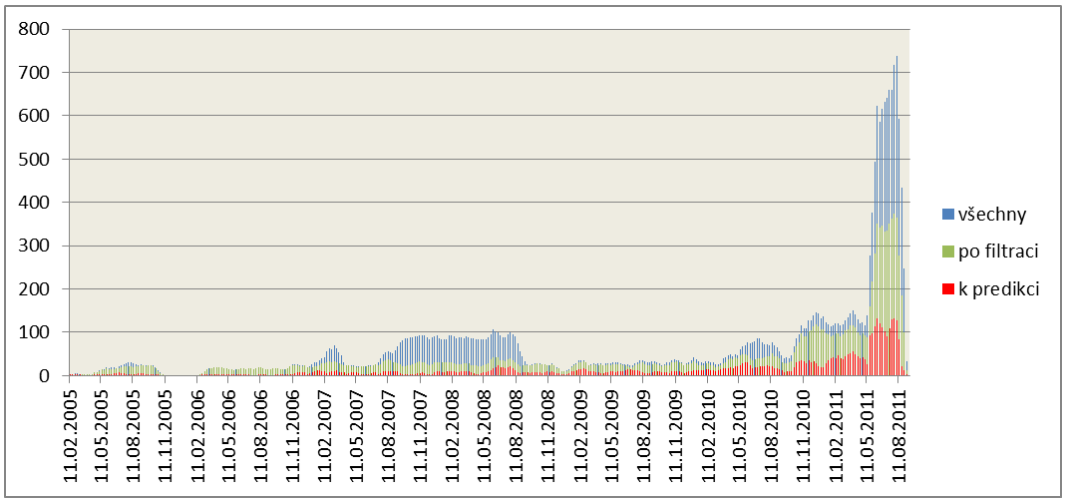
Většinu informací o vstupních datech jsme podali v kapitole 2, ovšem ještě jsme se chtěli podívat na data z jiného hlediska. Udělali jsme si graf výskytu textů v různých testovacích obdobích a tabulku shrnující průměrná data, abychom přiblížili, na jakém vzorku a za jakých podmínek provádíme naše predikce. Viz tabulka 9.1 a obrázek 9.1, který ukazuje rozložení textů mezi týdny. Pro každý týden zde jsou tři hodnoty. Všechny stažené texty napsané v rámci období pro předpověď vývoje ceny ropy na daný týden, z nich vyfiltrované texty tak, že zbyly pouze takové, které se zabývají cenou ropy, a nakonec texty, které šly k predikci, tzn. texty, které se zabývaly obdobím, jež obsahovalo predikované období a zároveň úspěšně prošly filtrem. Krajní období (do začátku roku 2005 a od poloviny srpna roku 2011), pro které nemáme žádné texty, jsme do tohoto grafu nezahrnuli.

Tabulka se zabývá stejnými daty jako graf. Postupně uvádíme podle sloupců: Kolik máme maximálně textů pro týden. Kolik máme průměrně textů pro týden v rámci období, pro které máme údaje o ceně ropy. Průměrný počet textů pro týden z období, které popisuje graf 9.1. Průměrný počet textů pro týden s alespoň jedním článkem. Počty týdnů s alespoň jedním článkem daného typu.

Nejdůležitější pro nás je počet týdnů, které mají alespoň jeden článek, na jehož základě lze vývoj ceny ropy v tomto týdnu predikovat. Těchto týdnů je 315. Pro tyto týdny máme průměrně 15,43 podkladových článků.

| | maximálně pro týden | průměrně pro týden | průměrně pro týden v období od 11.2.2005 do 2.9.2011 | průměrně pro týden s $> 1$ textem | týdnů s článkem |
|---|---|---|---|---|---|
| všech | 739 | 17,29 | 68,51 | 71,42 | 329 |
| po filtraci | 374 | 10,54 | 41,76 | 43,53 | 329 |
| k predikci | 133 | 3,58 | 14,17 | 15,43 | 315 |

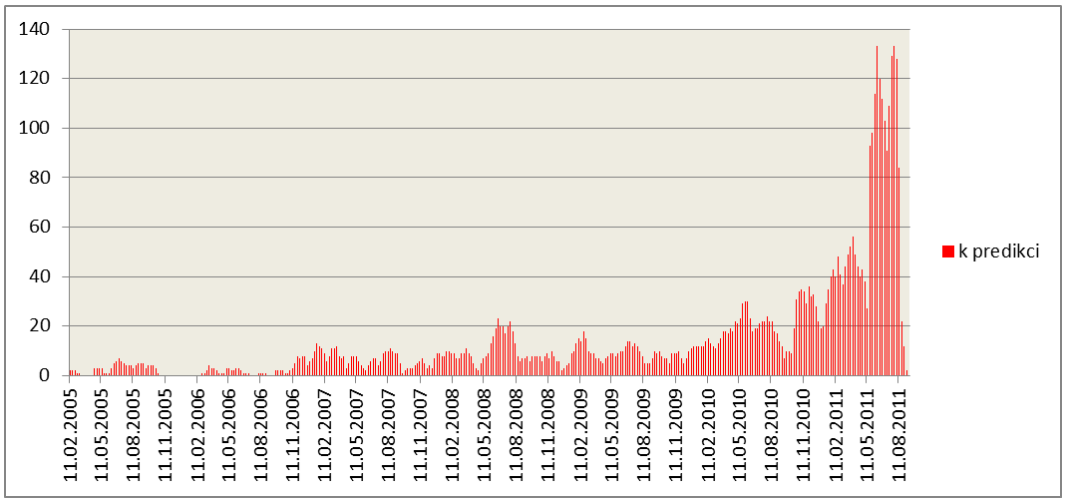
Tabulka 9.1: Souhrn rozložení textů




Obrázek 9.1: Počty textů z období pro předpověď vývoje ceny ropy na daný týden

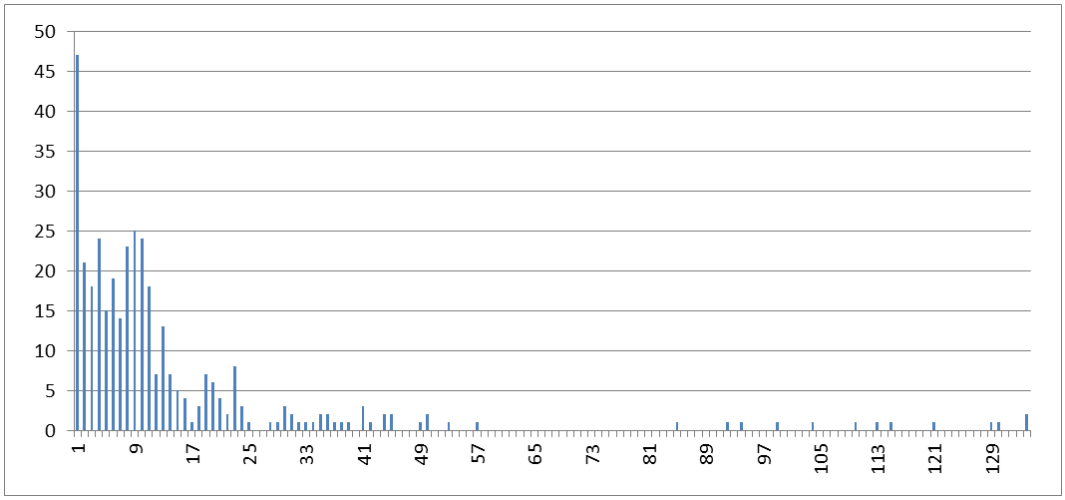
Když se podíváme na graf, vidíme, že jeho pravá část se pohybuje v úplně jiných hodnotách. To je způsobeno tím, že jsme naše zdroje hledali právě někdy v polovině roku 2011, což je konec naší osy. Tyto zdroje (webové stránky) většinou vznikly poměrně pozdě vzhledem k našemu období, případně neposkytují celou svou historii. Pro rovnoměrné rozdělení v grafu by bylo nutné kontinuálně získávat nové zdroje a stahovat nové články.

Zde je ještě jeden graf ukazující pouze rozložení textů, které se dostaly k predikci (obrázek 9.2).




Obrázek 9.2: Rozložení počtů textů k predikci po týdnech

Z rozložení vidíme, že naším prvním problémem bude nedostatek vstupních textů. Pro mnoho období nemáme ani jeden text a pro mnoho dalších jich je velmi málo. Z grafu na obrázku 9.3 je patrné, že téměř dvě třetiny týdnů jsou podporovány méně než deseti texty. Osa x v grafu vyjadřuje počet textů za týden a osa y počet týdnů s daným počtem textů.




Obrázek 9.3: Rozložení četnosti počtů textů k predikci (zanedbána krajní období bez textů)

## 9.2 Testování a analýza evaluace

Po několika počátečních testech a opravách chyb v programu jsme dostali výsledky (pro test jsme použili 1. sadu atributů, příloha E.1) uvedené v tabulce 9.2:

| | predikce | | | | |
|---|---|---|---|---|---|
| skutečnost | down | stag | up | celkem | bez hodnoty |
| down | 82 | 0 | 5 | 87 | 282 |
| stag | 27 | 0 | 0 | 27 | 529 |
| up | 112 | 0 | 1 | 113 | 321 |
| celkem | 221 | 0 | 6 | 227 | 1132 |
| Úspěšnost: | | | 0,3656 | | |
| Baseline: | | | 0,4978 | | |

Tabulka 9.2: Výsledky 1. fáze

Úspěšnost naší evaluace jsme počítali podělením počtu správně predikovaných období počtem všech období (pominuli jsme období, pro která jsme nemohli predikovat, protože pro ně nebyl žádný text):

$$\frac{82+0+1}{227} = 0,3656 \tag{9.1}$$

Baseline je pravděpodobnost úspěchu, pokud bychom při každé predikci odpověděli nejčastějším výsledkem, tzn. „up“:

$$\frac{113}{227} = 0,4978 \tag{9.2}$$

Po vypočtení je vidět, že úspěšnost našeho algoritmu nepřesahuje baseline. Pokud by se nám ji ani v budoucnu nepodařilo překonat, neměl by algoritmus smysl. Více o problematice baseline v sekci 9.3.

Z výsledků první fáze je zřejmý zmíněný nedostatek vstupních textů. Pro některá období v naší cross-evaluaci nebyl nalezen ani jeden vhodný článek (sloupec „bez hodnoty“), čímž se trochu zhoršil její efekt v anulování extrémů (Např.: Pokud by v určitém dlouhém období šly ceny jenom nahoru a my jsme testovali atributy pouze na tomto období, v případě, že později půjdou dolů, klasifikátor to nebude schopen předpovědět, protože se naučil, že ceny jdou za jakýchkoli podmínek nahoru). Pro mnoho dalších období bylo článků velmi málo k přesnější predikci. Tento problém se bude muset v budoucnu řešit, ovšem vzhledem k jeho časové náročnosti, jsme se zatím věnovali jiným aspektům viditelným v našich výsledcích.

Pokud pomineme období, pro která jsme neměli data (sloupec „bez hodnoty“), vidíme, že v předpovědích značně převyšuje hodnota „down“. Ručně jsme provedli analýzu textů a zjistili jsme, že v celé řadě atributy odpovídají našim předpokladům, ovšem předpovězená hodnota není správná. V rámci této ruční analýzy jsme znovu prošli naše atributy a snažili se je vylepšit. Tím jsme dostali druhou sadu atributů (příloha E.2). Zde jsou naše výsledky (tabulka 9.3):

| | predikce | | | | |
|---|---|---|---|---|---|
| skutečnost | down | stag | up | celkem | bez hodnoty |
| down | 111 | 0 | 6 | 117 | 252 |
| stag | 34 | 0 | 5 | 39 | 517 |
| up | 149 | 0 | 10 | 159 | 275 |
| celkem | 294 | 0 | 21 | 315 | 1044 |
| Úspěšnost: | | | 0,3841 | | |
| Baseline: | | | 0,5048 | | |

Tabulka 9.3: Výsledky 2. fáze

Zřejmě se nám podařilo vylepšit naše atributy, stojí za to si všimnout, že se nám podařilo ohodnotit alespoň jedním atributem mnoho dalších článků, takže jsme získali data pro předpověď v dalších 88 případech. I proto je i baseline jiná. Ovšem stále nám v předpovědích drtivě převažuje hodnota „down“. Udělali jsme si statistiku článků a přišli jsme na to, že článků předpovídajících propad ceny ropy bylo o mnoho více než článků předpovídajících vzestup. Stagnace na tom byla v počtu článků ještě mnohem hůře. Následující tabulka ukazuje rozdělení článků podle výsledků predikce pro období, kterým se články zabývají (v prvním sloupci), a rozdělení článků podle opravdové změny cena ropy od období napsání článku do období, kterým se článek zabývá (v druhém sloupci). Viz tabulka 9.4:

| | predikce | skutečný vývoj |
|---|---|---|
| down | 2775 | 1879 |
| stag | 4 | 283 |
| up | 640 | 1257 |

Tabulka 9.4: Rozložení článků mezi výsledky

Dále jsme si vypočítali, jaké bylo procentuální zastoupení výsledků pro skutečné hodnoty cen ropy (počítáno ze změny ceny za týden) a procentuální zastoupení výsledků pro články. Viz tabulka 9.5.

| | počet týdnů | % | počet článků | % |
|---|---|---|---|---|
| down | 117 | 0,3714 | 1879 | 0,5496 |
| stag | 39 | 0,1238 | 283 | 0,0828 |
| up | 159 | 0,5048 | 1257 | 0,3676 |
| celkově: | 315 | 1 | 3419 | 1 |

Tabulka 9.5: Reálné rozložení výsledků, tj. v kolika týdnech cena klesala, stagnovala a stoupala

Na naší tříprvkové množině řešení bylo vidět, že počty článků pro „down“, „up“ a „stag“ jsou hodně nevyrovnané. Pro potřeby klasifikátoru by měly poměry mezi počty textů odpovídat poměrům mezi počty skutečných výsledků. V opačném případě budou předpovědi klasifikátoru vždy nejvíce nakloněny hodnotě s největším počtem článků (s našimi daty hodnotě „down“). Zřejmě z tohoto důvodu jsme předpovídali tak špatně a nevyrovnaně. Proto jsme využili techniku tzv. undersamplingu. Tato technika řeší náš problém nevyrovnaných počtů vstupních textů jejich snížením pro výsledky, u kterých jich je mnoho (u nás by to bylo u „up“ a „down“).

