Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Bakalářská práce

Automatická detekce klíčových slov v textu

Plzeň 2022 Jan Krajňák

ZÁPADOČESKÁ UNIVERZITA V PLZNI
Fakulta aplikovaných věd
Akademický rok: 2021/2022

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	**Jan KRAJŇÁK**
Osobní číslo:	**A19B0103P**
Studijní program:	**B0613A140015 Informatika a výpočetní technika**
Specializace:	**Informatika**
Téma práce:	**Automatická detekce klíčových slov v textu**
Zadávající katedra:	**Katedra informatiky a výpočetní techniky**

Zásady pro vypracování

1. Prostudujte oblast zpracování přirozeného jazyka zabývající se automatickou extrakcí klíčových slov.
2. Prozkoumejte existující datové sady a vyberte jednu pro ověření vašich modelů.
3. Analyzujte dodaná data v českém jazyce a připravte druhou datovou sadu ve standardním formátu.
4. Otestujte několik existujících metod na obou datových sadách.
5. Navrhněte vlastní metodu, implementujte ji a otestujte na obou datových sadách.
6. Zhodnoťte dosažené výsledky.

Rozsah bakalářské práce: **doporuč. 30 s. původního textu**
Rozsah grafických prací: **dle potřeby**
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

Dodá vedoucí bakalářské práce.

Vedoucí bakalářské práce: **Ing. Jakub Sido**
Nové technologie pro informační společnost

Datum zadání bakalářské práce: **4. října 2021**
Termín odevzdání bakalářské práce: **5. května 2022**

L.S.

Doc. Ing. Miloš Železný, Ph.D.
děkan

Doc. Ing. Přemysl Brada, MSc., Ph.D.
vedoucí katedry

V Plzni dne 14. října 2021

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 4. května 2022

Jan Krajňák

Abstract

This thesis deals with automatic keywords extraction from documents. First, it analyzes already existing approaches to solve the problem then tests some of the methods on a dataset from SemEval 2010 competition. Same methods are then tested on a completely new czech dataset which has been obtained with cooperation with ČTK. As this new corpus contains enough annotated documents, the work in the final part deals with a proposal of a supervised method based on BERT models and the subsequent comparisson with already existing methods.

Abstrakt

Tato práce se zabývá problematikou automatické extrakce klíčových slov z textu. Nejprve jsou analyzovány již existující přístupy k řešení problému. Některé z existujících přístupů byly vybrány a vyzkoušeny na anglické datové sadě, použité i při soutěži SemEval 2010. Stejné metody byly rovněž otestovány na zcela nové, nikdy netestované datové sadě získané při spolupráci s Českou tiskovou kanceláří. Jelikož tento nový korpus obsahuje dostatek označených dat, práce se v další části zabývá návrhem metody využívající přístup učení s učitelem založené na BERT modelech a následném porovnání s již existujícími metodami.

Poděkování

Rád bych poděkoval Ing. Jakubu Sidovi za cenné rady a věcné připomínky, které mi velmi pomohly při vypracování bakalářské práce.

Obsah

1 Úvod

Klíčová slova v textu jsou termíny, které nejlépe charakterizují obsah celého dokumentu. Za pomoci klíčových slov může čtenář pátrat po hledané informaci s mnohem vyšší efektivitou. Jsou často používána v problematice zpracování přirozeného jazyka, například v kategorizaci textu nebo při získávání informací.

Cílem této práce je prozkoumat problematiku extrakce klíčových slov a analyzovat existující metody. Následně navrhnout vlastní postup pro extrakci klíčových slov a porovnat ho s vybranými metodami. Testování bude probíhat nejprve na anglické datové sadě, tvořené dokumenty ze čtyř různých oborů, kde se každý dokument skládá z 6 až 8 stránek textu. Poté budou stejné metody otestovány i na zcela nové české datové sadě získané při spolupráci s Českou tiskovou kanceláří (ČTK).

Metody pro extrakci klíčových slov obvykle bývají řešeny pomocí algoritmů učení bez učitele vzhledem k nedostatku označených článků (trénovacích dat) k důkladnému naučení modelu. Tato česká datová sada nebyla primárně určena pro problematiku extrakce klíčových slov. Nicméně díky tomu, že obsahuje dostatek anotovaných článků, může nad ní být vytvořena metoda pro extrakci klíčových slov využívající algoritmů učení s učitelem (*supervised*).

Výstupem práce bude studie úspěšnosti vybraných metod s vlastní, nově navrženou *supervised* metodou. Testování bude probíhat na dobře zdokumentovaných anglických datových sadách, ale také na nové, nikdy netestované české datové sadě.

2 Detekce klíčových slov

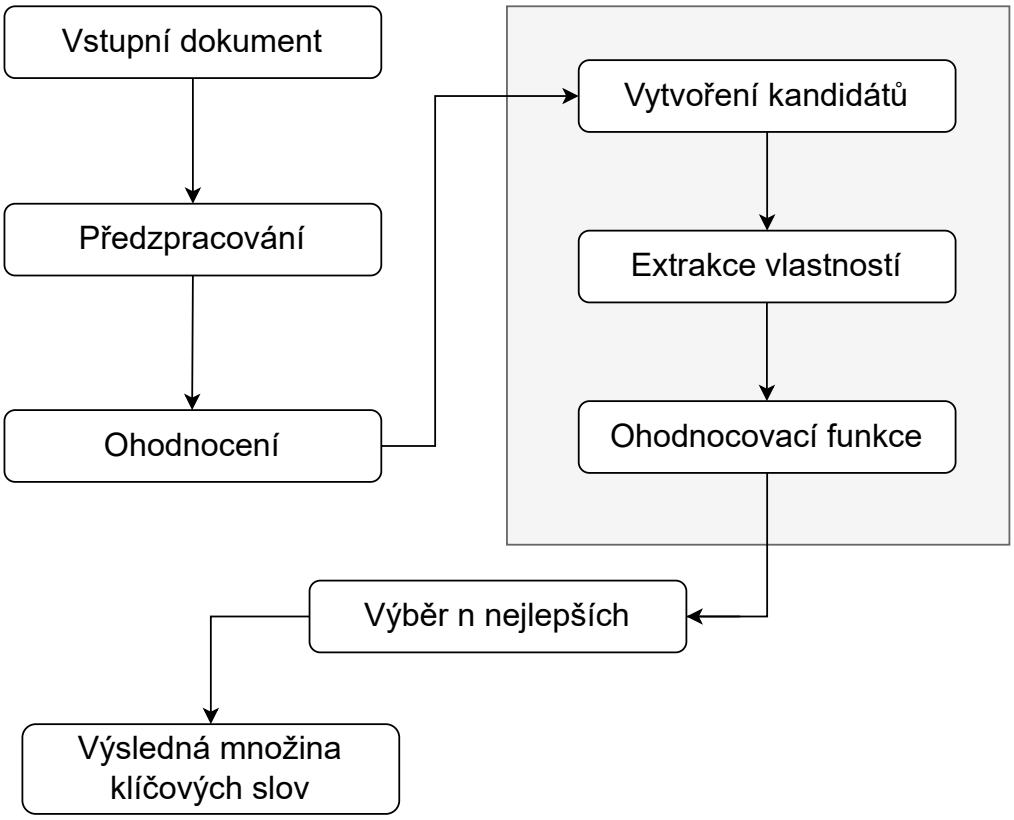
Detekce klíčových slov je úloha, jejímž účelem je automatické vytvoření množiny slov, které nejlépe vystihují obsah celého dokumentu. Výsledek může být užitečný pro nesčetné množství lidí, kteří se potřebují s daným tématem seznámit. Například novinář hledající informace o určitém tématu by byl pravděpodobně rád, kdyby mu byl nabídnut souhrn klíčových slov daného článku. Bylo by to také přínosné pro knihovníky a čtenáře, kteří si přejí rychlý pohled na daný dokument, nebo pro uživatele internetu, kteří hledají nejrelevantnější webové stránky, nebo dokonce pro vyhledávače, které chtějí zlepšit svůj proces indexování, načítání nebo prezentace výsledků [3]. Avšak anotace klíčových slov člověkem je časově velmi náročná. Proto je automatická extrakce klíčových slov stále populárnějším tématem, protože program dokáže vytvořit množinu klíčových slov během několika vteřin či minut i pro velmi rozsáhlé dokumenty [2]. V současné době většina dokumentů neobsahuje anotovaná klíčová slova, i kvůli časové náročnosti, a proto je zapotřebí je automaticky generovat [12].

Ačkoli byl tomuto tématu v průběhu let věnován značný výzkum, problém extrahování relevantních klíčových slov s vysokou přesností zůstává nevyřešen. Přispívá k tomu několik faktorů. Mezi nimi jsou potíže s definováním relevance a samotná jazyková rozmanitost (např. texty různých jazyků, velikostí a domén). Dalším značným problémem je nedostatek dokumentů, které mají anotovaná klíčová slova. Všechny tyto problémy zdůrazňují obtíže související s vývojem vytvoření globálního řešení, které funguje stejně ve všech scénářích, a motivuje potřebu dalšího výzkumu.

Automatickou anotaci klíčových slov lze rozdělit na dvě podkategorie – přiřazování a extrakci. Obě dvě kategorie se zabývají stejným problémem, a to určením nejvíce relevantních slov pro daný dokument. Zatímco u úlohy přiřazení klíčových slov, je předem určen slovník obsahující termíny, ze kterých se vybírají ty nejvhodnější, tak u extrakce jsou klíčová slova vybírána z jednotlivých slov samotného dokumentu. Tato práce se zabývá kategorií extrakce.

