# Analýza sentimentu recenzí filmů IMDB pomocí Stanza

**Kód předmětu:** 4IZ470

**Název předmětu:** Dolování znalostí z webu

**Autor:** Hoang Anh Tran Nguyenová

## Obsah

# Úvod

V rámci této semestrální práce jsem si jako nástroj sentimentální analýzy vybrala knihovnu Stanza, vyvinutou Stanfordovou univerzitou. Na tento nástroj jsem narazila při vyhledávání technik v oblasti zpracování přirozeného jazyka (NLP) na internetu. Stanza byla navržena tak, aby poskytovala robustní a přesné NLP nástroje pro více než 60 jazyků. Mezi hlavní vlastnosti knihovny Stanza patří modularita, jednoduchost použití, kvalita modelů, podpora více jazyků a integrace s dalšími nástroji.

Prozkoumáme dataset **imdb_movies.xlsx**, který obsahuje titul, žánry, rok a recenze filmů. Tento dataset jsem vytvořila v Excelu pomocí on-line databáze IMDB. Nejprve připravíme data a prostředí, načteme a předzpracujeme data, poté inicializujeme knihovnu Stanza a provedeme analýzu sentimentů na vzorku dat. Nakonec vyhodnotíme model a experimentujeme s parametry.

Experimentální část semestrální práce se zaměřuje na provedení a vyhodnocení sentimentální analýzy filmových recenzí z datasetu IMDB pomocí knihovny Stanza. Cílem této části je klasifikovat jednotlivé recenze jako pozitivní, neutrální nebo negativní.

Tato semestrální práce nemá textový ani věcný překryv, ani jinou věcnou souvislost, s jinými semestrálními nebo kvalifikačními pracemi, které jsem zpracovávala.

# Data set

Datová sada použitá v této práci byla vytvořena mnou pomocí stránky IMDB, kde jsem prohledávala jednotlivé filmy a přidávala atributy a hodnoty do Excel souboru. Jedná se o datovou sadu IMDB obsahující 12 filmů, přičemž ke každému filmu jsou 3 recenze, tedy dohromady 36 recenzí.

Popis sloupců jsou uvedeny níže:

- `movie_title`: název filmu.
- `genres`: žánry spojené s filmem oddělené středníkem.
- `year`: rok uvedení filmu.
- `reviews`: recenze spojené s filmem oddělené středníkem.

# Stanza

Stanza je NLP knihovna vyvinutá Stanfordovou univerzitou, která poskytuje nástroje pro zpracování přirozeného jazyka v mnoha jazycích. Obsahuje nástroje, které mohou být použity v pipeline k převodu řetězce obsahujícího text v lidském jazyce na seznamy vět a slov, k vygenerování základních tvarů těchto slov, jejich částí řeči a morfologických rysů, k syntaktickému rozboru závislostí struktury a k rozpoznávání pojmenovaných entit. Sada nástrojů je navržena tak, aby byla paralelní mezi více 70 jazyky a využívala formalismus univerzálních závislostí.