Pro rychlé ověření naší hypotézy jsme nejdříve zkusili snížit počet článků pro „down“ (protože jich bylo zdaleka nejvíce) na počet článků „up“ a pozorovali jsme, jak to změní rozložení hodnot v naší kontingenční tabulce. Články, které jsme vyškrtli z množiny pro učení klasifikátoru jsme vybírali z dané množiny článků s výsledkem „down“ náhodně, proto se mohou výsledky v různých případech nepatrně lišit. Výsledky jsou uvedeny v tabulce 9.6.

| | predikce | | | | |
|---|---|---|---|---|---|
| skutečnost | down | stag | up | celkem | bez hodnoty |
| down | 80 | 0 | 37 | 117 | 252 |
| stag | 26 | 0 | 13 | 39 | 517 |
| up | 110 | 0 | 49 | 159 | 275 |
| celkem | 216 | 0 | 99 | 315 | 1044 |
| Úspěšnost: | | | 0,4095 | | |
| Baseline: | | | 0,5048 | | |

Tabulka 9.6: Výsledky 3. fáze

Data se nám lépe rozmístila po tabulce, zčásti se přesunula k hodnotám „up“. Tím jsme si potvrdili, že naše domněnka byla správná, proto jsme počty článků pro učení klasifikátoru snížili tak, aby poměry odpovídaly reálnému rozložení, a provedli jsme další test. Výsledky viz tabulka 9.7.

| | predikce | | | | |
|---|---|---|---|---|---|
| skutečnost | down | stag | up | celkem | bez hodnoty |
| down | 26 | 0 | 91 | 117 | 252 |
| stag | 9 | 0 | 30 | 39 | 517 |
| up | 49 | 0 | 110 | 159 | 275 |
| celkem | 84 | 0 | 231 | 315 | 1044 |
| Úspěšnost: | | | 0,4317 | | |
| Baseline: | | | 0,5048 | | |

Tabulka 9.7: Výsledky 4. fáze

Dosáhli jsme lepších výsledků i jejich lepšího rozložení, otázkou ale je, proč klasifikátor neurčil ani pro jedno období výsledek „stag“. Mohlo to být kvůli špatně vybraným atributům, proto jsme zkusili další test, kde jsme porovnali 6 různých konfigurací atributů.

Pro první konfiguraci jsme použili všechny momentálně nalezené atributy, zahrnuli jsme určení času pro slovesa pomocí tektogramatické roviny (zde jsme se snažili hledat slovesa směřující do budoucnosti či přítomnosti) a přidali jsme atribut určující zdroj dat. Výsledky jsou v tabulce 9.8.

| | predikce | | | | |
|---|---|---|---|---|---|
| skutečnost | down | stag | up | celkem | bez hodnoty |
| down | 28 | 0 | 89 | 117 | 252 |
| stag | 11 | 0 | 28 | 39 | 517 |
| up | 42 | 0 | 117 | 159 | 275 |
| celkem | 81 | 0 | 234 | 315 | 1044 |
| Úspěšnost: | | | 0,4603 | | |
| Baseline: | | | 0,5048 | | |

Tabulka 9.8: Výsledky 1. konfigurace

Druhá konfigurace je stejná jako první, až na to, že jsme se nezabývali určením času u sloves. Výsledky viz tabulka 9.9.

| | predikce | | | | |
|---|---|---|---|---|---|
| skutečnost | down | stag | up | celkem | bez hodnoty |
| down | 22 | 0 | 95 | 117 | 252 |
| stag | 11 | 0 | 28 | 39 | 517 |
| up | 45 | 0 | 114 | 159 | 275 |
| celkem | 78 | 0 | 237 | 315 | 1044 |
| Úspěšnost: | | | 0,4317 | | |
| Baseline: | | | 0,5048 | | |

Tabulka 9.9: Výsledky 2. konfigurace

Třetí konfigurace se od druhé liší tím, že zde chybí atribut určující zdroj. Výsledky viz tabulka 9.10.

| | predikce | | | | |
|---|---|---|---|---|---|
| skutečnost | down | stag | up | celkem | bez hodnoty |
| down | 37 | 0 | 80 | 117 | 252 |
| stag | 10 | 0 | 29 | 39 | 517 |
| up | 59 | 0 | 100 | 159 | 275 |
| celkem | 106 | 0 | 209 | 315 | 1044 |
| Úspěšnost: | | | 0,4349 | | |
| Baseline: | | | 0,5048 | | |

Tabulka 9.10: Výsledky 3. konfigurace

Ve čtvrté konfiguraci je jen minimum atributů, abychom zjistili, jak velký vliv mají naše atributy na výsledné hodnoty. Výsledky viz tabulka 9.11.

| | predikce | | | | |
|---|---|---|---|---|---|
| skutečnost | down | stag | up | celkem | bez hodnoty |
| down | 8 | 0 | 109 | 117 | 252 |
| stag | 2 | 0 | 37 | 39 | 517 |
| up | 17 | 0 | 142 | 159 | 275 |
| celkem | 27 | 0 | 288 | 315 | 1044 |
| Úspěšnost: | | | 0,4762 | | |
| Baseline: | | | 0,5048 | | |

Tabulka 9.11: Výsledky 4. konfigurace

Na těchto výsledcích vidíme drtivou převahu odhadů hodnoty „up“.

Pátá konfigurace se od třetí liší tím, že úplně postrádá atributy vedoucí podle nás k výsledku „stag“. Touto a další konfigurací se snažíme dosáhnout alespoň nějaké předpovědi výsledku „stag“. Výsledky viz tabulka 9.12.

| | predikce | | | | |
|---|---|---|---|---|---|
| skutečnost | down | stag | up | celkem | bez hodnoty |
| down | 29 | 0 | 88 | 117 | 252 |
| stag | 9 | 0 | 30 | 39 | 517 |
| up | 48 | 0 | 111 | 159 | 275 |
| celkem | 86 | 0 | 229 | 315 | 1044 |
| Úspěšnost: | | | 0,4444 | | |
| Baseline: | | | 0,5048 | | |

Tabulka 9.12: Výsledky 5. konfigurace

Šestá konfigurace je složena z drtivé většiny z atributů podle našeho úsudku vedoucích k výsledku „stag“. Výsledky viz tabulka 9.13.

| | predikce | | | | |
|---|---|---|---|---|---|
| skutečnost | down | stag | up | celkem | bez hodnoty |
| down | 4 | 0 | 113 | 117 | 252 |
| stag | 1 | 0 | 38 | 39 | 517 |
| up | 10 | 0 | 149 | 159 | 275 |
| celkem | 15 | 0 | 300 | 315 | 1044 |
| Úspěšnost: | | | 0,4857 | | |
| Baseline: | | | 0,5048 | | |

Tabulka 9.13: Výsledky 6. konfigurace

Zde vidíme také drtivou převahu odhadů hodnoty „up“.

Z výsledků všech našich konfigurací na první pohled vidíme, že se nám nepodařilo ani v jednom případě donutit klasifikátor předpovědět stagnaci. Nějaký vliv na to může mít skutečnost, že se o stagnaci cen tolik nemluví. Také jsme zjistili, že v případě malého počtu atributů klasifikátor odpovídá s drtivou převahou jedním výsledkem.

Pak jsme si uvědomili, že jsme články, pro které nebyl nalezen ani jeden atribut, vyjadřovali atributem „no_feature“ (sloužil pro zlepšení odhadu ve prospěch nejčastější položky). Toto také přispívalo k naší chybě. Proto jsme se rozhodli články s tímto atributem odstranit z trénování. Po opětovném spuštění nám vyšly jiné výsledky. Pro konfigurace s malým počtem atributů se výsledky zhoršily, ale lépe se rozložily do celé tabulky, opět bohužel bez jediné předpovědi výsledku „stag“. Na ostatní konfigurace tento atribut neměl vliv.

Zajímavé bylo, že nám nepřibylo týdnů bez článku. Poté jsme zjistili, že i pro malý počet dobře vybraných atributů (např. v naší minimální konfiguraci) článků bez atributů není mnoho.

Dále jsme z předchozího testu vypozorovali, že atribut zdroje má na naše výsledky jen nepatrný vliv. Opět pouze konfiguracím s malým počtem atributů viditelně změní výsledky.

Výsledky vyhodnocení všech konfigurací jsou pouze orientační a mohou se při každém spuštění mírně lišit, a to z důvodu použití undersamplingu a jeho náhodného výběru článků pro učení. Ovšem po několika testech a výše uvedených úpravách nám nakonec nejlépe vyšla konfigurace se všemi atributy (viz tabulka 9.14). Ostatní konfigurace jsou na tom o něco hůře a co se týče rozložení, jsou nyní podobné té nejlepší.

| | predikce | | | | |
|---|---|---|---|---|---|
| skutečnost | down | stag | up | celkem | bez hodnoty |
| down | 29 | 0 | 88 | 117 | 252 |
| stag | 9 | 0 | 30 | 39 | 517 |
| up | 38 | 0 | 121 | 159 | 275 |
| celkem | 76 | 0 | 239 | 315 | 1044 |

| | |
|---|---|
| Úspěšnost: | 0,4762 |
| Baseline: | 0,5048 |

Tabulka 9.14: Průměrné výsledky konfigurace se všemi atributy (1. konfigurace)

Od prvního výsledku jsme udělali ohromný posun o více jak deset procent. Ovšem nepodařilo se nám za žádnou cenu dosáhnout toho, aby klasifikátor předpověděl výsledek „stag“.

## 9.3 Vyhodnocení predikce s defaultem

Už od počátku experimentů nám přišlo divné, že se nám nepodařilo dostat úspěšnost nad baseline. To znamená, že pro úlohu odhadu ceny ropy by bylo úspěšnější pokaždé zvolit výsledek „up“.

Zamysleli jsme se nad tím, jak baseline počítáme a co jsme pro ni označili za referenční množinu. Doteď jsme baseline počítali pouze z týdnů, pro které jsme měli nějaké články. Proto jsme vyzkoušeli spočítat úspěšnost ze všech týdnů s tím, že pro týdny bez článku jsme zvolili nejčastější hodnotu. Zde bylo zajímavé, že nejčastější pro všechny týdny nebyla hodnota „up“, jako tomu bylo pro týdny, pro které byl alespoň jeden článek, ale hodnota „stag“. To je pro nás také dokladem, že o stagnaci se tolik nepíše. Pro naposledy počítanou konfiguraci jsme tedy spočítali tento druh baseline a úspěšnosti (viz 9.15).

$$\text{Úspěšnost s defaultem stag: } \frac{29 + 121 + 517}{1359} = 0,4908 \tag{9.3}$$

$$\text{Baseline (stag): } \frac{39 + 517}{1359} = 0,4091 \tag{9.4}$$

| | predikce | | | | |
|---|---|---|---|---|---|
| skutečnost | down | stag | up | celkem | bez hodnoty |
| down | 29 | 0 | 88 | 117 | 252 |
| stag | 9 | 0 | 30 | 39 | 517 |
| up | 38 | 0 | 121 | 159 | 275 |
| celkem | 76 | 0 | 239 | 315 | 1044 |

| | |
|---|---|
| Úspěšnost s defaultem stag: | 0,4908 |
| Baseline (stag): | 0,4091 |

Tabulka 9.15: Jiné počítání baseline a úspěšnosti pro konfiguraci se všemi atributy (1. konfigurace)

Je vidět, že je pro nás výhodné použít vyhodnocení s defaultní hodnotou a započítat všechny týdny. Zvýšila se nám úspěšnost a zároveň jsme se dostali vysoko nad baseline.