V současné době již existuje několik způsobů jak řešit problematiku extrakce klíčových slov, lišících se v postupu řešení. Obecný postup metod extrakce klíčových slov odpovídá diagramu 2.1. První z algoritmů, které řešily tuto problematiku, využívaly prostého statistického přístupu, jako je například frekvence slov, avšak u těchto algoritmů se ukázalo, že nemají příliš uspokojivé výsledky [14]. V dnešní době je jedno z nejpoužívanějších

řešení hluboké učení [15]. Metody se dále dělí na ty, které využívají učení s učitelem a učení bez učitele. Některé dokonce nabízí i extrakci klíčových frází nikoli jen slov. Jednotlivé přístupy metod jsou popsány v kapitole 3.



Obrázek 2.1: Obecný, zjednodušený postup metod extrakce klíčových slov. Část uvnitř šedého rámečku se může mírně lišit v závislosti na zvoleném přístupu dané metody

Zdroj: [1]

3 Existující metody extrakce klíčových slov

V této kapitole je naznačeno základní rozdělení metod zabývajících se problematikou automatické extrakce klíčových slov v textu. Jak již bylo zmíněno v kapitole 2, tuto problematiku lze rozdělit na dvě podkategorie, a to přiřazování klíčových slov a extrahování klíčových slov. Hlavním rozdílem je, že u přiřazování se klíčové termíny vybírají z předem definovaných databází, takže se může například stát, že přiřazený termín se v dokumentu vůbec nevyskytuje. Zatímco u extrahování jsou brána v potaz všechna slova dokumentu. Z nich jsou následně vybrána ta, která dokument nejlépe charakterizují.

Přístupy pro extrahování klíčových slov lze dále rozdělit na metody učení s učitelem (*supervised methods*), metody učení bez učitele (*unsupervised methods*) a metody s kombinací učení s učitelem a bez učitele (*semi-supervised methods*) [2].

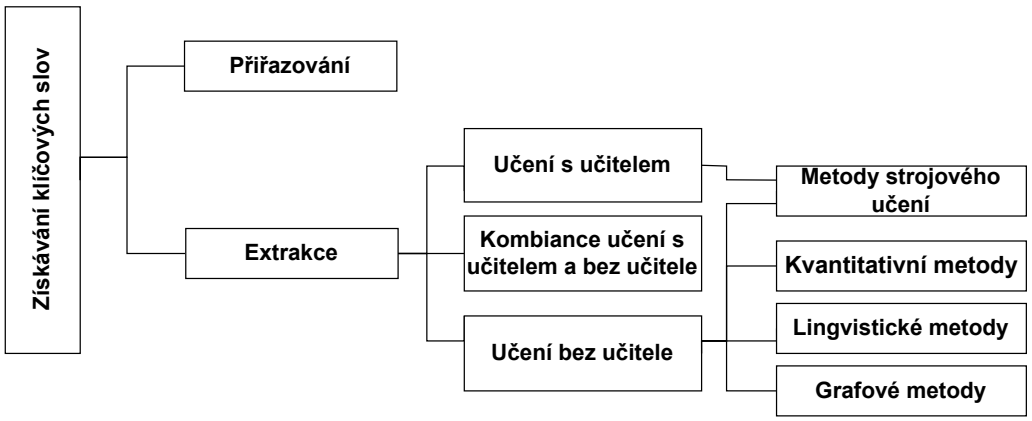
Tato práce se zabývá především postupy s učením bez učitele, které se dále dělí dle jejich přístupu k problematice na kvantitativní, lingvistické, grafové a metody strojového učení 3.1.

Algoritmy řešící problematiku extrakce klíčových slov se zpravidla skládají ze tří hlavních kroků [7]:

1. Výběr kandidátů – prvním krokem je výběr slov v textu, která by mohla být slovem klíčovým. To znamená například odstranění všech stop slov, matematických vzorců, zkratek a podobně.

2. Určení významnosti kandidátů – v tomto kroku je každému kandidátu přiřazena číselná hodnota, reprezentující jeho významnost v textu. V tomto kroku se jednotlivé algoritmy nejvíce odlišují, protože výpočet této hodnoty může být různý na základě přístupu dané metody. Může se například skládat z četnosti termínu, jeho pozice v dokumentu, fontu, jakým byl napsán a dalších statistických či lingvistických vlastností [3].

3. Výběr nejlepších slov – v posledním kroku se vybere n nejlepších slov, dle hodnoty spočítané v kroku předchozím. Číslo n může být pevně stanoveno nebo se může dynamicky měnit v závislosti na délce dokumentu a nebo se mohou vybrat všechny termíny s hodnotou přesahující předem stanovenou hranici.



Obrázek 3.1: Rozdělení způsobů získávání klíčových slov z textu
Zdroj: `https://hrcak.srce.hr/140857`

3.1 Učení s učitelem

Hlavní myšlenkou metod učení s učitelem pro určení klíčových slov je převést extrakci klíčových slov na úlohu binární klasifikace [2]. Průkopníky v problematice extrakce klíčových slov byly postupy využívající učení s učitelem a to metody Kea [24], která pro určení klíčového slova využívala Quinlanův C4.5 algoritmus a GenEx [22], který využíval Naivní Bayesův klasifikátor. Algoritmy nejprve vybraly kandidáty z textu, hlavním kritériem pro jejich výběr byla frekvence a pozice termínu v dokumentu. Poté následovala binární klasifikace, zda je kandidát klíčovým slovem či ne. V současnosti mezi nejúspěšnější a nejpoužívanější metody zbývající se problematikou NLP (*Natural Language Processing*) patří přístupy založené na modelech typu BERT.

Učení s učitelem vyžaduje ke svému natrénování ručně anotovanou datovou sadu, tedy dokument a k němu ručně anotovaná odpovídající klíčová slova, což může být někdy problém, protože ne vždy autoři dokumentů dodávají i seznam klíčových slov. A manuální anotace klíčových slov velkého množství dokumentů či dokumentů rozsáhlých, může být velmi zdlouhavý proces. I proto se tato práce zabývá v první řadě metodami učení bez učitele.

3.1.1 Modely typu BERT

BERT (*Bidirectional Encoder Representations from Transformers*) modely využívají tzv. *Transformery*, což je mechanismus, který se učí kontextové vztahy mezi slovy v textu. Ve své původní podobě obsahuje *Transformer* dva samostatné mechanismy – enkodér, který čte textový vstup, a dekodér, který vytváří předpověď pro daný úkol. Protože cílem BERT modelů je generovat

jazykový model, je zapotřebí pouze mechanismus enkodéru [4].

Pro reprezentaci jazyka používá BERT vícevrstvé obousměrné *transformerové* enkodéry. Na základě hloubky architektury modelu jsou definovány dva primární typy modelů BERT – *BERT-Base* a *BERT-Large*. Model *BERT-Base* využívá 12 vrstev neuronové sítě se skrytou velikostí 768 a se 12 tzv. *self-attention heads* [23] a má přibližně 110M trénovatelných parametrů. *BERT-Large* používá 24 vrstev neuronové sítě se skrytou velikostí 1024 a počtem *self-attention heads* 16 a má přibližně 340M trénovatelných parametrů [4]. BERT používá stejnou modelovou architekturu pro všechny úlohy, ať už jde o NLI (*natural language inference*) – určování pravdivosti hypotéz, klasifikaci, zodpovězení otázek, shrnutí abstraktu, generování odpovědí, rozdělení významu slova a mnoho dalších úkonů týkajících se NLP (zpracování přirozeného jazyka).

Na vstupu BERT model očekává sekvenci tokenů, přesněji řečeno, využívá tzv. *WordPiece* tokenizér. Ten funguje tak, že slova rozděluje buď do celých tvarů (např. jedno slovo se stane jedním tokenem) nebo do slovních částí — kde lze jedno slovo rozdělit na více tokenů 3.1.1.

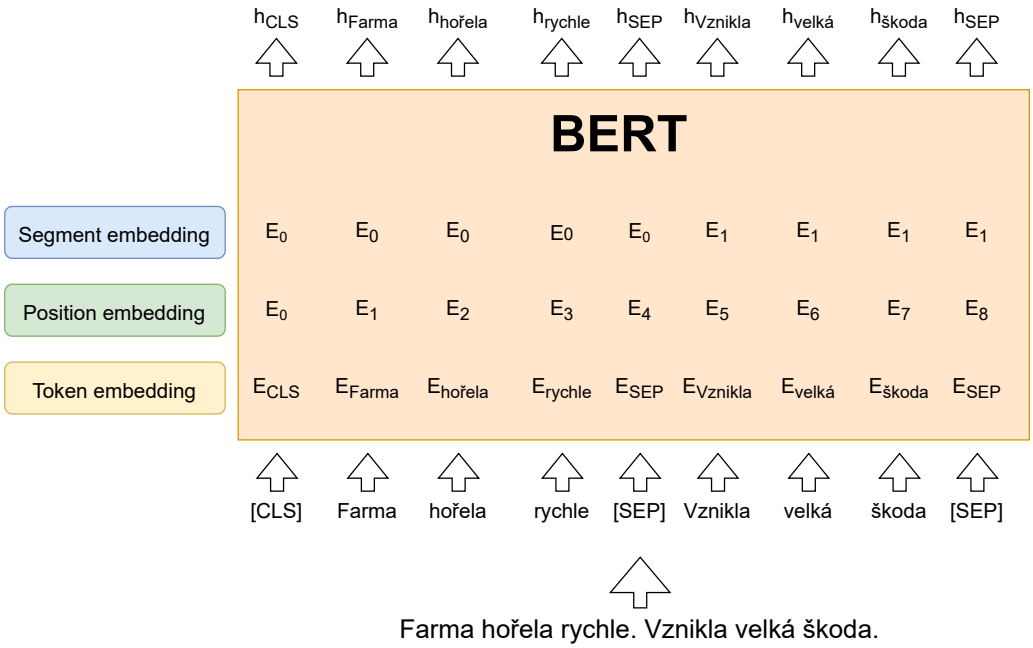
Slovo	**Token(y)**
potravinami	['potravin', '##ami']
průmyslová	['průmysl', '##ová']
údajů	['údajů']
severočeský	['servero', '##čes', '##ký']
český	['čes', '##ký']
být	['být']

Tabulka 3.1: Rozdělení slov na tokeny

Využití těchto tokenů umožňuje BERT modelům snadno identifikovat související slova, protože obvykle budou sdílet některé stejné vstupní tokeny, které se následně vkládají do jednotlivých vrstev neuronové sítě. BERT dále zavádí tři speciální tokeny – první token každé sekvence je vždy speciální klasifikační token *[CLS]*. Tento token charakterizuje celou sekvenci a nejčastěji se využívá v klasifikačních úlohách. Druhým speciální tokenem je token *[SEP]*. Ten je vkládán na konec každé věty, a pokud vstupní sekvence obsahuje více než jednu větu, pomáhá modelu pochopit, do jaké části textu daný token patří. Posledním speciálním tokenem je *[UNK]*, který značí, že daný token, se nenachází v originálním slovníku, na kterém byl BERT model natrénován.