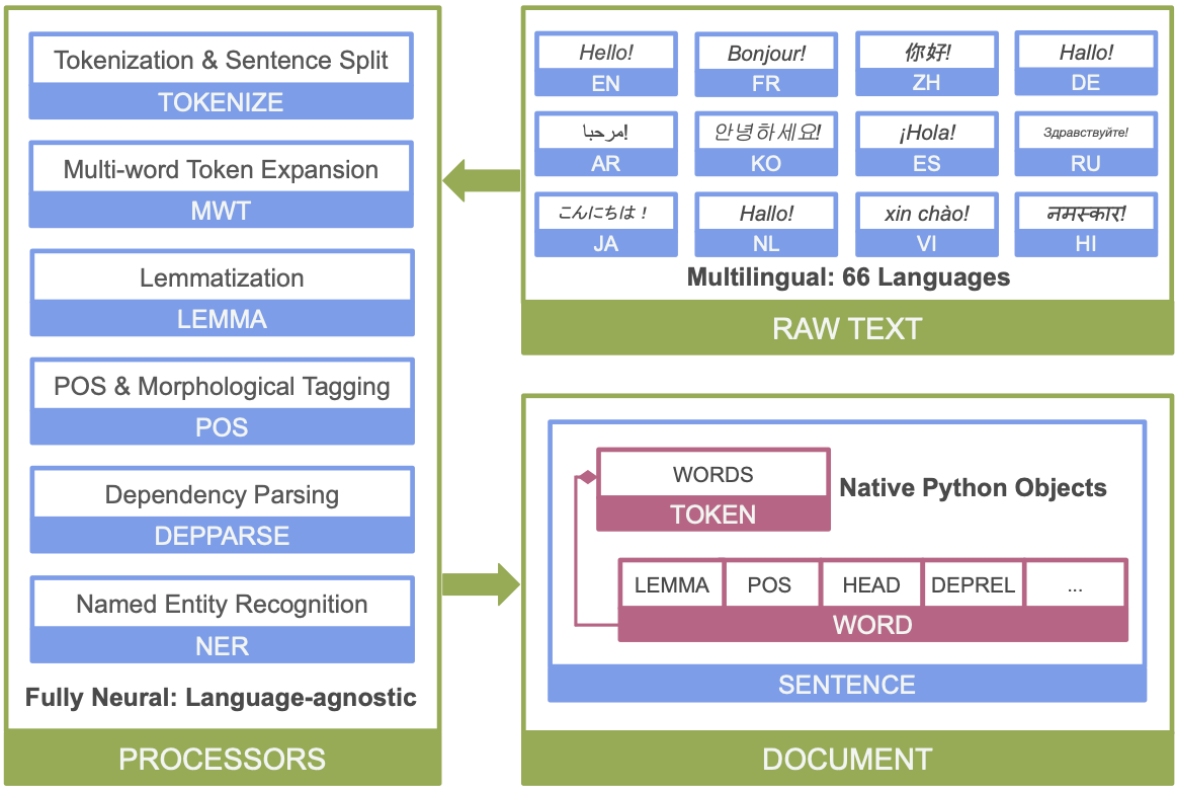
Stanza je postavena na vysoce přesných komponentách neuronových sítí, které rovněž umožňují efektivní trénování a vyhodnocování s vlastními anotovanými daty. Moduly jsou postaveny nad knihovnou PyTorch. Mnohem vyššího výkonu dosáhnete, pokud software spustíte na počítači s grafickým procesorem.

Stanza navíc obsahuje rozhraní jazyka Python k balíčku CoreNLP Java a dědí z něj další funkce, jako je například rozbor konstituentů, řešení koreferencí a porovnávání jazykových vzorů. (Pro použití základních nativních funkcí programu Stanza však není nutné mít balíček CoreNLP).

Stručně řečeno, Stanza nabízí:

- Kompletní neuronovou síť pro robustní analýzu textu, včetně tokenizace, expanze víceslovných tokenů (MWT), lemmatizace, označování částí řeči (POS) a morfologických rysů, rozboru závislostí a rozpoznávání pojmenovaných entit
- Předtrénované neuronové modely podporující 70 (lidských) jazyků
- Stabilní, oficiálně spravované rozhraní jazyka Python pro CoreNLP

Níže je uveden přehled pipeline neuronových sítí NLP nástroje Stanza:




*Obrázek 1 - Stanza pipeline*

## Tokenizace a segmentace vět

Tokenizaci a segmentaci vět ve Stanze provádí společně TokenizeProcessor. Tento procesor rozdělí nezpracovaný vstupní text na tokeny a věty, takže následná anotace může probíhat na úrovni vět. Tento procesor lze vyvolat příkazem Tokenize.

## Rozšíření víceslovných tokenů (MWT)

Rozšiřující modul Multi-Word Token (MWT) dokáže rozložit surový token na více syntaktických slov, což usnadňuje provádění analýzy univerzálních závislostí v některých jazycích. O to se stará MWTProcessor ve Stanze a lze jej vyvolat pod jménem mwt. Token, na kterém bude provedena expanze, předpovídá TokenizeProcessor ještě před vyvoláním MWTProcessoru.