Pokud bychom ovšem měli ve sloupci „bez hodnoty“ samé nuly, použitím vyhodnocení s defaultní hodnotou bychom úspěšnost nijak nezměnili.

## 9.4 Rozptyl

Jak už jsme se dříve zmínili, evaluace je rozdělena na období po deseti týdnech. Jedním z důvodů je možnost měření rozptylu pro procentuální zastoupení úspěšných predikcí v jednotlivých obdobích. Pro měření jsme použili naši nejlepší konfiguraci atributů (1. konfigurace) a uplatnili jsme vyhodnocení s defaultní hodnotou. Pro toto nastavení nám vyšel roztyl 0,07575. Z toho usuzujeme, že náš algoritmus má poměrně vysokou variabilitu. Je to zřejmě způsobeno velkým nepoměrem počtu textů pro různá období. Rozptyl našeho algoritmu jsme porovnali s rozptylem baseline, který vyšel 0,08779. Snadno vidíme, že jsme předčili baseline i v tomto směru.

V rámci porovnání našeho algoritmu s baseline jsme provedli tzv. paired bootstrap resampling (více informací viz [15]), který se používá pro porovnání kvality dvou algoritmů. V rámci tohoto testu jsme pracovali opět s obdobími po deseti týdnech, na kterých jsme algoritmy srovnávali. Pro každé období jsme zjistili, který z algoritmů předpověděl vícekrát správně. Poté jsme sečetli výhry algoritmů a vyšlo nám, že ve 29 případech vyhrál náš algoritmus, ve 3 případech baseline a ve 103 případech byly oba algoritmy stejně úspěšné. To je dáno tím, že pro období, pro která nemáme potřebná data, předpovídáme právě hodnotou baseline (viz sekce 9.3). Se spolehlivostí 97,8 % tedy můžeme říci, že náš algoritmus není horší než baseline (viz výpočet 9.5).

$$\text{Spolehlivost: } 1 - \frac{3}{3 + 29 + 103} = 0,978 \tag{9.5}$$

Dosud jsme používali jen jeden vzorek dat. Kvůli vysoké variabilitě naměřené na dosavadních datech a jejich možné specifičnosti jsme se rozhodli vyhodnotit náš algoritmus s vybranými atributy na jiné nepoužité a předem nezkoumané množině dat (viz následující kapitola).

# 10. Vyhodnocení na nových datech

V této kapitole vyhodnocujeme dosud nejlepší model na zcela nových datech. Stáhli jsme nové texty a provedli jsme s nimi všechny potřebné procedury. Nové texty jsme stahovali z období 24 týdnů (6 měsíců). Dohromady jich bylo 1464.

Dále jsme se museli rozhodnout na kterých datech budeme trénovat klasifikátor, zda na starých nebo na nových. Z počátku jsme zkusili natrénovat klasifikátor na nových datech, tzn. při evaluaci jsme postupovali stejně jako při experimentu. Pro každé období jsme vybrali texty k predikci a na zbytku jsme natrénovali klasifikátor (viz kapitola 8). Tímto způsobem nám ale vycházely dost různé výsledky, protože náš vzorek pro natrénování byl příliš malý a protože jsme v rámci experimentu doimplementovali do projektu undersampling s náhodným výběrem textů (viz 9.2).

Proto jsme zvolili druhou standardní možnost, natrénovat klasifikátor na celé množině starých textů. Použili jsme opět křížovou evaluaci, jen jsme v jejím rámci klasifikátor už znovu netrénovali. Zde jsou naše průměrné výsledky pro první i druhý způsob evaluace (viz tabulky 10.1 a 10.2).

| | predikce | | | | |
|---|---|---|---|---|---|
| skutečnost | down | stag | up | celkem | bez hodnoty |
| down | 6 | 0 | 3 | 9 | 0 |
| stag | 3 | 0 | 2 | 5 | 0 |
| up | 5 | 0 | 5 | 10 | 0 |
| celkem | 14 | 0 | 10 | 24 | 0 |
| Úspěšnost: | | | 0,4583 | | |
| Baseline: | | | 0,4167 | | |

Tabulka 10.1: Vyhodnocení s učením na nových textech

| | predikce | | | | |
|---|---|---|---|---|---|
| skutečnost | down | stag | up | celkem | bez hodnoty |
| down | 0 | 1 | 8 | 9 | 0 |
| stag | 0 | 2 | 3 | 5 | 0 |
| up | 1 | 0 | 9 | 10 | 0 |
| celkem | 1 | 3 | 20 | 24 | 0 |
| Úspěšnost: | | | 0,4583 | | |
| Baseline: | | | 0,4167 | | |

Tabulka 10.2: Vyhodnocení s učením na starých textech

Ze všeho nejdříve jsme si všimli, že se nám podařilo pro každý týden stáhnout

alespoň jeden článek, který byl použit pro jeho predikci, viz nuly ve sloupci „bez hodnoty“. Proto náš trik s defaultem neměl na tato data vliv.

V naší první tabulce máme rozložení hodnot i výsledky podobné jako na konci experimentu na starých datech. Zato v druhé tabulce vidíme, že klasifikátor poprvé předpověděl výsledek „stag“, a to dokonce pro několik výsledků. Tímto jsme dokázali, že náš algoritmus a naše atributy nejsou úplně chybné. Zároveň je z těchto dvou tabulek vidět, že je potřeba vytvořit větší a lepší vzorek pro testování.

Vyhodnocení s učením na starých i nových datech jsme spustili několikrát a pozorovali jsme, že pro učení na starých textech (mnohem větší množina než texty nové) byly výsledky mnohem stabilnější, že je náhodný výběr v undersamplingu tolik neovlivnil.

V těchto experimentech se nám podařilo překonat baseline, což dokládá smysluplnost našeho postupu. Je ovšem nutné vzít v úvahu i změněné rozložení skutečných výsledků. V tomto případě jsou hodnoty rovnoměrněji rozdělené.

V neposlední řadě chceme poukázat na očekávaný rozdíl úspěšnosti mezi experimentem, kde jsme atributy sestavovali podle testovaných článků, a tímto vyhodnocením, jehož články jsme nikdy nečetli, ani jsme je předtím nenechali nijak analyzovat počítačem. Tento rozdíl je ovšem poměrně malý, což z části může být způsobeno používáním stejných zdrojů článků nebo tím, že se nám podařilo dobře rozpoznat obecné atributy skupin článků pro dané výsledky. K malému rozdílu jistě přispívá také naše nízká celková úspěšnost.

# 11. Závěr

## 11.1 Cíle

Cílem naší práce bylo vytvořit systém, který bude na základě textové informace predikovat následující vývoj ceny ropy, a to pomocí techniky strojového učení. Toto se nám do značné míry podařilo, ovšem ještě je možné velmi pokročit s automatizací procesu predikce. V další fázi se pokusíme vyrobit grafické prostředí pro uživatele, aby bylo možné snadno spouštět všechny potřebné skripty v programu a aby se usnadnila manipulace s texty a jejich výsledky.

Dále jsme chtěli vytvořit co nejlepší sadu atributů, optimalizovanou pomocí části textů. Podařilo se nám to poměrně dobře s tím, že tato kapitola vývoje nemůže být nikdy uzavřena a skrývá velký potenciál pro zlepšení úspěšnosti predikce. Ovšem bude nutné tyto příznaky pečlivěji testovat na větším množství textů. V této části se bude muset projevit velká kreativita v tvorbě nových druhů atributů. Také bude rozumné převést ohodnocení atributů na reálná čísla, čímž získáme například četnosti daných atributů.

Nakonec jsme chtěli porovnat výsledky z našeho experimentu s výsledky dalších do té doby nezkoumaných dat. Tento pokus jsme provedli na nové množině textů vydaných za dalšího půl roku a z výsledků jsme si ověřili, že náš algoritmus alespoň do určité míry funguje.

## 11.2 Vyhlídky do budoucna

Možných zlepšení a rozšíření do budoucnosti je mnoho (vyjmenováno včetně výše zmíněných):

- Automatizace procesu predikce – Vytvoření grafického uživatelského prostředí a spojení komponent do jednoho projektu.
- Vylepšování atributů – Jak hledání nových kombinací tokenů, tak také implementace možných druhů atributů (např. sekvence tokenů).
- Převedení ohodnocení atributů na reálná čísla.
- Integrace slovníku synonym do projektu – Pro generování (rozpoznávání) synonymních atributů.
- Porovnání úspěšností TIMEX2 systémů a vybrání toho nejlepšího.
- Testování různých délek predikce a odpovídající úprava atributů.

- Manuální evaluace většího množství textů pro filtraci.
- Vývoj systému pro snadné stahování a parsování textů z Internetu.
- Získání velkého množství dat a jejich zdrojů a následné testy s těmito daty.
- Vývoj systému pro opravu chyb ve zdrojových dokumentech – Při automatické anotaci jsme objevili mnoho chyb ve formátování, které jsme museli řešit převážně ručně.

Naše dosavadní výsledky nejsou příliš povzbudivé, ale když přihlédneme k faktu, že máme ohromný prostor pro zlepšení jednotlivých částí samotné procedury predikce, určitě stojí za to pokračovat dále. Ačkoli naše metoda nikdy nebude nosnou pro predikci (už samotné texty se mohou mýlit), může jednou dotvářet systém složený z více algoritmů a zohledňovat pozorování, která jiným algoritmům unikají.

# Seznam použité literatury


[1] Wang, Shouyang L; Yu, Lean; Lai, K. K. *CRUDE OIL PRICE FORECASTING WITH TEI@I METHODOLOGY*[online]. Journal of Systems Science and Complexity, 2005, 18(2): 145-166. K 28. únoru 2012 dostupné na WWW: `http://madis1.iss.ac.cn/madis.files/pub-papers/2005/252wsy.pdf`.

[2] Kulkarni, Siddhivinayak; Haidar, Imad. *Forecasting Model for Crude Oil Price Using Artificial Neural Networks and Commodity Futures Prices*[online]. International Journal of Computer Science and Information Security, Vol.2, No.1, June 2009. K 17. dubnu 2012 dostupné na WWW: `http://arxiv.org/ftp/arxiv/papers/0906/0906.4838.pdf`.

[3] Xie, Wen; Yu, Lean; Xu, Shanying; Wang, Shouyang. *A New Method for Crude Oil Price Forecasting Based on Support Vector Machines*[online]. Institute of Systems Science, Academy of Mathematics and Systems Sciences, Chinese Academy of Sciences, 2006. K 17. dubnu 2012 dostupné na WWW: `http://madis1.iss.ac.cn/madis.files/pub-papers/2006/aa/6.pdf`.

[4] Nigam, Kamal; Lafferty, John; McCallum, Andrew. *Using Maximum Entropy for Text Classification*[online]. In IJCAI-99 Workshop on Machine Learning for Information Filtering. School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213. 1999. K 28. únoru 2012 dostupné na WWW: `http://www.kamalnigam.com/papers/maxent-ijcaiws99.pdf`.

[5] Charniak, Eugene. *A Maximum-Entropy-Inspired Parser*[online]. Proceedings of NAACL-2000. Department of Computer Science, Brown University. August 3, 2010. K 28. únoru 2012 dostupné na WWW: `http://www.ldc.upenn.edu/Catalog/docs/LDC2000T43/parser.pdf`.

[6] Ratnaparkhi, Adwait. *A Simple Introduction to Maximum Entropy Models for Natural Language Processing*[online]. Institute for Research in Cognitive Science. University of Pennsylvania, Philadelphia. May 1997. K 28. únoru 2012 dostupné na WWW: `http://repository.upenn.edu/cgi/viewcontent.cgi?article=1083&context=ircs_reports`.

[7] Berger, Adam L; Della Pietra, Stephen A.; Della Pietra, Vincent J. *A maximum entropy approach to natural language processing*[online]. Computational Linguistics, 1996, 22(1):39–71. K 28. únoru 2012 dostupné na WWW: `http://acl.ldc.upenn.edu/J/J96/J96-1002.pdf`.

[8] Mikulová, Marie; Bémová, Alevtina; Hajic, Jan; Hajicová, Eva; Havelka, Jirí; Kolárová, Veronika; Kucová, Lucie; Lopatková, Markéta; Pajas, Petr; Panevová, Jarmila; Ševcíková, Magda; Sgall, Petr; Štepánek, Jan; Urešová, Zdenka; Veselá, Katerina; Žabokrtský, Zdenek. *Anotace na tektogramatické rovině Pražského závislostního korpusu. Referenční příručka*[online]. Technical report no. 2006/31. ÚFAL MFF UK, Praha. December 2006. K 28. únoru 2012 dostupné na WWW: `http://ufal.mff.cuni.cz/pdt2.0update/doc/tr-ref-cz-cz.pdf`.

[9] Žabokrtský, Zdeněk; Bojar, Ondřej. *TectoMT, Developer's Guide*[online]. ÚFAL MFF UK, Praha. December 2008. K 28. únoru 2012 dostupné na WWW: `http://ufal.mff.cuni.cz/~bojar/publications/2008-FILE-zabokrtsky_tectomt_2008-2008-tectomt-techrep.pdf`.