Neuronová síť dokáže pracovat pouze s číselnými hodnotami, tudíž když byl BERT model trénován, každý token dostal jedinečné ID (identifikační číslo). Proto, když chceme použít předem trénovaný model BERT, je zapotřebí nejprve převést každý token ve vstupní větě na jeho odpovídající jedinečné ID.

Kromě token embedding používá BERT pro token rovněž position embedding a segment embedding 3.2. Position embeddings obsahuje informace o pozici tokenů v celé sekvenci. A segment embeddings určuje, o kterou větu se jedná.



Obrázek 3.2: Reprezentace modelu typu BERT a transformace vstupu

Fine tuning Na model typu BERT se následně aplikuje tzv. *fine tuning*, což je v podstatě použití natrénované neuronové sítě jako základ pro nový účelově specifický model. Nově vytvořený výsledný model, nezávisle na typu úlohy, je téměř stejný jako původní BERT.

Fine tuning BERT modelu se provádí obvykle přidáním jedné další vrstvy za poslední vrstvu BERT a trénováním modelu jen na několik epoch. Je to poměrně rychlá záležitost v porovnání s předtrénováním neuronové sítě, protože obecné jazykové struktury již byly naučeny během původního tréninku BERT modelu. Dle [23] zůstává při *fine tuning* většina hyperparametrů stejná jako v původním tréninku BERT modelu.

3.2 Učení bez učitele

Metody učení bez učitele se stávají stále více důležitými vzhledem k tomu, že manuální anotace klíčových slov je velmi drahá a časově náročná [3]. Právě oproti přístupům učením s učitelem, je hlavní výhodou těchto metod, že nepotřebují žádná trénovací data.

Základní metodou mezi *unsupervised* přístupy je TF-IDF, která porovnává frekvenci termínu v dokumentu s jeho frekvencí ve velké datové sadě. Ačkoli je implementace poměrně jednoduchá, TF-IDF vyžaduje přístup k velkému korpusu, který nemusí být vždy k dispozici. K překonání tohoto omezení bylo navrženo několik alternativních přístupů. Mezi další nejznámější metody tohoto typu patří například RAKE [19], YAKE! [3], TextRank [14] a z něho vycházející SingleRank, ExpandRank či Topic Rank.

3.2.1 Kvantitativní metody

Kvantitativní metody jsou jedny z nejjednodušších metod pro identifikaci klíčových slov v textu, navíc nepotřebují žádné předem definované slovníky a jsou tedy jazykově i doménově nezávislé. Princip fungování je založen na výpočtu statistických údajů pro každý termín a na základě těchto údajů je každý termín ohodnocen. Mezi nejběžněji používané statistické údaje patří například frekvence slov, kolokace slov či frekvence vzájemných výskytů. Avšak vzhledem k tomu, že spoléhají jen na tyto statistické údaje, může se stát, především v odborných dokumentech, že přehlédnou významově důležitý termín, který je v textu zmíněný pouze jednou. Mezi kvantitativní metody patří například TF-IDF nebo RAKE.

3.2.2 Lingvistické metody

Pro určení klíčových slov používají lingvistické metody jazykové vlastnosti slov, vět a dokumentů. Algoritmy těchto metod se skládají z lexikální, syntaktické a diskurzivní analýzy [26], čili využívají například slovních druhů (podstatná jména mohou mít větší váhu), vztahů mezi slovy nebo slovosledu. Existují přístupy, které využívají jak statistických údajů, tak jazykových vlastností a obecně lze dokázat, že přístupy, které využívají i jazykové vlastnosti překonávají ty, které nikoli [15].

3.2.3 Metody založené na grafech

Princip metod založených na grafech je převést text na graf, ve kterém vrcholy reprezentují jednotlivé termíny a hrany relace mezi nimi. Podmínka

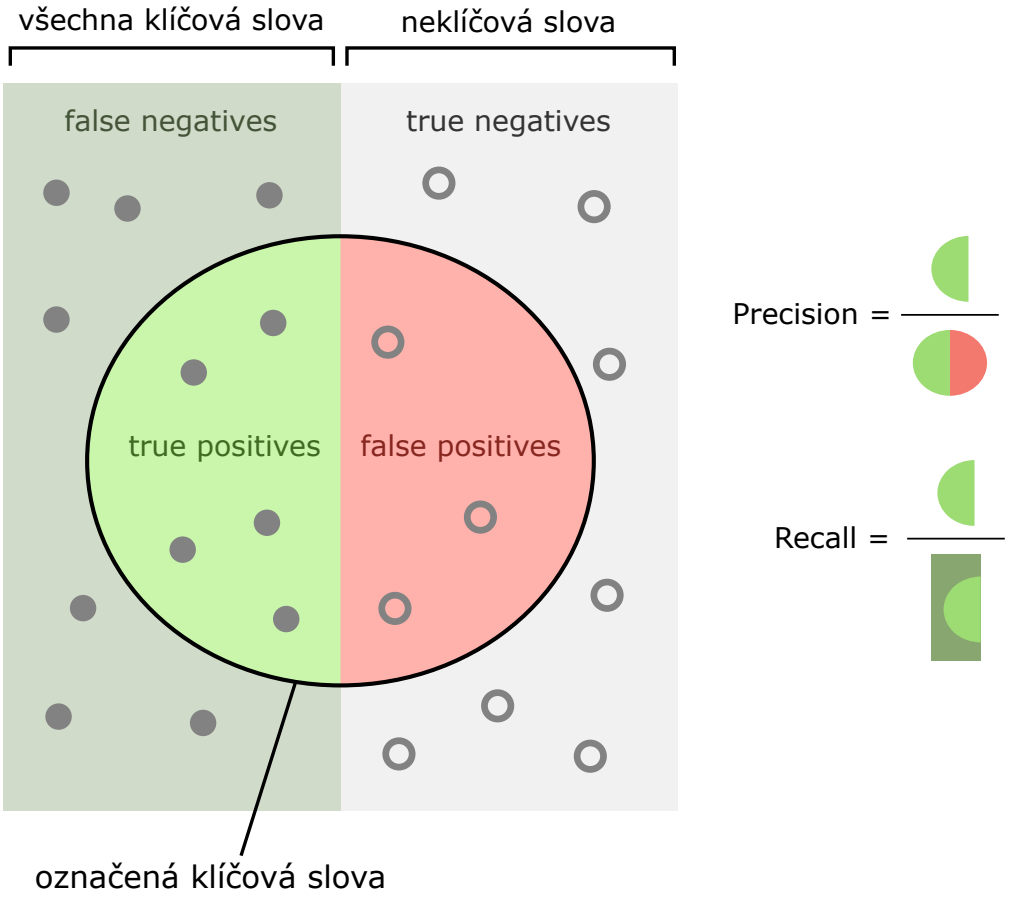
pro vytvoření hrany mezi dvěma vrcholy závisí na zvoleném přístupu jednotlivých metod. Některé vytváří hrany na základě výskytu dvou termínů vedle sebe, ve stejné větě nebo odstavci, jiné mohou například vytvářet hrany mezi takovými termíny, které mají podobnou sémantiku, jsou synonymy či antonymy. Jednou z nejznámějších a nejpoužívanějších metod založené na grafu je TextRank [14].

3.2.4 Metody strojového učení

Metody strojového učení mohou používat buď učení s učitelem nebo učení bez učitele. Přístup učení s učitelem, který je v problematice automatické extrakce klíčových slov tím používanějším [2], vytváří trénovací model, na kterém jsou ručně anotovaná data natrénována. Trénovací model může být založen například na Bayesově teorému, SVM, C4.5 a podobně. Jelikož tento přístup vyžaduje trénovací data, jeho nevýhodou může být častější závislost na doméně dokumentů.

3.3 Metrika úspěšnosti metod

Nejběžnějším způsobem pro zjištění úspěšnosti metod extrakce klíčových slov je určování hodnot přesnost (*precision*) a úplnost (*recall*) a jejich harmonického průměru, takzvané F-míry (*F-measure*). Po dokončeni algoritmu určité metody extrakce klíčových slov dostáváme množinu slov, kterou daná metoda určila jako klíčovou. Pro získaní hodnot precision a recall poté každé toto slovo zařadíme do jedné ze čtyř skupin dle obrázku 3.3.



Obrázek 3.3: Precisoin a recall
Zdroj: `https://en.wikipedia.org/wiki/Precision_and_recall`

True positives jsou taková slova, která byla metodou vybrána jako klíčová a skutečně klíčová jsou. False positives jsou naopak taková slova, která byla označena jako klíčová, ale klíčová ve skutečnosti nejsou. True negatives jsou poté slova, která metoda správně určila jako neklíčová. A nakonec false negatives jsou ta slova, která metoda určila jako neklíčová, ale ve skutečnosti klíčová jsou.