## Lemmatizace

Modul lemmatizace obnovuje tvar lemmatu pro každé vstupní slovo. Například vstupní sekvence „snědl jsem jablko“ bude lemmatizována na „jíst jablko“. Tento typ normalizace slov je užitečný v mnoha reálných aplikacích. V systému Stanza provádí lemmatizaci nástroj LemmaProcessor, který lze vyvolat pomocí jména lemma.

## POS a morfologické funkce

Modul pro označování částí řeči (POS) a morfologických rysů označuje slova univerzálními značkami POS (UPOS), značkami POS specifickými pro jednotlivé stromy (XPOS) a univerzálními morfologickými rysy (UFeats). To společně provádí POSProcessor v systému Stanza a lze jej vyvolat pomocí jména pos.

## Analýza sentimentu

Stanza používá konvoluční neuronovou síť (CNN) k klasifikaci sentimentu. CNN je typ neuronové sítě vhodný pro zpracování dat s mřížkovou topologií, jako jsou obrazy nebo sekvence textu.

# Experiment

Jak bylo uvedeno v úvodu, cílem tohoto experimentu je klasifikovat jednotlivé recenze jako pozitivní, neutrální nebo negativní pomocí přístupů zpracování přirozeného jazyka (NLP) a strojového učení. Tento experiment má pět hlavních cílů. Prvním cílem je, že vytvoříme automatickou klasifikaci recenzí. Vytvoříme model, který dokáže automaticky přiřadit sentiment k jednotlivým recenzím filmů. To zahrnuje trénování modelu strojového učení na datech recenzí a následné hodnocení jeho výkonu. Dalším cílem je získat vyhodnocení přesnost a efektivitu modelu strojového učení pomocí standardních metrik, jako jsou přesnost, preciznost, odvolání (recall) a F1 skóre. Poté výsledky budou vizualizovány pomocí wordcloudů, které zobrazují nejčastěji se vyskytující slova v pozitivních, neutrálních a negativních recenzích.

Tento experiment umožňuje automatizovanou a efektivní analýzu uživatelských recenzí filmů, což může být užitečné pro filmová studia, marketingové analýzy nebo pro samotné uživatele, kteří se chtějí rychle zorientovat v obecném názoru na konkrétní film.

Více o postup python kódu lze vidět v HTML souboru **sentiment_analysis.html.**

## Import knihoven a načtení dat

Nejprve je nutné zajistit, aby byly nainstalovány potřebné knihovny: pandas, stanza, scikit-learn, matplotlib a wordcloud.

```
pip install pandas openpyxl stanza scikit-learn matplotlib wordcloud
```

Poté importujeme potřebné knihovny v Pythonu, načteme soubor s daty recenzí filmů a připravíme data pro analýzu

```
import pandas as pd
import stanza
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix
import matplotlib.pyplot as plt
from wordcloud import WordCloud

# Load the Excel file
file_path = './imdb_movies.xlsx'
df = pd.read_excel(file_path)

# Display the first few rows of the dataframe
df.head()
```

| | movie_title | genres | year | reviews |
|---|---|---|---|---|
| 0 | The Fall Guy | action;comedy;drama | 2024 | "A disappointing spectacle ."; "Absolute pure ... |
| 1 | Hellboy | action;adventure; fantasy | 2019 | "Terrible Writing, Weird Directing and Awful C... |
| 2 | Train to Busan | action;horror;thriller | 2016 | "Unforgettable experience! One of the best zom... |
| 3 | Mother of the Bride | comedy;drama;romance | 2024 | "Poor acting."; "Another rom com nightmare .";... |
| 4 | The Iron Claw | biography;drama;sport | 2023 | "To me, this was good, but not great."; "Wheth... |

*Obrázek 2 - výstup 1*

## Čištění a předzpracování dat

Data vyčistíme a odstraníme chybějící hodnoty, rozdělíme jednotlivé recenze oddělené středníkem do seznamů a poté je rozšíříme do samostatných řádků pro další analýzu.

```
# Drop rows with missing values
df.dropna(inplace=True)

# Function to split semicolon-separated reviews into individual entries
df['reviews'] = df['reviews'].apply(lambda x: [review.strip().strip('"')
for review in x.split(';')])

# Verify the structure after splitting
print("After splitting:", df['reviews'].head())

# Explode the reviews into separate rows
df_exploded = df.explode('reviews')

# Print the length of the reviews column array
print(f'Number of reviews: {len(df_exploded["reviews"])}')

# Display the first few rows after processing
df_exploded.head()
```

```
After splitting: 0    [A disappointing spectacle ., Absolute pure fu...
1    [Terrible Writing, Weird Directing and Awful C...
2    [Unforgettable experience! One of the best zom...
3    [Poor acting., Another rom com nightmare ., Li...
4    [To me, this was good, but not great., Whether...
Name: reviews, dtype: object
Number of reviews: 36
```