[10] Bojar, Ondřej; Žabokrtský, Zdeněk. *CzEng 0.9: Large Parallel Treebank with Rich Annotation*[online]. The Prague Bulletin of Mathematical Linguistics. November 2009. K 28. únoru 2012 dostupné na WWW: `http://ufal.mff.cuni.cz/~bojar/publications/2009-FILE-czeng_pbml_2009-2009-pbml-czeng.pdf`.

[11] Tamchyna, Aleš; Bojar, Ondřej. *Bohatá anotace ve frázovém strojovém překladu*[online]. In ITAT 2010 Informacné technológie - Aplikácie a Teória, Zborník príspevkov prezentovaných na konferencii ITAT. September 2010. K 28. únoru 2012 dostupné na WWW: `http://ufal.mff.cuni.cz/~bojar/publications/2010-FILE-tamchyna_bojar_2010-CAMERA_READY.pdf`.

[12] Ahn, David; van Rantwijk, Joris; de Rijke, Maarten. *A Cascaded Machine Learning Approach to Interpreting Temporal Expressions*[online]. Proceedings NAACL-HLT 2007. April 2007. K 22. březnu 2012 dostupné na WWW: `http://www.science.uva.nl/~mdr/Publications/Files/hlt-naacl-2007-timex.pdf`.

[13] Ferro, Lisa; Gerber, Laurie; Mani, Inderjeet; Sundheim, Beth; Wilson, George. *2005 Standard for the Annotation of Temporal Expressions*[online]. MITRE, TIDES. September 2005. K 28. únoru 2012 dostupné na WWW: `http://lang.snu.ac.kr/clab/sites/default/files/2005_timex2_standard_v1.1.pdf`.

[14] Manning, Christopher D.; Schütze, Hinrich. *Foundations of Statistical Natural Language Processing*[online] Cambridge, Mass.: MIT Press. 1999.

[15] Koehn, Philipp. *Statistical Significance Tests for Machine Translation Evaluation*[online]. Computer Science and Artificial Intelligence Laboratory,

Cambridge. EMNLP 2004. K 20. květnu 2012 dostupné na WWW: http://homepages.inf.ed.ac.uk/pkoehn/publications/bootstrap2004.pdf.

# Seznam tabulek


9.1 Souhrn rozložení textů . . . . . 23
9.2 Výsledky 1. fáze . . . . . 25
9.3 Výsledky 2. fáze . . . . . 26
9.4 Rozložení článků mezi výsledky . . . . . 27
9.5 Reálné rozložení výsledků, tj. v kolika týdnech cena klesala, stagnovala a stoupala . . . . . 27
9.6 Výsledky 3. fáze . . . . . 28
9.7 Výsledky 4. fáze . . . . . 28
9.8 Výsledky 1. konfigurace . . . . . 28
9.9 Výsledky 2. konfigurace . . . . . 29
9.10 Výsledky 3. konfigurace . . . . . 29
9.11 Výsledky 4. konfigurace . . . . . 29
9.12 Výsledky 5. konfigurace . . . . . 30
9.13 Výsledky 6. konfigurace . . . . . 30
9.14 Průměrné výsledky konfigurace se všemi atributy (1. konfigurace) 31
9.15 Jiné počítání baseline a úspěšnosti pro konfiguraci se všemi atributy (1. konfigurace) . . . . . 32

10.1 Vyhodnocení s učením na nových textech . . . . . 34
10.2 Vyhodnocení s učením na starých textech . . . . . 34

C.1 Cesty k důležitým souborům a složkám v rámci CD . . . . . 67

# Seznam obrázků

8.1 Rozdělení období při jednom z cyklů evaluace . . . . . 21

9.1 Počty textů z období pro předpověď vývoje ceny ropy na daný týden . . . . . 24
9.2 Rozložení počtů textů k predikci po týdnech . . . . . 24
9.3 Rozložení četnosti počtů textů k predikci (zanedbána krajní období bez textů) . . . . . 25

A.1 Nástroj pro manuální ohodnocení textů pro filtraci . . . . . 49

B.1 Diagram aktivit . . . . . 52

# Seznam použitých zkratek

| | |
|---|---|
| Brent | Označení pro typ ropy zahrnující 15 druhů ropy z nalezišť v Severním moři. |
| HTML | HyperText Markup Language je značkovací jazyk pro hypertext. |
| LIBSVM | Knihovna pro klasifikaci pomocí metody Support vector machines (viz SVM). |
| MaxEnt | Maximum entropy classifier je klasifikátor fungující na principu maximální entropie. |
| PML | Prague Markup Language je XML formát pro ukládání souborů textů anotovaných ve více rovinách. |
| RSS | Rodina XML formátů určených pro čtení novinek na webových stránkách. |
| SVM | Skupina metod strojového učení, které mohou být použity pro klasifikaci a regresi. |
| Treex | Modulární software pro zpracování přirozeného jazyka implementovaný pro Linux. |
| TIMEX2 | Standardní schéma pro detekci časových výrazů v textu a jejich následnou normalizaci. |
| TimexTag | Modulární systém pro detekci a normalizaci časových výrazů v anglickém textu splňující standard TIMEX2. |
| WTI | Ropa západotexaská (West Texas Intermediate) je typ ropy používaný pro oceňování cen ropy a derivátů z ní odvozených na newyorské burze. |
| XML | Extensible Markup Language je obecný značkovací jazyk, který umožňuje snadné vytváření konkrétních značkovacích jazyků. Používá se pro serializaci dat. |
| SGML | Standard Generalized Markup Language je univerzální značkovací metajazyk, který umožňuje definovat značkovací jazyky jako své vlastní podmnožiny. |

# Přílohy

# A. Uživatelská dokumentace

## A.1 Úvod

COPF (Crude oil price forecast) je systém pro odhad vývoje ceny ropy na základě textových zpravodajských informací. Poskytuje také nástroje pro vylepšení tohoto odhadu.

COPF je určen pro platformu Linux (testováno na distribucích Debian a Ubuntu).

## A.2 Instalace

Nejdříve je nutné zkopírovat program z CD (cesta k adresáři v rámci CD je uvedena v tabulce C.1) do počítače, protože potřebuje zapisovat do svých složek.

Pro fungování systému je nutné nainstalovat několik externích komponent. Kromě JAVA SDK je potřeba v terminálu provést tyto příkazy:

Nejdříve je nutné nastavit do proměnné COMP_PATH cestu k adresáři components (cesta k adresáři components v rámci CD je uvedena v tabulce C.1). Tedy například:

```
COMP_PATH="./components/"
ABS_COMP_PATH=`cd $COMP_PATH; pwd`
```

Dále potřebujeme nainstalovat nástroje pro kompilaci:

```
sudo apt-get install g++
```

Poté už nainstalujeme první komponentu, Charniak Parser (zde jsme při stažení z Internetu měli problémy, museli jsme přepisovat některé hlavičky a trochu poopravit Makefile):

```
cd ${ABS_COMP_PATH}TIMEX2_system/Charniak/reranking-parser/
export GCCFLAGS="-march=native"
make clean
make
```

Dále potřebujeme knihovnu libsvm (je potřeba verze 2.81, novější verze mají jiné rozhraní), jež je napsána v Pythonu, který proto musíme také nainstalovat (testováno na verzích 2.7 a 2.6).

```
sudo apt-get install python2.7-dev
```

Pokud jsme nainstalovali jinou verzi Pythonu než 2.7, je potřeba ještě změnit cestu v Makefile pro Python. Např. pro verzi 2.6:

```
cd ${ABS_COMP_PATH}TIMEX2_system/libsvm-2.81/python/
sed -i 's/python[0-9]\.[0-9]*/python2.6/' Makefile
```

V tuto chvíli nám už nic neschází k nainstalování libsvm:

```
cd ${ABS_COMP_PATH}TIMEX2_system/libsvm-2.81/
make clean
make all
cd python/
make clean
make all
```

Nakonec ještě musíme přidat potřebná práva pro spuštění:

```
cd ${ABS_COMP_PATH}TIMEX2_system/
chmod +x find_all_timexes.sh
chmod +x find_timexes.sh
chmod +x ./timextag-0.3/deptools/charniak-dep-parser
chmod +x ./timextag-0.3/deptools/convert/shuffle-deps.pl
chmod +x ./timextag-0.3/deptools/convert/wsj2dep.pl
```

## A.3 Příprava

Abychom mohli hodnotit cenu ropy, musíme získat vstupní data pro náš program. Nejdříve si nastavíme potřebné cesty k budoucím datům, což se dělá v souboru path.txt. Zde jsou tři cesty, první ke kořeni celého projektu, druhá k adresáři s daty a třetí do složky components k adresáři časových určení získaných pomocí TimexTagu. Například:

```
projectRootPath=./
downloadedTextsPath=./data/
timextagSgmlTextsPath=./components/TIMEX2_system/Texts/
```

Dále v konsoli nastavíme cesty ke spouštěcím souborům. Tedy pokud jsme v kořenovém adresáři projektu:

```
JARDIR_PATH="."
ABS_JARDIR_PATH=`cd $JARDIR_PATH; pwd`

FINDTIM_PATH="./components/TIMEX2_system/"
ABS_FINDTIM_PATH=`cd $FINDTIM_PATH; pwd`
```

Poté stáhneme články a historii cen ropy z Internetu. Zde si můžeme vybrat od jakého data chceme stahovat, a to přidáním atributu podle vzoru yyyy-MM-dd. Články program stáhne pouze chybějící. Stahování článků je náročné na čas a připojení k Internetu.

```
cd $ABS_JARDIR_PATH
java -ea -classpath COPF.jar copf.downloaders.pagedownloaders.
   DownloadAllPages [yyyy-MM-dd]
java -ea -classpath COPF.jar copf.downloaders.pricedownloaders.
   DownloadAllPrices [yyyy-MM-dd]
```

Články z Internetu jsou uloženy v XML souboru. Pro další potřeby je třeba provést konverzi (co článek, to soubor):

```
java -ea -classpath COPF.jar copf.utilities.textsconverters.
   ConverterXml2Files
```

Na článcích je nutné provést automatickou anotaci pomocí komponenty Treex. Tuto anotaci jsme prováděli na katedře ÚFAL a komponentu Treex k projektu nepřikládáme. Z ÚFALu jsme dostali články zabalené v gz formátu, proto je musíme rozbalit:

```
java -ea -classpath COPF.jar copf.utilities.textsconverters.
   ConverterGz2Txt
```

Texty je potřeba ještě časově ohodnotit. Pro toto ohodnocení musíme vytvořit soubory v SGML formátu. Po ohodnocení je nutné toto časové ohodnocení zapsat zpět do XML. Časové ohodnocování je časově nejnáročnější část predikce.

```
java -ea -classpath COPF.jar copf.utilities.textsconverters.
   ConverterXml2Sgml
cd $FINDTIM_PATH
./find_all_timexes.sh
cd $ABS_JARDIR_PATH
java -ea -classpath COPF.jar copf.utilities.textsconverters.
   ConverterTimexXml2DateRange
```

V tuto chvíli musíme provést filtraci textů. Abychom ji mohli provést, musíme mít množinu manuálně ohodnocených textů nebo potřebný model pro klasifikátor sloužící k filtraci. Manuální evaluaci pro filtraci textů provádíme pomocí speciálního nástroje (viz A.5.1).

Takto spustíme samotnou filtraci:

```
java -ea -classpath COPF.jar copf.FilterTexts [-model {cesta k
   modelu pro filtraci}]
```

Pro druhou fázi evaluace je potřeba vytvořit model (pokud máme model již vytvořený, můžeme tento příkaz vynechat):

```
java -ea -classpath COPF.jar copf.CreateModelFromTexts
```

### A.3.1 Přiložená data na CD

Na přiloženém CD jsou data, která jsme používali pro experiment (složka DataForExperiment, cesta v rámci CD viz tabulka C.1) a pro konečnou evaluaci (složka LastEvalData, cesta v rámci CD viz tabulka C.1). Z důvodu velikosti je na CD v rámci těchto dat jen to, co je potřebné pro predikci a evaluaci. Texty anotované pomocí komponenty Treex jsme ponechali pouze zabalené. Pro použití

těchto dat k predikci a evaluaci je tedy nutné rozbalit anotované texty (viz krok A.3), ostatní kroky přípravy dat můžeme vynechat.

Dále jsme na CD přiložili model pro filtraci a model pro predikci (evaluaci). Cesty v rámci CD viz tabulka C.1.