Jakmile známe hodnoty precision a recall, výsledná F-míra se poté dopočítá dle následujícího vzorce 3.1.

$$F = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{3.1}$$

Samotná F-míra lze poté rozdělit ještě na dva způsoby výpočtu – mikro F-míra (mikro F1) a makro F-míra (makro F1). Makro F-míra se vypočítá tak, že se vezme aritmetický průměr (nevážený průměr) dílčích F1. V případě klíčových slov je běžnější využití mikro F-míry. Pro výpočet mikro F-míry je

potřeba nejprve vypočítat sumu všech true positives, false positives a false negatives. Z nich poté lze spočítat mikro precision a mikro recall a nakonec samotná mikro F-míra.

4 Analýza jednotlivých metod

V této kapitole budou podrobně vysvětleny principy jednotlivých metod, které budou využity jako výchozí bod (tzv. *baseline*) a způsob porovnání s navrženou, *supervised* metodou, kterou se tato práce zabývá. Vybrány budou zástupci kvantitativních, lingvistických, grafově založených metod i metod strojového učení.

4.1 TF-IDF

TF-IDF je statistická technika určená pro vyhodnocování relevance slova v dokumentu. Toto vyhodnocování je založeno na výsledku dvou metrik TF(*term frequency* – četnost slova) a IDF(*inverse document frequency* – převrácená četnost slova ve všech dokumentech). Oblast využití je velmi široká, mezi nejdůležitější oblasti patří například zpracování přirozeného jazyka či automatizovaná analýza textu.

Složka TF obsahuje frekvenci výskytů každého slova v dokumentu. Počet výskytů daného slova se může zásadně lišit v závislosti na délce dokumentu. Čili frekvence určitého slova v dokumentu o 100 slovech může být mnohem nižší než frekvence stejného slova v dokumentu o 1000 slovech, což ale ihned nemusí znamenat, že relevance slova je v delším dokumentu vyšší. Právě proto se počet výskytů všech slov navíc normalizuje – vydělí se celkovým počtem slov v dokumentu. To ale stále neřeší problém takzvaných stop slov. Stop slova jsou takové termíny, které se takřka pokaždé vyskytují v textu, ale ve skutečnosti nenesou žádný význam, typicky jsou to předložky a spojky (v angličtině pak navíc třeba členy `a, an, the`).

Nejen pro řešení problému stop slov je zde druhá složka – IDF. Pro určení IDF je zapotřebí nejprve zjistit hodnotu DF(*document frequency* – četnost slova v dokumentech). Ta nám říká, v kolika dokumentech, z celé datové sady, se dané slovo vyskytuje. Stejně jako u složky TF se i zde počet výskytů normalizuje, IDF je už po té jen inverzí k DF. Díky tomu bude výsledná hodnota značně nízká pro velmi často vyskytující se slova, jako jsou například již zmiňovaná stop slova. Problém u IDF může nastat u velice rozsáhlých datových sad (například 10000 dokumentů), v takovém případě by výsledná hodnota mohla být enormní. Z tohoto důvodu se hodnota IDF logaritmuje.

Výslednou hodnotu, která udává relevanci daného slova, pak lze získat

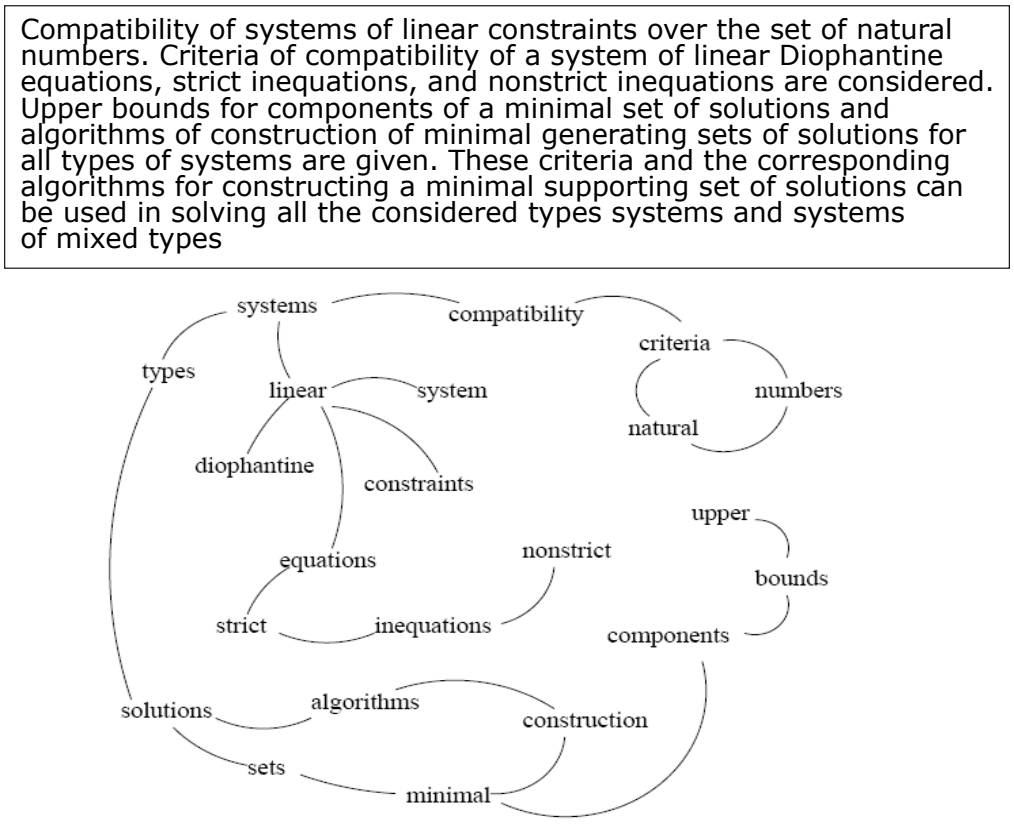
vynásobením obou složek 4.1.

$$\mathrm{TFIDF}(t,d) = \mathrm{TF}(t,d) \cdot log(\frac{N}{\mathrm{DF}+1}) \tag{4.1}$$

4.2 TextRank

TextRank je metoda založená na grafech a je nejčastěji používaná pro určení relevance vět v textu nebo pro autmatickou extrakci klíčových slov. Vychází z algoritmu PageRank [16], který se zabývá určováním relevance webových stránek.

Metody založené na grafech si nejprve řešenou úlohu převedou na graf, kde poté dochází k ohodnocování každého vrcholu. Na obrázku 4.1 lze vidět příklad vytvořeného grafu z abstraktu anglicky psaného dokumentu z původní testovací sady, kterou TextRank využíval [14]. Způsob ohodnocování vrcholu je založen na počtu dalších vrcholů, se kterými je spojen. To znamená, že čím více hran do vrcholu vede, tím důležitější vrchol je.



Obrázek 4.1: Převedený text na graf TextRankem
Zdroj: `https://aclanthology.org/W04-3252.pdf`

Pro orientovaný graf $G = (V, E)$, kde V značí množinu všech vrcholů a E je množina všech hran grafu, lze matematicky ohodnocovaní vrcholů vyjádřit následujícím vzorcem 4.2.

$$S(V_i) = (1 - d) + d \cdot \sum_{j \in In(V_i)} \frac{1}{|Out(V_j)|} S(V_j) \tag{4.2}$$

Kde pro vrchol V_i je $In(V_i)$ taková množina vrcholů, ze kterých existuje hrana do vrcholu V_i. $Out(V_i)$, je taková množina vrcholů, do kterých vede hrana z vrcholu V_i. d je takzvaný faktor tlumení, který může být v rozsahu $(0, 1)$ a jeho účelem je vložení pravděpodobnosti do modelu, díky níž může dojít k přeskočení z jednoho vrcholu na náhodný další [14]. Na začátku algoritmu je každému vrcholu přiřazena libovolná hodnota a poté výpočet iteruje, dokud konvergence neklesne pod určený práh. Po dokončení algoritmu má každý vrchol přiřazenou hodnotu, která reprezentuje jeho důležitost v grafu.

Očekávaným výsledkem TextRanku je množina slov, která nejlépe charakterizuje obsah celého dokumentu. Proto je tedy zapotřebí převést si samotný text na graf. Jako vrcholy grafu jsou zvoleny jednotlivé termíny z textu a pro určení hran TextRank využívá společný výskyt slov v určité vzdálenosti definovanou takzvaným okénkem, která dle autorů TextRanku může být nastavena v rozsahu 2 až 10. Pokud bychom tedy měli okénko velikosti 3, existovala by hrana mezi takovými slovy, která jsou od právě řešeného slova ve vzdálenosti 3 napravo i nalevo. Při přidávání vrcholů do grafu, mohou slova projít takzvaným syntaktickým filtrem, který redukuje slova, na základě jejich lexikální podoby. Například může do grafu přidat pouze podstatná jména a slovesa. Nejlepších výsledků dosahuje TextRank se syntaktickým filtrem nastaveným pouze pro podstatná a přídavná jména [14]. Algoritmus TextRanku se skládá ze 4 kroků:

1. Pomocí tokenizace je každému termínu přiřazen jeho slovní druh. Tento krok je potřeba, aby bylo možné použít syntaktické filtry.

2. Všechny termíny, které prošly syntaktickým filtrem jsou přidány do grafu a jsou vytvořeny hrany v závislosti na velikosti okénka. Všem vrcholům je přiřazena inicializační hodnota 1. Práh pro ukončení cyklu je nastavený na 0.0001.

3. Po dokončení cyklu má každý vrchol vlastní konečnou hodnotu. Vrcholy jsou seřazeny sestupně, podle jejich ohodnocení a jako finální klíčová slova je vybráno N prvních vrcholů.

4. Celý text se projde ještě jednou, a pokud se dvě či více vybraných klíčových slov nachází v textu bezprostředně za sebou, vznikne z nich klíčová fráze.

4.3 YAKE!

YAKE je stejně jako TF-IDF a TextRank z rodiny metod učení bez učitele (`unsupervised`). Algoritmus začíná předzpracováním vstupních dat, následuje extrahování vlastností slov, výpočet hodnoty jednotlivých termínů, příprava kandidátů na klíčová slova a naposled odstranění podobných slov.