| | movie_title | genres | year | reviews |
|---|---|---|---|---|
| **0** | The Fall Guy | action;comedy;drama | 2024 | A disappointing spectacle . |
| **0** | The Fall Guy | action;comedy;drama | 2024 | Absolute pure fun, you won't be disappointed. |
| **0** | The Fall Guy | action;comedy;drama | 2024 | One Of The Most Entertaining Films Of The Year . |
| **1** | Hellboy | action;adventure; fantasy | 2019 | Terrible Writing, Weird Directing and Awful CGI . |
| **1** | Hellboy | action;adventure; fantasy | 2019 | Neither a complete disaster nor a triumph. |

*Obrázek 3 - výstup 2*

## Stažení a inicializace knihovny Stanza

Provedeme analýzu pomocí Stanza a posoudíme, jestli recenze je:

- Pozitivní = 2
- Neutrální = 0
- Negativní = 1

Používáme seznamy pozitivních a negativních slov k dodatečné úpravě sentimentu.

```
# Download the stanza model
stanza.download('en')
nlp = stanza.Pipeline('en',
processors='tokenize,mwt,pos,lemma,sentiment')

# Lists of positive and negative words
positive_words = set([
    'good', 'great', 'excellent', 'amazing', 'wonderful', 'best', 'love',
'fantastic', 'awesome', 'superb'
])
negative_words = set([
    'bad', 'terrible', 'awful', 'worst', 'poor', 'hate', 'disappointing',
'weird', 'waste', 'horrible'
])

# Function to preprocess text using stanza and assign sentiment
def preprocess_and_assign_sentiment(text):
    doc = nlp(text)
    processed_text = ' '.join([word.lemma for sent in doc.sentences for
word in sent.words if word.text.isalpha()])

    # Assign sentiment based on stanza sentiment analysis
    sentiment_scores = [sent.sentiment for sent in doc.sentences if
sent.sentiment is not None]
    sentiment_score = sum(sentiment_scores) if sentiment_scores else 0

    # Custom sentiment adjustment based on positive and negative words
    words = set(processed_text.split())
    positive_count = len(words & positive_words)
    negative_count = len(words & negative_words)

    if sentiment_score > 0 or positive_count > negative_count:
        sentiment = 2  # Positive
    elif sentiment_score < 0 or negative_count > positive_count:
        sentiment = 1  # Negative
    else:
        sentiment = 0  # Neutral

    return processed_text, sentiment

# Apply the function to create the processed_reviews and sentiment
columns
df_exploded['processed_reviews'], df_exploded['sentiment'] =
zip(*df_exploded['reviews'].apply(preprocess_and_assign_sentiment))

# Display the first few rows of the processed dataframe
df_exploded.head()
```

```
2024-06-01 00:09:59 INFO: Loading these models for language: en (English):
================================
| Processor | Package           |
--------------------------------
| tokenize  | combined          |
| mwt       | combined          |
| pos       | combined_charlm   |
| lemma     | combined_nocharlm |
| sentiment | sstplus_charlm    |
================================

2024-06-01 00:09:59 INFO: Using device: cpu
2024-06-01 00:09:59 INFO: Loading: tokenize
2024-06-01 00:09:59 INFO: Loading: mwt
2024-06-01 00:09:59 INFO: Loading: pos
2024-06-01 00:09:59 INFO: Loading: lemma
2024-06-01 00:09:59 INFO: Loading: sentiment
2024-06-01 00:09:59 INFO: Done loading processors!
```