## A.4 Použití

Samotnou predikci ceny ropy pro určité datum provedeme následujícím příkazem s argumentem predikovaného data. Pokud nezadáme datum, algoritmus predikuje pro aktuální datum. Pokud bychom měli jiný model, podle kterého bychom chtěli predikovat, zadáme cestu k souboru, ve kterém je uložen (první argument), a model bude překopírován na správné místo a použije se při predikci. Momentálně je algoritmus nastaven tak, aby predikoval týden dopředu z článků minulých třiceti dnů (protože byl na této konfiguraci evaluován). Toto musíme vzít v potaz při posuzování důvěryhodnosti predikce.

```
java -ea -classpath COPF.jar copf.Predict [-model {cesta k modelu}]
   [-date yyyy-MM-dd]
```

Výstupem predikce je jedna z odpovědí „up“, „stag“, „down“, parametry predikce (predikované datum, délka predikce) a počet článků, z kterých se predikovalo. Zde uvádíme příklad:

```
predicted date: 2011-08-01
forecast length: 7 days
text count: 127
prediction result: down
```

Kromě predikce můžeme také evaluovat zadané atributy (pokud nezadáme žádný argument). Pokud bychom chtěli evaluovat již vytvořený model, zadáme cestu k souboru, ve kterém je uložen (první argument), a model bude překopírován na správné místo a ohodnocen.

```
java -ea -classpath COPF.jar copf.featuremanagement.FeatureEvaluator
    [-model {cesta k modelu}]
```

Výstupem evaluace je tabulka s reálnými (v řádcích) a predikovanými (ve sloupcích) hodnotami, úspěšností a baseline (definováno v 9.2). Zde je ukázka tabulky:

```
          down      stag      up        no_value
down      27        0         90        252
stag      10        0         29        517
up        38        0         121       275
Succes: 0,4698
Baseline: 0,5048
```

## A.5 Nástroje

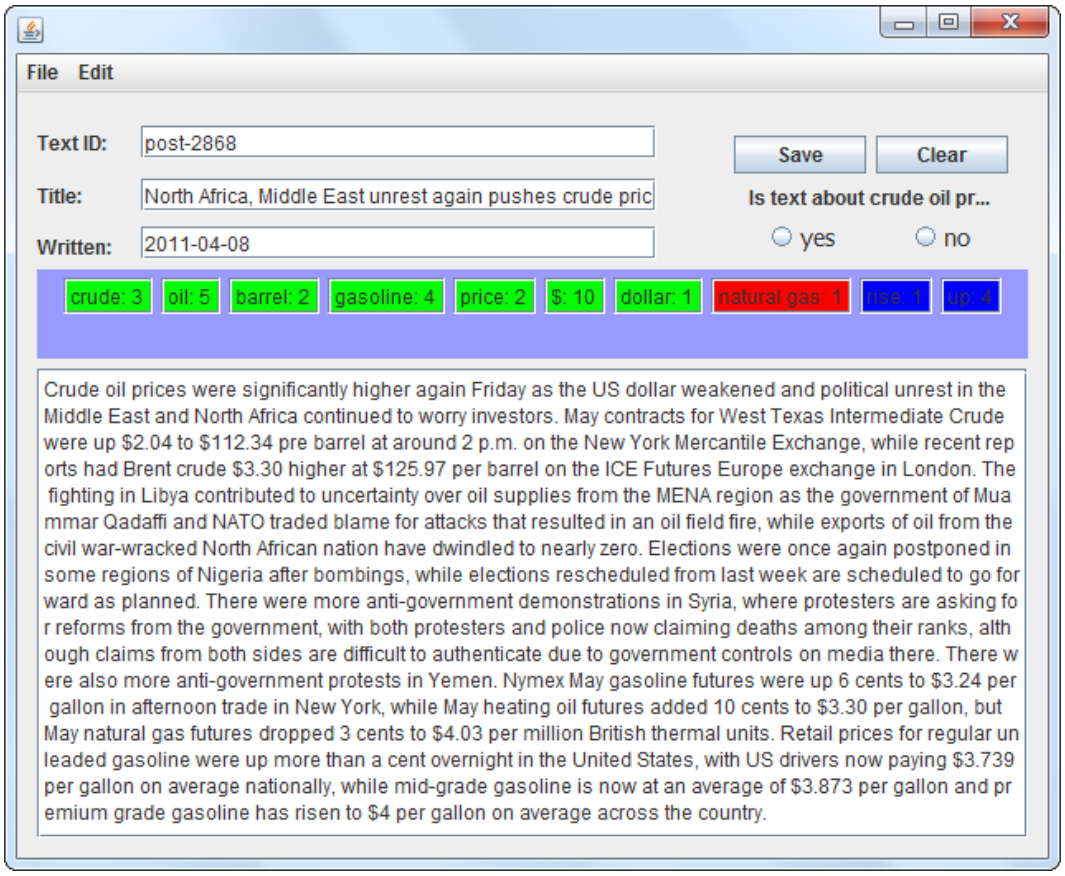
V rámci projektu COPF existuje několik užitečných nástrojů pro usnadnění práce.

### A.5.1 Nástroje pro filtraci

Nástroj pro manuální evaluaci textů spustíme tímto příkazem:

```
java -ea -classpath COPF.jar copf.utilities.manualevaluation.
    TextFilterManEvalWindow
```

Pro manuální evaluaci je nutné nahrát jeden z XML souborů, ve kterých jsou uloženy texty. Tento soubor nahrajeme na horní liště pomocí File > Open. Poté uvidíme např.:




Obrázek A.1: Nástroj pro manuální ohodnocení textů pro filtraci

Nahoře jsou tři pole s identifikátorem, titulkem a datem vytvoření textu. V dolní části je samotný text. Toto všechno můžeme editovat.

Uprostřed je modrý box, v kterém se objevují některá důležitá slova pro usnadnění evaluace. Zelená slova jsou ta, která ve většině případů indikují, že se text zabývá cenou ropy. Červená slova většinou indikují, že se text nezabývá cenou ropy, a modrá barva indikuje další pro nás zajímavá slova.

V pravé horní části vidíme tlačítka „yes“ a „no“. Těmito tlačítky určujeme, zda se článek zabývá cenou ropy. Tlačítkem „Clear“ vymažeme změny od posledního uložení a tlačítkem „Save“ uložíme všechny změny a náhodně se nám načte další neohodnocený článek (stejně jako po nahrání souboru). Pokud už žádný takový není, program to ohlásí.

### A.5.2 Nástroje pro vizualizaci

V rámci projektu umíme prezentovat některá data lepší cestou, než jen zkoumáním XML souborů.

Náš první nástroj ukáže graf vývoje ceny ropy:

```
java -ea -classpath COPF.jar copf.visualization.
   featureevalvisualization.PriceHistVis
```

Další nástroj ukáže výsledky poslední evaluace. Výstupem je tabulka s reálnými a predikovanými hodnotami (viz výstup evaluace A.4). Spustíme ho tímto příkazem:

```
java -ea -classpath COPF.jar copf.visualization.
   featureevalvisualization.ContingencyTableVis
```

Protože se naše výsledky mohou pro každé spuštění mírně lišit (kvůli náhodnému výběru článků ve fázi undersamplingu), vytvořili jsme další nástroj, který ukáže průměrný výsledek pro všechny provedené evaluace, které jsou uloženy v souboru „{cesta k datům}/Features/feature_evaluation.xml“. Výstupem je stejná tabulka jako v předchozím případě. Nástroj spustíme tímto příkazem:

```
java -ea -classpath COPF.jar copf.visualization.
   featureevalvisualization.AverageContTableVis
```

# B. Programátorská dokumentace

COPF (Crude oil price forecast) je systém pro odhad vývoje ceny ropy na základě textových zpravodajských informací. Poskytuje také nástroje pro vylepšení tohoto odhadu.

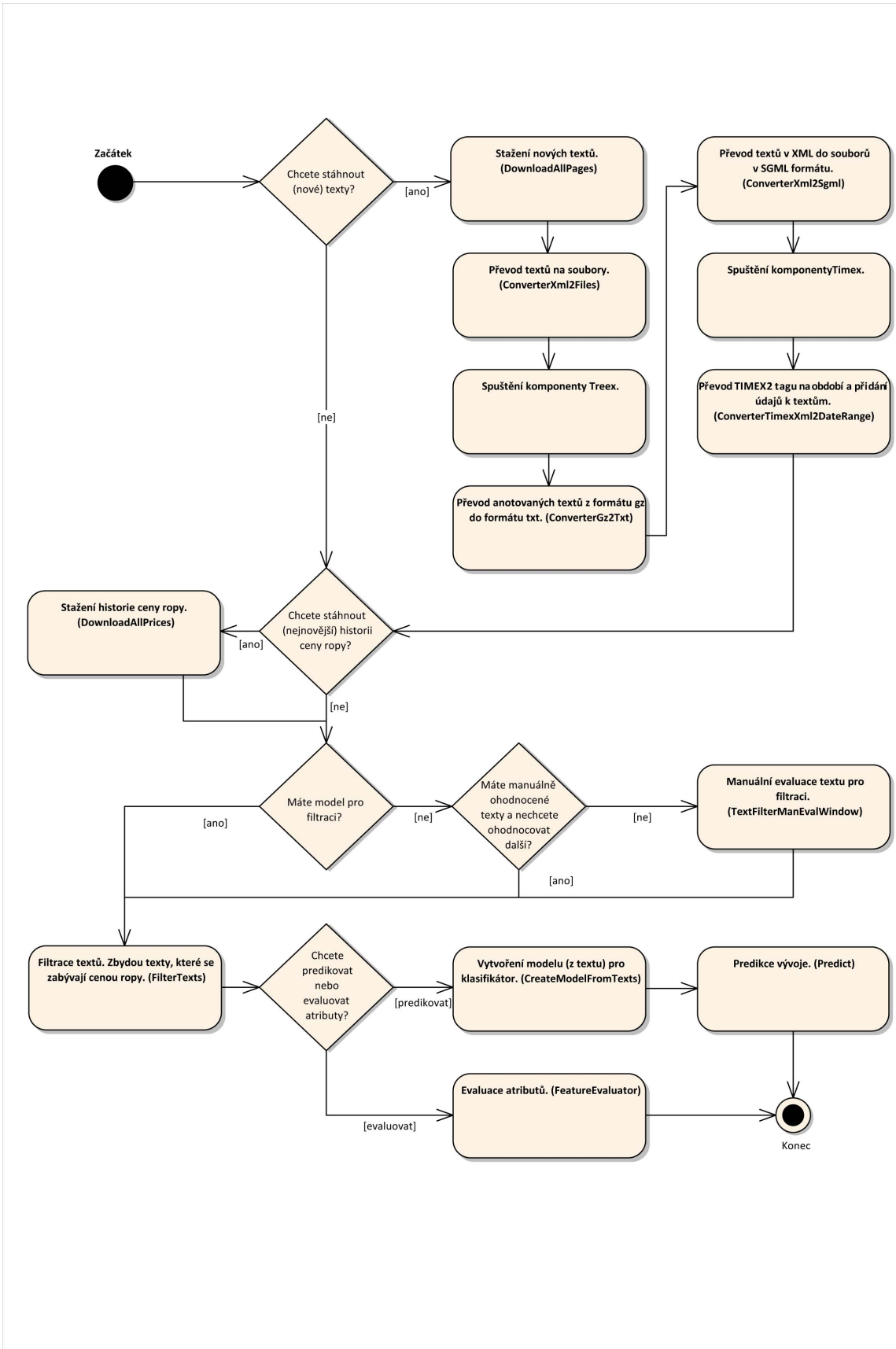
Systém COPF je napsaný v programovacím jazyce Java (JDK 1.7) ve vývojovém prostředí Netbeans 7.1. Externí komponenty jsou napsány též v jazycích Perl a Python. COPF je programován pro operační systém Linux.

## B.1 Architektura systému

Celý systém je v momentálním stavu uspořádán spíše jako soubor skriptů, které je nutné volat ve správném pořadí. Důvodů pro toto uspořádání je více: v první řadě experimentální povaha projektu, kdy se struktura použití ještě často obměňuje, dále pak časté použití (nahraditelných) externích komponent v jiných programovacích jazycích, včetně komponenty Treex spouštěné na zcela jiných počítačích katedry, a též vysoká časová náročnost některých kroků v rámci celé predikce či evaluace.

### B.1.1 Diagram aktivit

Pro přehled toho, co se provádí v rámci evaluace a predikce, jsme vytvořili diagram aktivit (viz B.1).




Obrázek B.1: Diagram aktivit

## B.2 Rozšíření systému

Systém je připraven na dva druhy rozšíření. Prvním typem je rozšíření funkcí systému a tím druhým je přidání zdrojů dat.

### B.2.1 Rozšíření funkčnosti

Do tohoto typu rozšíření spadá přidání nových typů atributů pro predikci a přidání vizualizací dat.

#### Nové typy atributů pro predikci

Přidání nového typu atributů pro predikci docílíme poděděním od abstraktní třídy FeatureType v balíčku copf.featuremanagement a přidáním nové položky do proměnné featTemplTypes ve třídě FeatureManager v témže balíčku. V rámci tohoto přidání definujeme řetězec, který bude reprezentovat nový typ v souborech s atributy.

#### Přidání nové vizualizace

Novou vizualizaci přidáme poděděním od abstraktní třídy Visualization v balíčku copf.visualization.featureevalvisualization. Vizualizace spouštíme jako samostatné skripty.

### B.2.2 Přidání zdrojů dat

V rámci projektu můžeme přidávat nové zdroje dat pro texty a historii ceny ropy.