Předzpracování textu znamená v případě YAKEu upravení vstupních dat do dobře čitelného formátu pro počítač a odstranění takových termínů, které nenesou důležitou informaci. Daný text je nejprve rozdělen na věty a následně je každá věta rozdělena na ještě menší části (například vedlejší věty). Jednotlivá slova jsou poté označena značkami, podle toho, zda se jedná o číslo, akronym, slovo začínající velkým písmenem, neanalyzovatelná slova (například taková slova, která obsahují více interpunkčních znamének, jako je třeba URL adresa), nebo analyzovatelná slova, což jsou taková slova, která nespadají ani do jedné z vyjmenovaných kategorií. Dále je každý termín převeden na malá písmena a v závislosti na jazyku jsou odstraněna z textu stop slova. Výsledkem předzpracování je tedy seznam vět, které jsou rozděleny na takzvané *chunky*, které obsahují označené termíny.

Po předzpracování přichází na řadu statistická analýza textu. Nejprve se vytvoří datová struktura, která bude obsahovat statistické údaje pro každý termín. Statistické údaje pro jednotlivá slova, které YAKE získává jsou: frekvence termínu; index věty, ve které se termín nachází; frekvence výskytu termínu s velkým počátečním písmenem a frekvence výskytu termínu jako akronym. Všechny tyto údaje jsou uloženy do struktury pro další využití. Jakmile jsou struktury pro každý termín připraveny, přichází na řadu extrahování vlastností. YAKE extrahuje pro každý termín pět vlastností:

1. Velikost písmen výrazu – YAKE udává větší relevanci slovům, která se častěji vyskytují buď jako akronym nebo často začínají velkým písmenem (kromě začátku věty). Výsledkem je hodnota získaná z vzorce 4.3, kde $TF(U(t))$ je počet výskytů slova s počátečním velkým písmenem, $TF(A(t))$ je počet výskytů jako akronym a $TF(t)$ je frekvence daného termínu.
$$\mathrm{T}_{case} = \frac{max(\mathrm{TF}(U(t)), \mathrm{TF}(A(t)))}{ln(\mathrm{TF}(t))} \tag{4.3}$$
2. Pozice termínu – dalším kritériem pro určení relevance slova je pozice

termínu v textu. Slova, která se vyskytují na začátku dokumentu bývají častěji klíčovými slovy, než ta slova, která se vykytují až ke konci [5]. Výsledná hodnota toho kritéria se tedy spočítá dle 4.4

$$\mathrm{T}_{position} = ln(ln(3 + median(Sen_t))) \tag{4.4}$$

kde Sen_t je množina indexů vět, ve kterých se daný termín vyskytuje.

3. Normalizovaná frekvence výskytu – následující extrahovaná vlastnost je počet výskytů termínu. V případě YAKEu je samotná frekvence výskytu jednoho termínu navíc vydělena průměrnou frekvencí, aby se předešlo příliš vysokým hodnotám v abnormálně dlouhých dokumentech.

4. Kontext termínu – toto kritérium se snaží eliminovat slova, která se v textu vyskytují sice často, ale pokaždé v jiném kontextu. Čím větší je počet rozdílných slov na obou stranách právě řešeného termínu, tím menší je jeho relevantnost [13]. Výslednou hodnotu lze tedy dostat za pomocí vzorce 4.5.

$$\mathrm{T}_{context} = 1 + (\mathrm{DL} + \mathrm{DR}) \cdot \frac{\mathrm{TF}(t)}{\mathrm{TF}} \tag{4.5}$$

kde DL (DR) je disperze nalevo, respektive napravo, od kandidáta a $MaxTF$ je maximální frekvence ze všech termínů.

5. Výskyt v různých větách – posledním kritériem je počet rozdílných vět, ve kterých se daný termín vyskytuje. Předpokládá se, že v čím více větách se slovo vyskytuje, tím je vyšší pravděpodobnost toho, že slovo bude klíčové. Číselná hodnota je tedy poměrem mezi počtem vět, ve kterých se termín nachází a celkovým počtem vět v dokumentu.

Po extrahování jednotlivých vlastností termínu přichází na řadu určení finálního skóre každého termínu, což znamená zkombinovat jednotlivá kritéria do výsledné hodnoty S_t 4.6.

$$S_t = \frac{T_{context} \cdot T_{pos}}{T_{case} + \frac{TF_{norm}}{T_{context}} + \frac{T_{sent}}{T_{context}}} \tag{4.6}$$

Dalším krokem algoritmu je generování kandidátů na klíčová slova. To je založeno na principu posuvného okénka velikosti 1 až `n` a dle termínů v okénku je generována posloupnost slov – kandidátů na klíčová slova. Aby se předešlo k vytvoření v dokumentu neexistujících spojení, výslední kandidáti na klíčová slova (fráze) jsou vybírány pouze v rámci vět, čili pokud jsou

dvě slova oddělena tečkou, nebudou mezi kandidáty zařazeny. Po vytvoření všech kandidátů je jednotlivým slovům (frázím), přiřazeno jejich finální skóre v závislosti na hodnotách získaných v předešlém kroku.

V posledním kroku se algoritmus snaží eliminovat vzájemně podobné termíny a to za pomoci Levenštejnovy vzdálenosti. Kandidáty na klíčová slova jsou nejprve seřazeny vzestupně, a pokud Levenštejnova vzdálenost mezi dvěma kandidáty přesáhne určený práh, kandidát s vyšším skóre bude odebrán z množiny klíčových slov. Výsledkem algoritmu je poté uživatelem `n` zadaných klíčových slov (frází) vybraných z finální, vzestupně seřazené množiny kandidátů.

4.4 KeyBert

KeyBert je stejně jako předchozí zmíněné metody *unspupervised* (nevyžaduje žádná trénovací data). Je založen na předtrénovaných modelech typu *BERT* 3.1.1, díky nimž dokáže získat potřebné informace o jednotlivých výrazech i celém dokumentu a poté za pomoci kosinové vzdálenosti určit, které slovo (fráze) je celému dokumentu nejvíce podobné. Dále KeyBert umožňuje filtrovat podobná klíčová slova z výsledné množiny díky jedné ze dvou metodik (*MMR* a *MaxSum*).

Samotný algoritmus KeyBert se skládá ze čtyř kroků. Nejprve jsou za pomoci jakéhokoli předtrénovaného BERT modelu extrahovány tzv. *embedding* celého dokumentu. *Embedding* je v podstatě vektorová reprezentace daného textu, se kterou následně pracuje neuronová síť. Následuje extrahovaní *embedding* pro jednotlivé n-gramy za pomoci techniky *Bag of Words* [27]. Každý kandidát na klíčové slovo je poté reprezentován vektorem s pevnou velikostí, aby mohla být spočítána výsledná kosinová vzdálenost. Posledním krokem je samotný výpočet kosinové vzdálenosti pro každého z kandidátů, a taková slova, která jsou celému dokumentu nejvíce podobná (mají nejnižší kosinovou vzdálenost), jsou vyhodnocena jako hledaná klíčová slova [6].

MMR (*Maximal Marginal Relevance*) je jedním ze dvou způsobů, jakým KeyBert vybírá nejvíce relevantní slova. Cílem této metodiky je omezit výskyt podobných slov ve výsledné množině. *MMR* zvažuje podobnost slov (frází) s celým dokumentem spolu s podobností již vybrané množiny klíčových slov – čili počítá kosinovou vzdálenost nejen mezi jednotlivými n gramy a celým dokumentem, ale i mezi n gramy a slovy nacházejícími se již v množině označených slov. Výsledkem je tedy výběr co nejrozmanitějších kandidátů vzhledem k obsahu celého dokumentu.

MaxSum Účel této metodiky je stejný jako u *MMR*, tedy odstranit z výsledné množiny slova, která jsou si podobná. Princip spočívá ve vytvoření množiny obsahující dvojnásobný počet výrazů. Mezi jednotlivými výrazy je poté opět spočítána kosinová vzdálenost a ta slova, která jsou nejvíce odlišná (mají vyšší hodnotu kosinové vzdálenosti) jsou označena jako slova klíčová.

5 Analýza datových sad

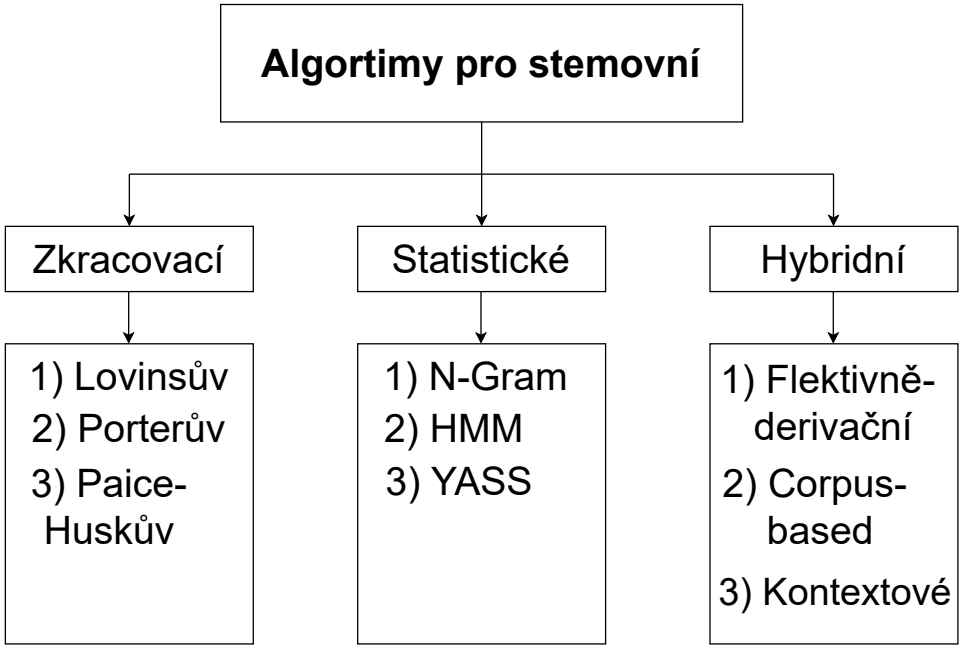
Datové sady určené pro problematiku extrakce klíčových slov jsou v podstatě kolekce dokumentů, kde ke každému dokumentu existuje množina ručně anotovaných klíčových slov. Je ale třeba si uvědomit, že v každém dokumentu se mohou vyskytovat taková slova, která mají jiný tvar, ale stále stejný význam, například jsou v jiném pádu či tvaru. Jednou z možností pro předzpracovaní tedy může být, aby se neidentické termíny se stejnou sémantikou identifikovaly jako jedno slovo. Mezi nejpoužívanější metody pro nalezení jednotného tvaru slov dnes patří lematizace a stemování [11].