*Obrázek 4 - výstup 3*

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | movie_title | genres | year | reviews | processed_revi | sentiment |
| 2 | The Fall Guy | action;comedy;d | 2,024 | A disappointing | a disappointing | 1 |
| 3 | The Fall Guy | action;comedy;d | 2,024 | Absolute pure fu | absolute pure fu | 2 |
| 4 | The Fall Guy | action;comedy;d | 2,024 | One Of The Mos | one of the most | 2 |
| 5 | Hellboy | action;adventure | 2,019 | Terrible Writing, | terrible write we | 1 |
| 6 | Hellboy | action;adventure | 2,019 | Neither a compl | neither a comple | 0 |
| 7 | Hellboy | action;adventure | 2,019 | Fun, entertaining | fun entertaining | 2 |
| 8 | Train to Busan | action;horror;thr | 2,016 | Unforgettable ex | unforgettable ex | 2 |
| 9 | Train to Busan | action;horror;thr | 2,016 | An overall excell | a overall excelle | 2 |
| 10 | Train to Busan | action;horror;thr | 2,016 | Starts off good. | start off good tu | 2 |
| 11 | Mother of the Bı | comedy;drama;r | 2,024 | Poor acting. | poor acting | 1 |
| 12 | Mother of the Bı | comedy;drama;r | 2,024 | Another rom co | another ROM co | 0 |

*Obrázek 5 - výstřižek souboru imbd_movies_sentiment_scores.xlsx*

Celou tabulku lze vidět v souboru **imbd_movies_sentiment_scores.xlsx**.

## Rozdělení dat a vektorizace TF-IDF

Kód připravuje textová data pro strojové učení tím, že je rozdělí na trénovací (80 %) a testovací (20 %) sady a transformuje textová data do numerické podoby pomocí TF-IDF.

```
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer

# Split the data into training and testing sets
X = df_exploded['processed_reviews']
y = df_exploded['sentiment']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Vectorize the text data using TF-IDF
vectorizer = TfidfVectorizer(max_features=5000)
X_train_tfidf = vectorizer.fit_transform(X_train)
X_test_tfidf = vectorizer.transform(X_test)

# Display the shape of the TF-IDF feature matrix
X_train_tfidf.shape
```

(28, 100)

Výstup ukazuje, že trénovací TF-IDF matice má 28 dokumentů a pro každý dokument je 100 vlastností reprezentovaných TF-IDF hodnotami.

## Trénování modelu a hodnocení

V této části použijeme logistickou regresi pro trénování modelu a vyhodnotíme jeho výkon na testovací sadě.

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix

# Train a logistic regression model
model = LogisticRegression()
model.fit(X_train_tfidf, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test_tfidf)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy}')
print('Classification Report:')
print(classification_report(y_test, y_pred, zero_division=0))
print('Confusion Matrix:')
print(confusion_matrix(y_test, y_pred))
```

```
Accuracy: 0.75
Classification Report:
              precision    recall  f1-score   support

           0       0.50      1.00      0.67         1
           1       0.00      0.00      0.00         1
           2       0.83      0.83      0.83         6

    accuracy                           0.75         8
   macro avg       0.44      0.61      0.50         8
weighted avg       0.69      0.75      0.71         8

Confusion Matrix:
[[1 0 0]
 [0 0 1]
 [1 0 5]]
```

*Obrázek 6 - výstup 4*

Accuracy (přesnost) 0,75 znamená, že model správně klasifikoval 75 % testovacích dat.

U klasifikační zprávy model nedokázal správně klasifikovat žádnou z neutrálních recenzí, což je zřejmé z nízkých hodnot preciznosti, odvolání a F1-skóre pro třídu 1, ale naopak ukazuje dobrý výkon při klasifikaci negativních a pozitivních recenzí, což je patrné z vysokého odvolání a F1-skóre pro třídy 0 a 2.

Matice záměn ukazuje, jak byly recenze klasifikovány v porovnání s jejich skutečnými třídami:

- [1 0 0]: 1 negativní recenze byla správně klasifikována jako negativní.
- [0 0 6]: Všechny neutrální recenze byly nesprávně klasifikovány.
- [0 1 5]: 5 pozitivních recenzí bylo správně klasifikováno jako pozitivní, ale 1 pozitivní recenze byla nesprávně klasifikována jako neutrální.

## Generování a vizualizace wordcloudů

Vygenerujeme a zobrazíme wordcloudy pro pozitivní, neutrální a negativní recenze. Word cloud je vizuální reprezentace slov v textu, kde velikost každého slova odpovídá jeho frekvenci nebo důležitosti v textu.