#### Přidání nového zdroje textů

Pro přidání nového zdroje textů musíme podědit od abstraktní třídy DownloadPage v balíčku copf.downloaders.pagedownloaders a implementovat potřebné parsování zdroje za účelem získání pro nás důležitých informací. Mnoho potřebných funkcí můžeme nalézt ve třídě Downloader v balíčku copf.downloaders. Nakonec je nutné přidat volání metody download na této třídě do funkce main třídy DownloadAllPages v balíčku copf.downloaders.pagedownloaders.

#### Přidání nového zdroje historie ceny ropy

Pro přidání nového zdroje historie ceny ropy musíme podědit od abstraktní třídy DownloadPrices v balíčku copf.downloaders.pricedownloaders a implementovat potřebné parsování zdroje za účelem získání pro nás důležitých informací.

Mnoho potřebných funkcí můžeme nalézt ve třídě Downloader v balíčku copf.downloaders. Nakonec je nutné přidat volání metody download na této třídě do funkce main třídy DownloadAllPrices v balíčku copf.downloaders.pricedownloaders.

## B.3 Externí komponenty

Externí komponenty mají své vlastní dokumentace, proto u nich popíšeme pouze důležité části, které jsme používali.

### B.3.1 Timextag

V rámci komponenty TimexTag jsme používali tyto tři skripty, které najdeme v podadresáři ./TIMEX2_system/timextag-0.3/timextag/ adresáře components (cesta k adresáři components v rámci CD je uvedena v tabulce C.1).

Vstupem komponenty TimexTag jsou SGML soubory, každý tento soubor je třeba rozdělit na dvě části, soubor s XML tagy (prázdné jsou vynechány) a soubor s textovým obsahem původního SGML souboru zbavený XML tagů a jakýchkoli komentářů, procesních instrukcí atd. Toto rozdělení provádíme tímto příkazem:

```
python splitSgml.py \
      {cesta k souboru} \
      {cesta k výstupnímu XML souboru} \
      {cesta k výstupnímu textovému souboru}
```

Dále je potřeba udělat syntaktický rozbor textů. Pro toto se používá Charniakův parser, jehož výstupem je XML s rozborem. Toto se dělá příkazem:

```
python parseCharniak.py \
      {cesta ke složce, ve které jsou textové soubory vytvořené
         pomocí splitSgml.py} \
      {cesta ke složce, ve které jsou XML soubory vytvořené pomocí
         splitSgml.py} \
      {cesta ke složce, do které budou uloženy výstupní XML soubory}
```

Nakonec, pokud máme všechno připraveno, používáme pro určení časových údajů skript timextool.py. Tento skript provede časovou analýzu textu a vrátí XML, které je ve standardu TIMEX2.

```
python timextool.py \
      --inputtxt {cesta ke složce, ve které jsou textové soubory
         vytvořené pomocí splitSgml.py} \
      --inputxml {cesta ke složce, ve které jsou XML soubory
         vytvořené pomocí splitSgml.py} \
      --inputparses {cesta ke složce, ve které jsou soubory
         vytvořenému pomocí parseCharniak.py} \
      --all \
```

```
--outputxml {cesta ke složce, do které budou uloženy výstupní
    XML soubory}/
```

Více informací nalezneme v samotných souborech a v souborech README a Examples.txt, které najdeme v podadresáři ./TIMEX2_system/timextag-0.3/ adresáře components (cesta k adresáři components v rámci CD je uvedena v tabulce C.1).

### B.3.2 Treex

V rámci komponenty Treex jsme se zabývali pouze formátem vstupních a výstupních souborů, protože samotný rozbor pomocí této komponenty byl prováděn externě. Vstupem komponenty Treex byly soubory s texty, které jsme potřebovali ohodnotit (v každém souboru jeden text). Výstupem byly soubory zabalené ve formátu gz, opět co soubor, to jeden gz archiv. Více informací o komponentě je k nalezení v literatuře [8], [9], [10] a [11]).

## B.4 Externí knihovny

Kromě externích komponent jsme použili i další externí knihovny napsané v jazyce Java.

### B.4.1 OpenNLP MaxEnt

OpenNLP MaxEnt je knihovna implementující MaxEnt klasifikátor s příznivým uživatelským rozhraním. Knihovna obsahuje třídy pro snadné načítání dat, trénování modelu a samotnou klasifikaci. Zde je několik nejdůležitějších tříd:

- MaxentModel je rozhraní pro MaxEnt modely.
- AbstractModelWriter je abstraktní třída, která se stará o serializaci modelu.
- AbstractModelReader je abstraktní třída, která se stará o načtení modelu ze souboru.
- GISModel je implementace rozhraní MaxentModel, kterou používáme.

### B.4.2 JFreeChart

Tato knihovna slouží k vizualizaci dat pomocí grafů. Zatím ji používáme pouze na zobrazení vývoje ceny ropy. Problematické je, že dokumentace k této knihovně není volně ke stažení (samotná knihovna je zdarma, ale dokumentace je zpoplatněna). Protože jsme se k dokumentaci nedostali, zmíníme pouze jednu třídu:

- XYPlot je třída sloužící k vykreslení grafu funkce.

### B.4.3 HtmlCleaner

Knihovna HtmlCleaner je analyzér jazyka HTML. Protože je HTML na Internetu často špatně zformované a nepoužitelné pro další zpracování, je ho potřeba nejdříve správně zformovat, správně seřadit tagy atd. Právě k tomuto slouží HtmlCleaner. HtmlCleaner při čištění HTML kódu zároveň vytváří DOM strukturu, což je výhodné pro následné dotazování.

V této knihovně pro nás byly důležité tyto dvě třídy:

- TagNode je třída reprezentující jeden uzel DOM struktury.
- PrettyXmlSerializer je třída převádějící DOM strukturu opět na řetězec.

## B.5 Důležité datové struktury

V systému COPF existují kromě externích knihoven ještě dvě komponenty. Méně důležitou je knihovna, která se nazývá SkalCore a obsahuje pomocné třídy a jednoduché funkce, které nám chyběly ve standardních knihovnách jazyka Java. Dále jsme zde obalili některé třídy, často z důvodu ošetření výjimek, abychom zpřehlednili kód v důležitých třídách hlavního projektu. Tento projekt se nazývá COPF a obsahuje tyto důležité balíčky a třídy:

### B.5.1 copf

Balíček copf umožňuje samotnou predikci a důležité akce potřebné k jejímu provedení.

- CreateModel je třída pro tvorbu MaxEnt modelu z datového souboru.
- CreateModelFromTexts je třída pro tvorbu MaxEnt modelu z textů.
- FilterTexts je třída pro filtraci textů.
- Paths je třída, ve které jsou uloženy cesty v rámci projektu. Kromě souboru paths.txt, který může být v kořenovém adresáři projektu, zde můžeme měnit cesty k datům a komponentám.
- Predict je třída pro predikci z textů.
- Standards je třída, která obsahuje standardy v rámci projektu (např. formát data a času).

### B.5.2 copf.downloadedtexts

Balíček copf.downloadedtexts slouží k snadnějšímu přístupu ke staženým článkům uloženým v XML. Balíček obsahuje třídy vygenerované ze souboru s XML Schematu definujícím uložení textů v XML souborech. K těmto třídám je přidáno několik potřebných atributů pro zpracování.

### B.5.3 copf.downloaders

Balíček copf.downloaders obsahuje třídy pro stahování dat z Internetu.

- Downloader je třída obsahující potřebné funkce pro stahování textů.

### B.5.4 copf.downloaders.pagedownloaders

Balíček copf.downloaders.pagedownloaders obsahuje třídy pro stahování textů z Internetu.

- DownloadAllPages je třída, která slouží ke stahování textů ze všech zdrojů, pro které je vytvořen skript ke stáhnutí.
- DownloadPage je abstraktní třída, od které jsou poděděny všechny skripty pro stahování textů.

### B.5.5 copf.downloaders.pricedownloaders

Balíček copf.downloaders.pricedownloaders obsahuje třídy pro stahování historie ceny ropy z Internetu.

- DownloadAllPrices je třída, která slouží ke stahování historie ceny ropy ze všech zdrojů, pro které je vytvořen skript ke stáhnutí. Momentálně to je z jednoho zdroje.
- DownloadPrices je abstraktní třída, od které jsou poděděny všechny skripty pro stahování historie ceny ropy.

### B.5.6 copf.exportformatparser

Balíček copf.exportformatparser poskytuje snadnou práci s texty v Exportním formátu komponenty Treex.

- ExportFormatSentence je třída reprezentující větu z textu, který je výstupem komponenty Treex.
- ExportFormatText je třída reprezentující text, který je výstupem komponenty Treex.

### B.5.7 copf.featuremanagement

Balíček copf.featuremanagement slouží k práci se sadami atributů (nahrávání, hledání, ohodnocování atd.).

- FeatureEvaluator je třída pro ohodnocení konfigurací atributů. Je v ní implementována křížová evaluace.
- FeatureInvestigator je třída pro hledání atributů v textech.
- FeatureManager je třída sloužící k načítání atributů a hledání jejich výskytů v textech.
- FeatureType je abstraktní třída, od které jsou poděděny všechny typy atributů, které jsou vyhledávány v textech.
- PredictionResult je třída určená k uchovávání výsledků a informací o predikci.

### B.5.8 copf.pricehistorymanagement

Balíček copf.pricehistorymanagement slouží k práci s historií ceny ropy.

- OutcomeArbiter je třída sloužící k přiřazování hodnot z množiny výsledků změně ceny za nějaké období.
- PriceHistoryManager je třída pro přístup k historii cen ropy.

### B.5.9 copf.utilities.manualevaluation

Balíček copf.utilities.manualevaluation slouží k manuální evaluaci textů.

- TextFilterManEvalWindow je třída sloužící k manuální evaluaci textů pro filtraci.

### B.5.10 copf.utilities.textsconverters

Balíček copf.utilities.textsconverters slouží k převádění formátů dat v rámci projektu.

- ConverterGz2Txt je třída pro rozbalování textů zabalených do formátu gz.
- ConverterTimexXml2DateRange je třída převádějící XML výstup z komponenty TimexTag na rozmezí dvou dat.

- ConverterXml2Files je třída pro převod textů z XML do textových souborů (každý text v samostatném souboru).
- ConverterXml2Sgml je třída pro převod textů z XML do SGML souborů (každý text v samostatném souboru).

### B.5.11 copf.visualization.featureevalvisualization

Balíček copf.visualization.featureevalvisualization slouží k vizualizaci dat v rámci projektu.

- Visualization je abstraktní třída, od které dědí všechny vizualizace v rámci projektu.
- ContingencyTableVis je třída pro vizualizaci kontingenční tabulky pro evaluaci atributů.
- AverageContTableVis je třída pro vizualizaci kontingenční tabulky pro průměrné hodnoty ze všech evaluací atributů.
- PriceHistVis je třída pro vizualizaci vývoje ceny ropy v minulosti.

## B.6 Data

Pro COPF je velmi důležitá práce s daty. Formáty dat a jejich popis jsme rozčlenili podle toho, k čemu slouží. V rámci příkladů formátů pro lepší přehlednost používáme stále stejný text.

### B.6.1 TimexTag

TimexTag je externí komponenta popsaná výše (viz B.3.1). Vstupnem pro ni jsou SGML soubory. Tyto soubory jsou uloženy v podsložkách adresáře Texts (cesta v rámci CD je uvedena v tabulce C.1). Celý obsah soubory obaluje tag „DOC“ v němž jsou zanořeny tyto tagy:

- Tag „DATETIME“, který určuje referenční datum pro všechny časové údaje v textu.
- Tag „TITLE“, v kterém je uložen titulek textu.
- Tag „TEXT“, v kterém je uložen samotný text (bez titulku).