5.1 Metody předzpracování

Stemování Stemování je krok předběžného zpracování používaný především při práci s textem jako je získáváni informace z dokumentů či zpracování přirozeného jazyka počítačem. Hlavním cílem je redukovat rozdílné gramatické tvary slov a převést je na kořen, který nese jejich základní význam, tím pádem dochází i ke zmenšení počtu unikátních slov v dokumentu. Princip stemování je obvykle založen na odstranění jakékoli přípony a předpony označeného slova. V závislosti na použitém algoritmu, může určený kořen slova vypadat jinak (v angličtině např. introduce/introduc), důležité však je, aby daný algoritmus určil stejný kořen pro všechny výskyty slova v dokumentu. Stemovací algoritmy lze rozdělit do tří skupin: zkracovací, statistické a hybridní 5.1[9].

Zkracovací algoritmy, jak již napovídá jejich název, pouze upraví dané slovo tak, že odstraní veškeré předpony a přípony. Například jeden z nejjednodušších zkracovacích algoritmů pouze vybírá `n` písmen z právě stemovaného slova a vše ostatní odstraní, v případě, že dané slovo obsahuje méně znaků než je `n`, je zachováno. Statistické algoritmy pro stemování odstraňují přípony po určité statistické proceduře lišící se v závislosti na použitím algoritmu. Hybridní algoritmy jsou poté kombinací zkracovacích a statistických přístupů.

U stemování může dojít ke dvěma chybám a to *over-stemming*, což je situace, kdy nastane k určení stejného kořene dvou slov, které ale stejný kořen nemají, nebo *under-stemming*, což znamená, že dvě slova, která by měla mít stejný kořen nebyla algoritmem vybrána.



Obrázek 5.1: Stemovací algoritmy
Zdroj: `https://www.researchgate.net/publication/284038938_A_Comparative_Study_of_Stemming_Algorithms`

Lematizace Lematizace je podobně jako stemování neodmyslitelnou částí při práci s textem a při zpracování přirozeného jazyka. Dalším společným prvkem lematizace a stemování je jejich cíl, oba se snaží redukovat rozdílné gramatické tvary a převést je na normalizovaný tvar, v případě lematizace tzv *lemma*. Hlavním rozdílem je, že zatímco stemování pracuje s každým jedním slovem zvlášť, lematizace při určování *lemmatu* pracuje i s kontextem, díky tomu dokáže rozpoznat slova s jiným významem. Na druhou stranu je lematizace náročnější na implementaci a je celkově pomalejší než stemování. Lematizace je založena na hledání slovníkového tvaru lematizovaného slova, aby toho bylo možné dosáhnout, *lematizer* musí obsahovat slovník daného jazyka a soubor pravidel, podle kterého dokáže určit *lemma* požadované slovo, nebo si tento slovník musí vytvořit pomocí strojového učení [18]. Mezi nejpoužívanější *lemmatizery* patří WordNet lemmatizer [8] a TextBlob pro angličtinu a pro český jazyk například UDPipe [21].

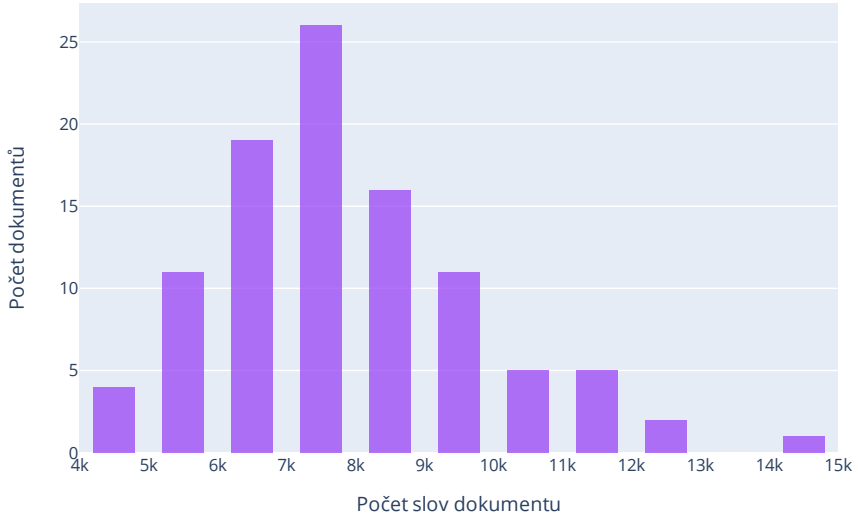
5.2 Existující datové sady

Anglická datová sada Pro otestování metod na anglicky psaných dokumentech bude použita testovací část datové sady, která byla využita na sou-

těži SemEval 2010 úloha číslo 5 [10]. Datová sada obsahuje dokumenty v rozsahu 6 až 8 stránek včetně obrázků, grafů, tabulek a podobně. Dokumenty byly získány ze čtyř různých oborů – distribuované systémy, vyhledávání informací v textu, distribuovaná umělá inteligence a sociální a behaviorální vědy 5.2. Všechny dokumenty mají k sobě i autorem ručně anotovaná klíčová slova, důležitá pro určení úspěšnosti jednotlivých metod.

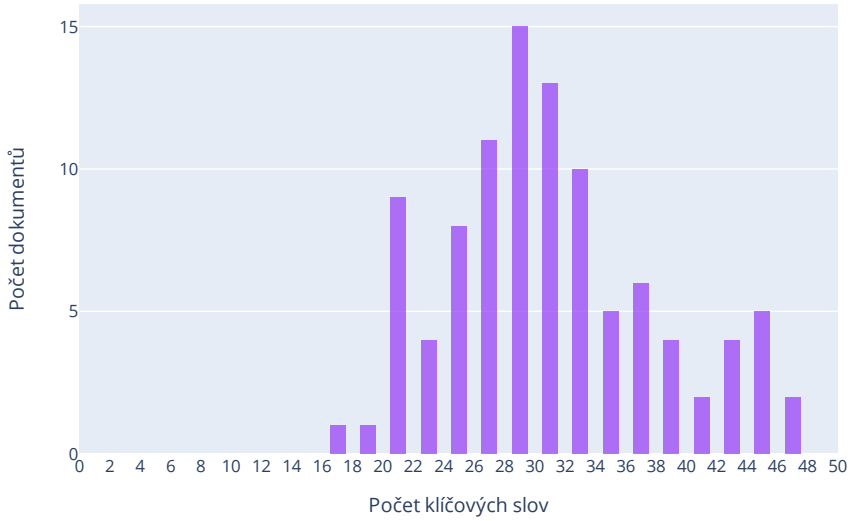
Obor	**Počet dokumentů**
Distribuované systémy	25
Vyhledávání informací v textu	25
Distribuovaná umělá inteligence	25
Sociální a behaviorální vědy	25

Tabulka 5.1: Distribuce oborů článků anglické datové sady ze soutěže SemEval 2010



Obrázek 5.2: Počet slov dokumentů anglické datové sady

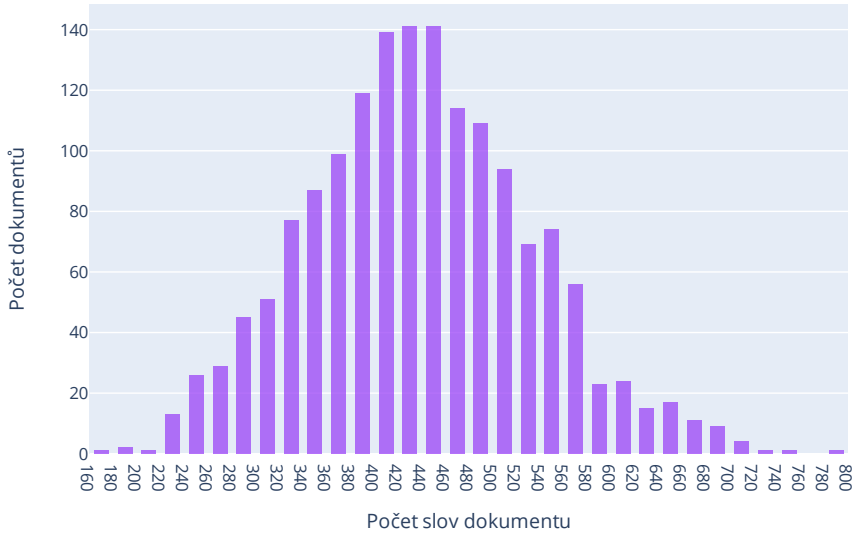
Z histogramu 5.2 lze vyčíst, že největší množství článků v anglické datové sadě je složeno z 7 až 8 tisíc slov. Histogram 5.3 poté ukazuje distribuci anotovaných klíčových slov jednotlivých článků. Anotátoři v soutěži SemEval tedy v každém článku označili 16 až 48 klíčových slov.