```
import matplotlib.pyplot as plt
from wordcloud import WordCloud

# Generate word clouds
positive_reviews = ' '.join(df_exploded[df_exploded['sentiment'] ==
2]['processed_reviews'])
neutral_reviews = ' '.join(df_exploded[df_exploded['sentiment'] ==
0]['processed_reviews'])
negative_reviews = ' '.join(df_exploded[df_exploded['sentiment'] ==
1]['processed_reviews'])

# Check the lengths of the review strings
print(f'Length of positive reviews: {len(positive_reviews)}')
print(f'Length of neutral reviews: {len(neutral_reviews)}')
print(f'Length of negative reviews: {len(negative_reviews)}')

# Generate word clouds only if there are words in the reviews
positive_wordcloud = WordCloud(width=800, height=400,
background_color='white').generate(positive_reviews) if
len(positive_reviews) > 0 else None
neutral_wordcloud = WordCloud(width=800, height=400,
background_color='gray').generate(neutral_reviews) if
len(neutral_reviews) > 0 else None
negative_wordcloud = WordCloud(width=800, height=400,
background_color='black').generate(negative_reviews) if
len(negative_reviews) > 0 else None

plt.figure(figsize=(15, 5))

if positive_wordcloud:
    plt.subplot(1, 3, 1)
    plt.imshow(positive_wordcloud, interpolation='bilinear')
    plt.title('Positive Reviews')
    plt.axis('off')

if neutral_wordcloud:
    plt.subplot(1, 3, 2)
    plt.imshow(neutral_wordcloud, interpolation='bilinear')
    plt.title('Neutral Reviews')
    plt.axis('off')

if negative_wordcloud:
    plt.subplot(1, 3, 3)
    plt.imshow(negative_wordcloud, interpolation='bilinear')
    plt.title('Negative Reviews')
    plt.axis('off')

plt.show()
```

```
Length of positive reviews: 778
Length of neutral reviews: 247
Length of negative reviews: 148
```


*Obrázek 7 - výstup 5*

Výsledné hodnoty znamenají, že kombinovaný text pro pozitivní recenze má 778 znaků, pro neutrální recenze 247 znaků a pro negativní recenze 148 znaků.

# Závěr

Experimenty ukazují, že analýza sentimentu recenzí filmů IMDB je proveditelná a poskytuje užitečné poznatky o názorech uživatelů. Kombinace knihovny Stanza a modelu logistické regrese umožňuje efektivní klasifikaci recenzí.

Hlavním úkolem experimentu bylo analyzovat sentiment recenzí filmů IMDB a klasifikovat tyto recenze jako pozitivní, neutrální nebo negativní. Tento úkol se nám podařilo splnit, i když o její věrohodnosti lze jistě pochybovat.

Přestože model dosáhl dobré přesnosti, existuje prostor pro vylepšení, například použitím pokročilejších modelů strojového učení, rozšířením seznamů pozitivních a negativních slov nebo vytvořením větší množství dat uživatelských recenzí. Tímto způsobem by se daly výsledky dále zlepšit a model by mohl dosáhnout vyšší přesnosti a lepší klasifikace recenzí.

Výsledky tohoto experimentu mohou být využity k lepšímu porozumění spokojenosti diváků a identifikaci klíčových faktorů, které ovlivňují jejich názory.

Práce s knihovnou Stanza byla jednoduchá a nenarazila jsem na žádné problémy. Celkově lze říci, že knihovna Stanza je velmi dobrý nástroj pro analýzu sentimentu díky své schopnosti efektivně zpracovávat a analyzovat textová data, podpoře vícejazyčných textů a pokročilým modelům strojového učení. Její použití je uživatelsky přívětivé a dobře dokumentované, což usnadňuje integraci do různých projektů zaměřených na analýzu textu a sentimentu. Přestože je její instalace a konfigurace o něco složitější než u některých jiných knihoven, výhody, které poskytuje, tuto nevýhodu výrazně převyšují.