Zde je příklad jednoho takového souboru:

```
<DOC>
<DATETIME>31.01.2008</DATETIME>
<TITLE>Factoid on Oil Rigs</TITLE>
<TEXT>
In December of 2007, 2290 oil rigs were in operation in North
   America, 37 less than a year before.
</TEXT>
</DOC>
```

Výstupem komponenty je XML soubor ve standardu TIMEX2 (více viz [13]), který je uložený v podadresáři xmloutput adresáře, v kterém je vstupní SGML soubor. Zde je příklad jednoho souboru:

```
<?xml version="1.0" encoding="UTF-8"?>
<DOC generator="timexdoc.py">
<reftime rstart="1" rend="10">
</reftime>
<TEXT rstart="32" rend="131">
<TIMEX2 set="" rend="43" tmxclass="point" rstart="36" dirclass="
   before" parsenode=".1 p4" prenorm="|dex|Y|XXXX-12">December</
   TIMEX2>
<TIMEX2 set="" rend="51" val="2007" tmxclass="point" rstart="48"
   dirclass="before" parsenode=".1 p8" prenorm="|fq|_2007">2007</
   TIMEX2>
<TIMEX2 set="" rend="57" val="2290" tmxclass="point" rstart="54"
   dirclass="before" parsenode=".1 w5" prenorm="|fq|_2290">2290</
   TIMEX2>
</TEXT>
</DOC>
```

### B.6.2 Treex

Komponenta Treex potřebovala na vstupu textové soubory, které jsou uloženy v podsložkách složky DownloadedTextsInFiles (cesta v rámci CD je uvedena v tabulce C.1):

```
Factoid on Oil Rigs

In December of 2007, 2290 oil rigs were in operation in North
   America, 37 less than a year before.
```

Pro každý vstupní soubor byl výstupem soubor komprimovaný pomocí programu Gzip. Po rozbalení do odpovídající podsložky složky AnotatedTextsInFiles/txt (cesta v rámci CD je uvedena v tabulce C.1) soubor vypadal například takto (více viz [11]):

```
DownloadedTextsInFiles/oil_prize_net/FactoidonOilRigs20080131.parsed
   .treex.gz:s1  Factoid|factoid|NN|1|0|ExD on|on|IN|2|1|AuxP Oil|
```

```
   Oil|NNP|3|4|Atr Rigs|Rigs|NNP|4|2|Atr  factoid|PRED|1|0|complex|n
   :|n.denot|sg|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|0|-|- Oil|RSTR|2|3|
   complex|n:attr|n.denot|sg|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|0|-|-
   Rigs|LOC|3|1|complex|n:on+X|n.denot|sg
   |-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|0|-|-  0-0 2-1 3-2 1-2
DownloadedTextsInFiles/oil_prize_net/FactoidonOilRigs20080131.parsed
   .treex.gz:s2  In|in|IN|1|9|AuxP December|December|NNP|2|1|Adv of|
   of|IN|3|2|AuxP 2007|2007|CD|4|3|Atr ,|,|,|5|9|AuxX 2290|2290|CD
   |6|8|Atr oil|oil|NN|7|8|Atr rigs|rig|NNS|8|9|Sb were|be|VBD|9|0|
   Pred in|in|IN|10|9|AuxP operation|operation|NN|11|10|Adv in|in|IN
   |12|9|AuxP North|North|NNP|13|14|Atr America|America|NNP|14|12|
   Adv ,|,|,|15|14|AuxX 37|37|CD|16|14|Atr less|less|JJR|17|12|AuxP
   than|than|IN|18|17|AuxP a|a|DT|19|20|AuxA year|year|NN|20|17|NR
   before|before|IN|21|20|AuxC .|.|.|22|0|AuxK  December|TWHEN|1|6|
   complex|n:in+X|n.denot|sg|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|0|-|-
   2007|RSTR|2|1|complex|n:of+X|n.quant.def
   |-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|0|-|- 2290|RSTR|3|5|complex|n:
   attr|n.quant.def|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|0|-|- oil|RSTR
   |4|5|complex|n:attr|n.denot|sg
   |-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|0|-|- rig|ACT|5|6|complex|n:subj
   |n.denot|pl|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|0|-|- be|PRED|6|0|
   complex|v:fin|v|-|neg0|ant|ind|decl|-|-|-|-|disp0|-|it0|-|-|res0
   |-|-|1|-|- operation|LOC|7|6|complex|n:in+X|n.denot|sg
   |-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|0|-|- North|RSTR|8|9|complex|n:
   attr|n.denot|sg|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|0|-|- America|LOC
   |9|6|complex|n:in+X|n.denot|sg
   |-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|0|-|- 37|RSTR|10|9|complex|n:
   attr|n.quant.def|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|0|-|- year|CPR
   |11|9|complex|n:less_than_before+X|n.denot|sg
   |-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|0|-|- 1-0 3-1 5-2 6-3 7-4 8-5
   10-6 12-7 13-8 15-9 19-10 0-0 2-1 4-5 9-6 11-8 14-8 16-10 17-10
   18-10 20-10
```

### B.6.3 Ceny ropy

Historie ceny ropy je ukládána v prostém textovém souboru s názvem price_history_all.txt, který je ve složce DownloadedPriceHistory (cesta v rámci CD je uvedena v tabulce C.1). Jedna cena za určité období je uložena na samostatný řádek ve formátu: {datum}|{cena ropy za barel}. Datum je ve formátu podle normy ISO 8601 a cena je udávána v dolarech. Zde je krátký příklad:

```
1986-01-03|25.78
1986-01-10|25.99
1986-01-17|24.57
1986-01-24|20.31
1986-01-31|19.69
```

K ceně ropy se vztahuje i definice množiny našich výsledků. Tato množina se definuje v souboru available_outcomes.txt ve složce Features (cesta v rámci CD je uvedena v tabulce C.1). V souboru se definují názvy výsledků a jejich rozmezí.

Každý řádek má tvar „{název výsledku} < {horní hranice rozmezí}“. Jen na poslední řádce je pouze název výsledku, protože horní hranice rozmezí je nekonečno. Spodní hranice rozmezí daného výsledku je definována horní hranicí předchozího výsledku. První výsledek má spodní hranici nekonečno.

Výsledek se potom určuje podle toho, do kterého rozmezí padne tato hodnota: {cena v období, které předpovídáme} - {cena v období, z kterého předpovídáme}. Takto vypadá náš aktuální soubor:

```
down < -0.5
stag < 0.5
up
```

### B.6.4 Stahování textů

Texty jsou stahovány do XML souborů. Pro každý zdroj je vytvořena jedna podsložka s jedním XML souborem ve složce DownloadedTextsInXml (cesta v rámci CD je uvedena v tabulce C.1). Kořenovým tagem XML je tag „texts“, jenž obsahuje datum vytvoření, tag „created“ a texty, kde každý text je reprezentován tagem „text“. Ten obsahuje po stažení tyto tagy:

- Tag „id“, který představuje unikátní id textu.
- Tag „url“ obsahuje URL, ze kterého byl článek stažen.
- Tag „title“, v kterém je uložen titulek textu.
- Tag „date“, který určuje datum a čas vytvoření textu (formátováno podle standardu ISO 8601).
- Tag „content“, v kterém je uložen samotný text (bez titulku).

Po zjištění časových údajů pomocí komponenty TimexTag a filtraci přibudou pro článek tyto tagy:

- Tag „determined_date_from“, který určuje začátek období, kterým se zabývá (formátováno podle standardu ISO 8601).
- Tag „determined_date_to“, který určuje konec období, kterým se zabývá (formátováno podle standardu ISO 8601).

- Tag „filter_text“, který určuje, zda text prošel filtrem či nikoliv („yes“ znamená, že se zabývá vývojem ceny ropy, „no“ znamená, že se nezabývá vývojem ceny ropy).

Dále, pokud byl článek manuálně ohodnocen pro filtraci, přibude ještě tag „manualevaluation“, v kterém je vnořen ještě jeden tag „filter“ (hodnota „yes“ znamená, že se zabývá vývojem ceny ropy, „no“ znamená, že se nezabývá vývojem ceny ropy).

Zde je příklad XML souboru s jedním staženým článkem:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<texts>
  <text>
    <id>FactoidonOilRigs20080131</id>
    <url>http://www.oil-price.net/en/articles/oil-rigs.php</url>
    <title>Factoid on Oil Rigs</title>
    <date>2008-01-31</date>
    <content>In December of 2007, 2290 oil rigs were in operation in
        North America, 37 less than a year before.</content>
    <determined_date_from>2007-01-01</determined_date_from>
    <determined_date_to>2290-12-31</determined_date_to>
    <filter_text>no</filter_text>
  </text>
  <created>2011-07-28T21:27:31</created>
</texts>
```

```
Factoid on Oil Rigs

In December of 2007, 2290 oil rigs were in operation in North
   America, 37 less than a year before.
```

### B.6.5 Atributy

Konfigurace atributů pro filtraci i pro predikci jsou uloženy ve stejném formátu. Konfiguraci atributů pro filtraci najdeme ve složce Features/Filter v souboru features.txt. Konfiguraci atributů pro predikci najdeme ve složce Features v souboru features.txt (cesty v rámci CD jsou uvedeny v tabulce C.1).

Každá řádka souboru s atributy je prázdná nebo obsahuje komentář nebo obsahuje atribut. Komentář se pozná podle symbolu # na začátku řádky. Atribut se může skládat ze dvou částí, první je typ atributu a druhou tvoří parametry atributu. První tabulátor na řádce odděluje typ od parametrů. Typem atributu je například „set_text“ (množina slov ve větě). Parametry jsou od sebe odděleny mezerou. Jsou to tokeny, které hledáme v textu. Tokeny mohou být tří druhů: slovo v základním formátu, a-lemma, t-lemma (více viz 5). A-lemma (t-lemma) se

skládá z mnoha částí oddělených svislítkem. Pokud nechceme specifikovat všechny části, můžeme vynechat dané místo mezi dvěma svislítky.

Soubor s atributy může vypadat například takto:

```
# up
set_text  _T_gain|||||||||post||||||||||||||||
set_text  grow
set_sentence  barrel cent up

# down
set_text  decrease
set_text  fall

# stagnation
set_text  average
```

Připomínáme, že při tvorbě konfigurace atributů člověk neurčuje, jestli atribut indikuje vzrůst či pokles ceny. To se naučí program sám až podle skutečného vývoje ceny.

Výsledky evaluace atributů pro predikci se zapisují do XML souboru feature_evaluation.xml ve složce Features (cesty v rámci CD jsou uvedeny v tabulce C.1).

Kořenovým tagem souboru je tag „evaluation_history“, který obsahuje tagy „evaluation“. Tag „evaluation“ představuje jednu křížovou evaluaci atributů a obsahuje tyto tagy:

- Tag „evaluation_date“, který určuje datum a čas evaluace (formátováno podle standardu ISO 8601).
- Tag „forecast_length“, který představuje délku predikce ve dnech.
- Tag „tested_range“, který představuje šířku období (ve dnech) pro jednu predikci.
- Tag „tested_dates_per_eval“, který určuje počet týdnů predikovaných na jedno naučení klasifikátoru.
- Tag „results“, který obsahuje výsledky evaluace. Každý výsledek je reprezentován tagem „result“.

Tag „result“ obsahuje tyto tagy:

- Tag „article_count“, který představuje počet článků, z kterých se předpovídalo.

- Tag „prediction_date“, který určuje předpovídané datum (formátováno podle standardu ISO 8601).
- Tag „real_value“, který představuje výsledek podle reálného vývoje.
- Tag „outcomes“, který obsahuje všechny hodnoty z množiny řešení a jejich procentuální ohodnocení. Každá hodnota je reprezentována tagem „outcome“.
  - Tag „outcome“ má dva atributy. Atribut „name“ udává jméno hodnoty z množiny řešení a atribut „value“ udává předpovězenou pravděpodobnost pro danou hodnotu.

Zde uvádíme jako příklad výstup evaluace pro dva týdny (zbytek týdnů jsme smazali pro úsporu místa):

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<evaluation_history>
    <evaluation>
        <evaluation_date>2012-02-24T00:00:04</evaluation_date>
        <forecast_length>7</forecast_length>
        <tested_range>30</tested_range>
        <tested_dates_per_eval>10</tested_dates_per_eval>
        <results>
            <result>
                <article_count>12</article_count>
                <prediction_date>2011-08-26</prediction_date>
                <real_value>up</real_value>
                <outcomes>
                    <outcome name="down" value
                        ="0.3730728348183343"/>
                    <outcome name="up" value="0.508670087302741"/>
                    <outcome name="stag" value
                        ="0.1182570778789246"/>
                </outcomes>
            </result>
            <result>
                <article_count>2</article_count>
                <prediction_date>2011-09-02</prediction_date>
                <real_value>up</real_value>
                <outcomes>
                    <outcome name="down" value
                        ="0.35655478493486514"/>
                    <outcome name="up" value="0.5388129496896914"/>
                    <outcome name="stag" value
                        ="0.1046322653754435"/>
                </outcomes>
```

```
            </result>
        </results>
    </evaluation>
</evaluation_history>
```

### B.6.6 MaxEnt

V rámci predikce je nutné připravit pro klasifikátor vstupní data. Tato data se automaticky připravují z textů pomocí konfigurace atributů. Pro klasifikátor je potřeba soubor s daty pro natrénování a soubor s daty, podle kterých se bude predikovat.