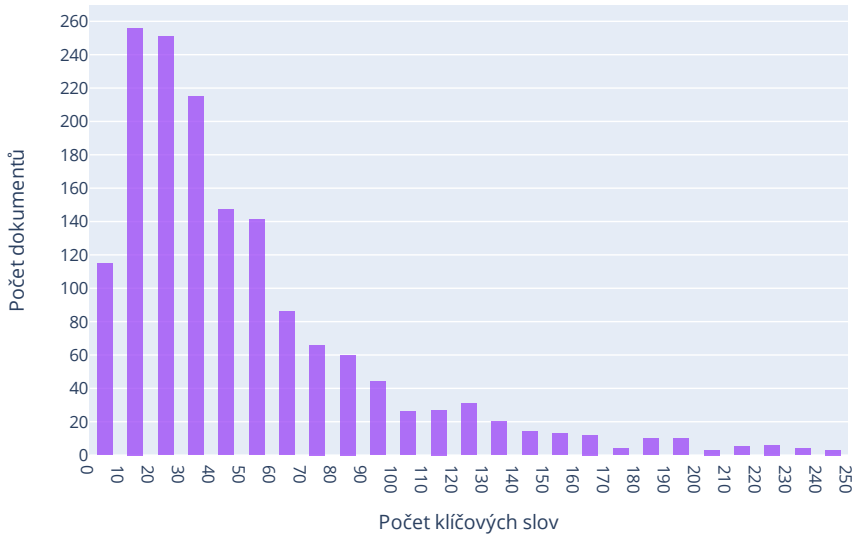
Obrázek 5.3: Počet označených klíčových slov anglické datové sady

Česká datová sada Testování metod pro český jazyk bude probíhat na zcela nové, zatím netestované datové sadě získané při spolupráci s Českou tiskovou kanceláří (ČTK). Datová sada obsahuje 1593 článků z různých oborů s rozsahem 1 až 3 normo-stran. Jelikož dokumenty nebyly anotovány autory, byli požádáni o anotaci korespondujících klíčových slov k dokumentům studenti fakulty sociálních věd Univerzity Karlovy. Data nebyla rozdělena na testovací a trénovací část, čili toto rozdělení bude muset být provedeno v průběhu práce. Jelikož se jedná o první kolo anotování, datová sada není precizně označena. V budoucnu bude uskutečněno ještě druhé kolo s profesionálními anotátory, kde každý článek bude označen více anotátory pro pevnější výslednou datovou sadu.

Podle 5.4 lze soudit, že česká datová sada obsahuje články, které jsou výrazně kratší než je tomu u anglické datové sady 5.2. Dále lze z grafu vyčíst, že největší množství článků obsahuje 380 až 480 slov. Z grafu 5.5 je poté patrné, že distribuce označených klíčových slov je mnohem širší než u anglické datové sady (přibližně 35% článků má anotováno více než 50 slov) a tím pádem neodpovídá přístupu, který byl zvolen na soutěži SemEval 2010, tedy maximálně 50 označených klíčových slov.



Obrázek 5.4: Počet slov dokumentů české datové sady



Obrázek 5.5: Počet označených klíčových slov české datové sady

6 Příprava datových sad

Česká datová sada získaná při spolupráci s ČTK je uložena ve dvou souborech formátu *csv*. V jednom ze souborů se nacházejí jednotlivé články, kde každý článek odpovídá jedné řádce. Druhý soubor poté obsahuje anotovaná klíčová slova pro každý článek. Nejprve bylo tedy zapotřebí konvertovat českou datovou sadu do stejné struktury, jako tomu bylo v soutěži SemEval [10], aby mohly být obě datové sady porovnávány. Výsledná datová sada tudíž musela být v podobě dvou adresářů. Jeden z nich se skládal z textových souborů, kde každý soubor obsahoval pouze jeden článek a ve druhém adresáři se nacházely anotovaná klíčová slova, rovněž v oddělených textových souborech.

Filtrování Jelikož články nebyly anotovány profesionálními anotátory a každý článek označoval jiný člověk, vyskytují se v datové sadě velmi rozdílné přístupy anotování klíčových slov. Cílem filtrování datové sady je odstranit z korpusu takové články, které neodpovídají počtu klíčových slov na článek dle zadání soutěže SemEval. Ve vyfiltrovaných datových sadách jsou tím pádem odstraněny takové články, které mají anotovány více než 50 unikátních klíčových slov, to znamená, že pokud má daný článek anotováno jedno slovo vícekrát (např. v jiném pádě) počítá se jako jedno. Odfiltrovány jsou rovněž takové články, ve kterých je více než jedna třetina označených slov stop slovo.

6.1 Předzpracování datových sad

Připravený korpus byl dále rozdělen na čtyři datové sady podle varianty předzpracování.

1. ČTK raw – Na tuto datovou sadu nebyla aplikována žádná forma předzpracování, obsahuje tedy přesně taková data, jaká byla získána od ČTK.

2. ČTK lemma – U této datové sady, byly všechny články i klíčová slova zlemmatizovány. K lemmatizaci byla použita knihovna UDPipe 2.0 [21]. UDPipe provádí segmentaci vět, tokenizaci, určování slovních druhů, lemmatizaci a analýzu závislostí. Nicméně pro potřeby této práce byla využita pouze funkce lemmatizování.

3. ČTK raw, filtrované – Výsledkem tohoto předzpracování je datová sada zredukovaná o nevyhovující články dle *filtrování*. Všechny články i klíčová slova zůstávají oproti původní datové sadě nezměněné.

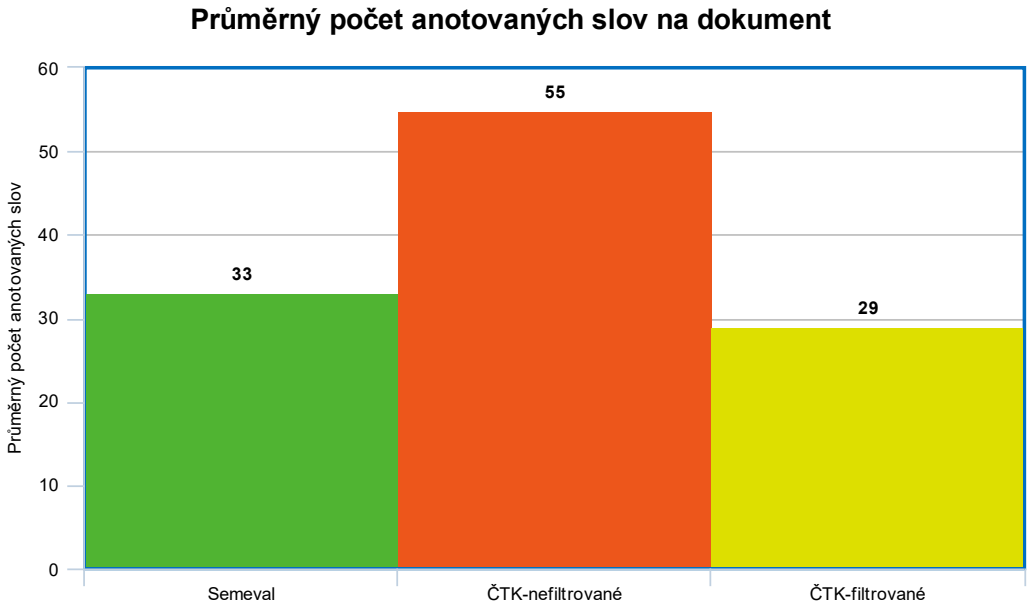
4. ČTK lemma, filtrované – Stejně jako u předchozí metody předzpracování, i zde jsou odfiltrovány stejné články. S tím rozdílem, že zde jsou články a klíčová slova navíc zlemmatizovány.

Z původního datové sady, byly tedy vytvořeny čtyři nové, oddělené korpusy, na kterých probíhaly jednotlivé experimenty. V tabulce 6.1 je porovnání počtu dokumentů jednotlivých datových sad. Lze vidět, že více než 400 článků bylo z české datové sady odstraněno vzhledem k nekonzistentnímu anotování.

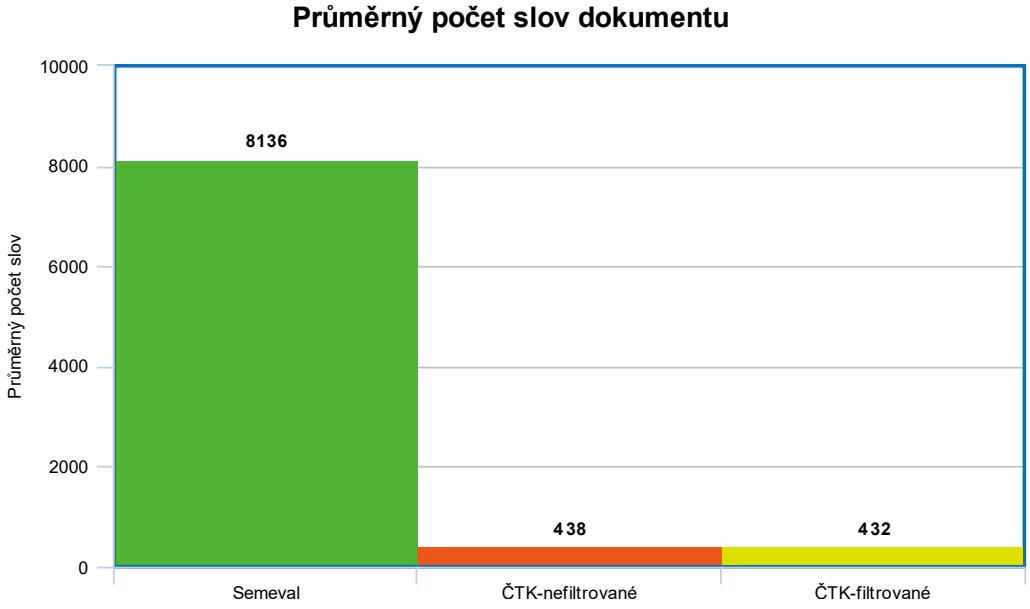
Korpus	Semeval	ČTK-raw	ČTK-lemma	ČTK-raw filtrované	ČTK-lemma filtrované
Celkem	100	1593	1593	1124	1124
Train	80	1274	1274	899	899
Test	20	319	319	225	225

Tabulka 6.1: Počet dokumentů datových sad

Z grafu 6.1 můžeme poté vidět, že v nefiltrované datové sadě se vyskytují dokumenty, které mají anotovány více něž již zmíněných 50 klíčových slov a díky filtrování se počet klíčových slov ve filtrované datové sadě dostává na podobnou hodnotu, jako tomu bylo u soutěže Semeval [10].