Soubor pro natrénování se skládá z atributů všech textů, které jsme vybrali k natrénování klasifikátoru. Každý článek je v souboru reprezentován jednou řádkou, na které jsou atributy daného textu oddělené mezerou (každý atribut je nahrazen unikátním číselným identifikátorem). Za poslední mezerou je správná hodnota z množiny výsledků (počítáno z historie ceny ropy), tj. aktuálně jedna z hodnot „up“, „stag“, „down“. Viz příklad:

```
16=1.0 11=1.0  79=1.0  down
14=1.0  79=1.0  down
66=1.0 11=1.0  79=1.0  62=1.0  down
11=1.0  79=1.0  down
68=1.0 11=1.0  79=1.0  down
79=1.0  stag
37=1.0 47=1.0  79=1.0  down
11=1.0  79=1.0  down
66=1.0 37=1.0 8=1.0 11=1.0  79=1.0  62=1.0  up
79=1.0  up
```

Soubor, podle kterého klasifikátor predikuje, se liší od souboru pro natrénování pouze v řetězci za poslední mezerou na řádce. Za poslední mezerou v tomto souboru je znak „?“.

```
16=1.0 18=1.0  81=1.0  ?
50=1.0 11=1.0 47=1.0  81=1.0  ?
16=1.0 11=1.0  81=1.0  ?
17=1.0 50=1.0 19=1.0 47=1.0  81=1.0  ?
11=1.0 14=1.0  81=1.0  ?
```

# C. Obsah CD

Na CD přiloženém k bakalářské práci je samotná práce ve formátu PDF, COPF systém se všemi externími komponentami kromě komponenty Treex, zdrojové kódy aplikace, programátorská dokumentace, data a další. Zde jsou důležité cesty v rámci CD:

| cesta | obsah |
|---|---|
| `./BakalarskaPrace.pdf` | bakalářská práce (tento dokument) |
| `./Program/` | kořenová složka programu |
| `./Program/components/` | cesta k externím komponentám systému |
| `./Program/components/TIMEX2_system/` | cesta k TIMEX2 systému |
| `./Program/components/TIMEX2_system/find_all_timexes.sh` | cesta k skriptu spouštějícímu časové určení pro texty |
| `./Program/components/TIMEX2_system/Texts/` | cesta k SGML souborům, které jsou vstupem komponenty TimexTag |
| `./Program/java_src/` | zdrojové kódy napsané v jazyce Java |
| `./Program/javadoc/` | vygenerovaná dokumentace hlavního projektu v systému COPF |
| `./Program/lib/` | složka s potřebnými knihovnami |
| `./Program/COPF.jar` | spouštěcí soubor pro všechny skripty v projektu napsané v jazyku Java |
| `./Program/paths.txt` | soubor upravující cesty k složkám v rámci projektu |
| `./Program/data/DataForExperiment/` | data použitá při experimentu |
| `./Program/data/LastEvalData/` | data použitá při závěrečné evaluaci |
| `./Program/data/Models/filtering_model.txt` | model pro filtraci |
| `./Program/data/Models/prediction_model.txt` | model pro predikci |
| `./Program/data/NewData/` | složka pro nová data |
| Další cesty jsou v rámci jedné z datových složek DataForExperiment, LastEvalData, NewData: | |
| `./DownloadedTextsInFiles/` | složka s texty v souborech, které používá komponenta Treex |
| `./DownloadedTextsInXml/` | složka s texty v XML souborech, jejichž součástí jsou další doplňující údaje |
| `./AnotatedTextsInFiles/txt/` | složka s oanotovanými texty v souborech |
| `./DownloadedPriceHistory/` | složka s historií ceny ropy |
| `./Features/` | složka se soubory potřebnými pro atributy |
| `./Features/Filter/` | složka se soubory potřebnými pro atributy filtrace |

Tabulka C.1: Cesty k důležitým souborům a složkám v rámci CD

# D. Formáty souborů

## D.1 Příklad SGML formátu

```
<DOC>
<DATETIME>2012-02-24</DATETIME>
<TITLE>Gold futures lower in Asian trade</TITLE>
<TEXT>
Forexpros - Gold futures were lower in Asian trade on Friday. On the
    Comex division of the New York Mercantile Exchange, Gold futures
    for April delivery traded at USD1779.45 a troy ounce at time of
   writing falling 0.38%. It earlier traded at a session low USD1778
   .85 a troy ounce. Gold was likely to find support at USD1730.25
   and resistance at USD1788.85. US Dollar Index, which tracks the
   performance of the greenback versus a basket of six other major
   currencies, rose 0.01% to trade at USD78.69. Elsewhere on the
   Comex, Silver for March delivery fell 0.74% to trade at USD35.293
    a troy ounce while Copper for March delivery rose 0.12% to trade
    at USD3.822 a pound.
</TEXT>
</DOC>
```

## D.2 Příklad výstupu komponenty TimexTag

```
<?xml version="1.0" encoding="UTF-8"?>
<DOC generator="timexdoc.py">
<reftime rstart="1" rend="10" val="2011-12-12">
<TIMEX2 rstart="1" rend="10" val="2011-12-12">2011-12-12</TIMEX2>
</reftime>
<TEXT rstart="43" rend="1503">
<TIMEX2 set="" rend="84" val="2011-12-12" tmxclass="point" rstart
   ="79" dirclass="same" parsenode=".1 p16" prenorm="|dex|W|XXXX-WXX
   -1">Monday</TIMEX2>
<TIMEX2 set="" rend="334" val="P3M" anchor_dir="BEFORE" tmxclass="
   duration" rstart="323" anchor_val="2011-12" parsenode=".2 p48"
   prenorm="P3M">three months</TIMEX2>
<TIMEX2 set="" rend="804" val="2010" tmxclass="point" rstart="788"
   dirclass="before" parsenode=".6 p3" prenorm="|dex|Y|_" mod="
   APPROX">much of last year</TIMEX2>
<TIMEX2 set="" rend="939" val="PAST_REF" anchor_dir="BEFORE"
   tmxclass="genpoint" rstart="932" anchor_val="2011-12-12" dirclass
   ="before" parsenode=".7 p11" prenorm="PAST_REF">
recently</TIMEX2>
```

<TIMEX2 set="" rend="1028" val="P1M" anchor_dir="ENDING" tmxclass="
duration" rstart="1015" anchor_val="2011-12" parsenode=".8 p23"
prenorm="P1M">the past month</TIMEX2>
</TEXT>
</DOC>

# E. Sady atributů

## E.1 Sada atributů pro 1. fázi experimentu

```
# up
set_text  gain
set_sentence  price rise
set_sentence  price climb
set_sentence  price increase
set_sentence  price skyrocket
set_sentence  price boost
set_sentence  price push higher
set_sentence  price growth
set_sentence  price grow
set_sentence  price go up
set_sentence  price high
set_sentence  price exceed
set_sentence  oil expensive
set_sentence  trade high
set_sentence  barrel high $
set_sentence  barrel cent up
set_sentence  barrel cent high

# down
set_text  loss
set_sentence  price decrease
set_sentence  price fall
set_sentence  price descent
set_sentence  price drop
set_sentence  price low
set_sentence  price go down
set_sentence  price cheap
set_sentence  price plunge
set_sentence  price decline
set_sentence  trade low
set_sentence  barrel low $
set_sentence  barrel cent drop
set_sentence  barrel cent down
set_sentence  barrel cent low

# stagnation
set_sentence  price stagnate
set_sentence  price stagnation
set_sentence  price settle
```

```
set_sentence  price average
set_sentence  price comparable
set_sentence  price stable
set_sentence  price constant
set_sentence  price fixed
set_sentence  price change _T_#Neg|||||||||||||||||||||||||
set_sentence  price alter _T_#Neg|||||||||||||||||||||||||
set_sentence  cost equal
set_sentence  cost same
```

## E.2 Sada atributů pro 2. fázi experimentu

```
# up
set_text  gain
set_sentence  price rise
set_sentence  price raise
set_sentence  price climb
set_sentence  price increase
set_sentence  price skyrocket
set_sentence  price boost
set_sentence  price push higher
set_sentence  price growth
set_sentence  price grow
set_sentence  price go up
set_sentence  price high
set_sentence  price exceed
set_sentence  oil expensive
set_sentence  trade high
set_sentence  barrel up $
set_sentence  barrel high $
set_sentence  barrel cent up
set_sentence  barrel cent high

# down
set_text  loss
set_sentence  price decrease
set_sentence  price fall
set_sentence  price descent
set_sentence  price drop
set_sentence  price low
set_sentence  price go down
set_sentence  price cheap
set_sentence  price plunge
set_sentence  price decline
set_sentence  trade low
set_sentence  barrel drop $
```

```
set_sentence  barrel down $
set_sentence  barrel low $
set_sentence  barrel cent drop
set_sentence  barrel cent down
set_sentence  barrel cent low

# stagnation
set_sentence  price stagnate
set_sentence  price stagnation
set_sentence  price settle
set_sentence  price average
set_sentence  price comparable
set_sentence  price stable
set_sentence  price constant
set_sentence  price fixed
set_sentence  price change _T_#Neg|||||||||||||||||||||||||
set_sentence  price alter _T_#Neg|||||||||||||||||||||||||
set_sentence  cost equal
set_sentence  cost same
```

## E.3 Poslední sada atributů

```
# clever configuration with past and future tense
# and with feature describing source

source

# up
set_text  _T_gain|||||||||post|||||||||||||||||
set_sentence  price _T_rise|||||||||post|||||||||||||||||
set_sentence  price _T_raise|||||||||post|||||||||||||||||
set_sentence  price _T_climb|||||||||post|||||||||||||||||
set_sentence  price _T_increase|||||||||post|||||||||||||||||
set_sentence  price _T_skyrocket|||||||||post|||||||||||||||||
set_sentence  price _T_boost|||||||||post|||||||||||||||||
set_sentence  price _T_push|||||||||post||||||||||||||||| higher
set_sentence  price growth
set_sentence  price _T_grow|||||||||post|||||||||||||||||
set_sentence  price _T_go|||||||||post|||||||||||||||||| up
set_sentence  price high
set_sentence  price _T_exceed|||||||||post|||||||||||||||||
set_sentence  price above average
set_sentence  oil expensive
set_sentence  _T_trade|||||||||post|||||||||||||||||| high
set_sentence  barrel up $
set_sentence  barrel high $
```

```
set_sentence  barrel cent up
set_sentence  barrel cent high
## nr
set_text  _T_gain|||||||||nr||||||||||||||||
set_sentence  price _T_rise|||||||||nr||||||||||||||||
set_sentence  price _T_raise|||||||||nr||||||||||||||||
set_sentence  price _T_climb|||||||||nr||||||||||||||||
set_sentence  price _T_increase|||||||||nr||||||||||||||||
set_sentence  price _T_skyrocket|||||||||nr||||||||||||||||
set_sentence  price _T_boost|||||||||nr||||||||||||||||
set_sentence  price _T_push|||||||||nr||||||||||||||||| higher
set_sentence  price _T_grow|||||||||nr||||||||||||||||
set_sentence  price _T_go|||||||||nr||||||||||||||||| up
set_sentence  price _T_exceed|||||||||nr||||||||||||||||
set_sentence  _T_trade|||||||||nr||||||||||||||||| high

# down
set_text  _T_loss|||||||||post||||||||||||||||
set_sentence  price _T_decrease|||||||||post||||||||||||||||
set_sentence  price _T_fall|||||||||post||||||||||||||||
set_sentence  price _T_descent|||||||||post||||||||||||||||
set_sentence  price _T_drop|||||||||post||||||||||||||||
set_sentence  price low
set_sentence  price _T_go|||||||||post||||||||||||||||| down
set_sentence  price cheap
set_sentence  price _T_plunge|||||||||post||||||||||||||||
set_sentence  price _T_decline|||||||||post||||||||||||||||
set_sentence  price below estimate
set_sentence  price below normal
set_sentence  price discount
set_sentence  trade low
set_sentence  barrel _T_drop|||||||||post||||||||||||||||| $
set_sentence  barrel down $
set_sentence  barrel low $
set_sentence  barrel cent _T_drop|||||||||post||||||||||||||||
set_sentence  barrel cent down
set_sentence  barrel cent low
## nr
set_text  _T_loss|||||||||nr||||||||||||||||
set_sentence  price _T_decrease|||||||||nr||||||||||||||||
set_sentence  price _T_fall|||||||||nr||||||||||||||||
set_sentence  price _T_descent|||||||||nr||||||||||||||||
set_sentence  price _T_drop|||||||||nr||||||||||||||||
set_sentence  price _T_go|||||||||nr||||||||||||||||| down
set_sentence  price _T_plunge|||||||||nr||||||||||||||||
set_sentence  price _T_decline|||||||||nr||||||||||||||||
set_sentence  barrel _T_drop|||||||||nr||||||||||||||||| $
```

```
set_sentence  barrel cent _T_drop|||||||||nr||||||||||||||||

# stagnation
set_text  balance
set_sentence  price _T_stagnate|||||||||post||||||||||||||||
set_sentence  price stagnation
set_sentence  price _T_settle|||||||||post||||||||||||||||
set_sentence  price average
set_sentence  price comparable
set_sentence  price stable
set_sentence  price constant
set_sentence  price fixed
set_sentence  price _T_change|||||||||post|||||||||||||||||  _T_#Neg
    |||||||||||||||||||||||||
set_sentence  price _T_alter|||||||||post|||||||||||||||||  _T_#Neg
    |||||||||||||||||||||||||
set_sentence  cost equal
set_sentence  cost same
## nr
set_sentence  price _T_stagnate|||||||||nr||||||||||||||||
set_sentence  price _T_settle|||||||||nr||||||||||||||||
set_sentence  price _T_change|||||||||nr||||||||||||||||  _T_#Neg
    |||||||||||||||||||||||||
set_sentence  price _T_alter|||||||||nr||||||||||||||||  _T_#Neg
    |||||||||||||||||||||||||
```