Obrázek 6.1: Průměrný počet anotovaných slov na dokument



Obrázek 6.2: Průměrný počet slov dokumentu

Z obrázku 6.2 vyplývá, že anglická datová sada obsahuje markantně delší články než je tomu u české datové sady, což také může mít dopad na výsledky jednotlivých metod. Z obou grafů lze dále vyčíst, že délka dokumentu neovlivňuje počet klíčových slov.

6.2 Trénovací a testovací korpus

Všechny získané datové sady byly následně rozděleny na trénovací a testovací část v poměru 80:20. Aby mohly být datové sady porovnávány, byla použita stejná distribuce, tzn. že v testovací části datové sady ČTK raw se nacházejí stejné články jako v testovací části datové sady ČTK lemma. Trénovací část je určená k natrénování jednotlivých modelů či metod a testovací část slouží k ověření funkčnosti samotných metod a výpočtu jejich metrik.

Nad testovacími částmi vznikly i evaluační soubory, které slouží právě k výpočtu jednotlivých metrik precision, recall a f–measure. Evaluační soubor je uložen ve formátu `tsv`, což je soubor určený k zapisování a čtení málo strukturovaných dat, kde každá řádka reprezentuje jeden záznam, jehož struktura je určena oddělovacím znakem tabulátor. *Tsv* soubor tedy může vypadat nějak takto:

Jméno<TAB>Příjmení
Jan<TAB>Novák
Tonda<TAB>Novotný
Marie<TAB>Nová

V našem případě je každý záznam složen z daného slova a z čísla 1 – klíčové nebo 0 – neklíčové, v závislosti na tom, která slova anotátor označil. Označena jsou rovněž taková slova, jež mají stejné lemma. A jelikož v testovací části je vždy více než jeden soubor (článek), byl zaveden pro oddělení článků oddělovací znak – `###`. Evaluační soubor ve výsledné formě vypadá takto:

WORD<TAB>KEYWORD
D1_Slovo1<TAB>0
D1_Slovo2<TAB>0
D1_Slovo3<TAB>1
D1_Slovo4<TAB>0
###<TAB>###
D2_Slovo1<TAB>0
D2_Slovo2<TAB>1
D2_Slovo3<TAB>1
D2_Slovo4<TAB>1
D2_Slovo5<TAB>0
D2_Slovo6<TAB>0
###<TAB>###

Díky tomuto souboru lze následně vypočítat úspěšnosti jednotlivých metod na daných datových sadách a při budoucích výzkumech může být použit například pro porovnání výsledků. Více k výpočtu metrik se nachází v kapitolách 3.3 a 6.3

6.3 Evaluační skript

Pro určení dílčích metrik byl implementován evaluační skript, který na vstupu očekává dva soubory formátu `tsv` – jeden vytvořený danou metodou a druhý vytvořený z testovací datové sady se správnými odpověďmi. Z obou souborů jsou nejprve odstraněny nežádoucí prvky jako jsou nadpisy sloupců a oddělovací znaky mezi dokumenty. A jelikož první sloupec obou `tsv` souborů, reprezentující jednotlivá slova dokumentů, je identický, evaluační skript z nich získá pouze jejich druhý sloupec, který je složen z 0 a 1. V případě, že první sloupec obou souborů identický není, evaluační skript ohlásí, že metriky nemohou být spočteny. Z těchto dvou vektorů jsou poté za pomoci modulu *sklearn* [17] napočítány samotné metriky – *accuracy*, *precision*, *recall* a *F-score*.

6.4 Vyznačení překryvu anotátorů

V přípravě je druhá česká datová sada, ve které byl zvolen odlišný způsob anotování klíčových slov jednotlivých článků. U této datové sady má na starost označení relevantních slov každého článku více anotátorů. To nabízí možnost filtrovat klíčová slova v závislosti na shodě několika osob. Nad rámec zadání bakalářské práce byl tedy vytvořen skript, jehož výsledkem je vyznačení překryvu anotátorů.

Každý z anotátorů určil množinu slov, které považoval za nejvíce relevantní vzhledem k obsahu celého článku. Skript nejprve postupně načte všechny tyto množiny a následně zjistí počet výskytů jednotlivých slov v množinách s tím, že pokud se v jedné množině dané slovo vyskytuje vícekrát, je počítáno pouze jednou. Na závěr skript projde uživatelem zadaný článek po slovech a v závislosti na tom, kolik anotátorů dané slovo označilo jako klíčové, ho vyznačí barvou ve výsledném souboru formátu *HTML*. Příklad výsledného souboru lze vidět na obrázku 6.3.

Praha 15. prosince (ČTK) - Senát za 20 let své existence dostál v důležitých okamžicích své roli pojistky ústavnosti a demokracie a prokázal své opodstatnění, shodují se politologové s většinou politiků., Po chuti není těm, kteří se s ním dostali do křížku., Patří mezi ně prezident Miloš Zeman či předseda ANO Andrej Babiš., Největší chybou Senátu bylo podle prvního z jeho čtyř předsedů Petra Pitharta (KDU-ČSL) uzákonění přímé volby prezidenta, čímž z něj prý učinil neodvolatelného monarchu., Pithart to řekl na slavnostním shromáždění k 20. výročí ustavující schůze Senátu., Horní komoru hájí mimo jiné premiér Bohuslav Sobotka (ČSSD) jako potřebnou demokratickou pojistku, jejíž zrušení by se mohlo v budoucnu vymstít., Podle předsedy Senátu Milana Štěcha (ČSSD) je baštou politické kultury a hrází, která by mohla v případě potřeby zabránit ovládnutí země extremisty ústavní cestou., Zeman, jemuž Senát odmítl několik kandidátů na ústavní soudce a některé z jeho kroků kritizoval, naposledy veřejně zapochyboval o potřebnosti horní komory v souvislosti s nízkou účastí v letošních senátních volbách., Podobně Babiš mluvil o zbytečnosti Senátu poté, co se mu ve finále senátních voleb nepodařilo zopakovat vítězství z kola prvního., Senát jako ústavní pojistka podle většiny politiků i politologů zafungoval v dobách opoziční smlouvy mezi ČSSD a ODS, ušetřil státu desítky miliard korun za nákup gripenů nebo při snaze o změny zdanění hazardu před pěti lety., Z více než 2000 posuzovaných návrhů zákonů dosud Senát umožnil přijetí 70 procent z nich, téměř 500 předloh vrátil poslancům k novému projednání., Sněmovna vyhověla připomínkám horní komory ve zhruba 310 případech., V senátorských křeslech se za dvě desetiletí vystřídalo 285 lidí., Nejdéle sloužícím senátorem je Štěch, který je členem horní komory od jejího ustavení v roce 1996., Nejkratší senátorství - jen tříměsíční - měl chomutovský lékař Petr Skála, který na mandát musel rezignovat ze zdravotních důvodů.

Legenda

procento anotátorů, kteří označili dané slovo jako klíčové
>= 80% >= 60% >= 40% >= 20%

Obrázek 6.3: Vyznačený překryv anotátorů

7 Experimenty

V této kapitole budou rozebrány všechny experimenty, které v rámci bakalářské práce probíhaly. Testování probíhalo na anglické datové sadě SemEval a české datové sadě získané při spolupráci s ČTK. Byly testovány všechny *baseline* metody i *supervised* metoda založená na *fine tuning* modelů typu BERT.

7.1 Testování baseline

Pro základní porovnání datových sad a jednotlivých přístupů, byly implementovány čtyři metody – TF-IDF (4.1), TextRank (4.2), YAKE! (4.3) a KeyBert (4.4). U metody TF-IDF byla hodnota IDF spočítána na datové sadě, pro kterou bylo testování spuštěno. Pro všechny dokumenty ze zadané datové sady nalezne každá z těchto metod množinu slov, které považuje za klíčové a tato slova zlemmatizuje. Následně projde každý z dokumentů a vyznačí v něm všechny výskyty slov z metodou určené množiny. Výstupem jednotlivých experimentů je tedy soubor formátu `tsv` se stejnou strukturou jako je tomu u evaluačního souboru 6.2, se kterým se následně porovnává. Pokud zadaná datová sada má navíc již daný evaluační soubor vytvořený, jsou výsledkem jednotlivé metriky. Každá z metod byla testována s různými parametry, aby bylo dosaženo co nejlepších výsledků. Jednotlivé metody byly rovněž spuštěny pro všechny datové sady, aby mohly být porovnány výsledky českého i anglického korpusu a aby mohly být posouzeny výsledky předzpracování.

Pro určení hran TextRank využívá společný výskyt slov v určité vzdálenosti definovanou takzvaným okénkem, která dle autorů TextRanku může být nastavena v rozsahu 2 až 10. V tabulce 7.1 jsou zobrazeny výsledky na české i anglické datové sadě v závislosti na velikosti okénka. Jelikož u korpusů je růst metriky *F-score* obrácený, byla ve finálním testování nastavena velikost okénka pro anglickou datovou sadu na hodnotu 10 a u českého korpusu na hodnotu 2. Tato obrácená tendence je pravděpodobně způsobena strukturou stavby vět v obou jazycích. Kvůli tomu, že v angličtině je pevně daná skladba věty, klíčová slova v delším kontextu více vyniknou ve výsledném grafu. Zatímco u českého jazyka, kde je volná stavba vět, se neklíčová slova mohou vyskytovat v jakékoliv části věty a tím pádem jsou počty vazeb všech slov (klíčových i neklíčových) ve výsledném grafu podobné.

U metody KeyBert byly otestovány oba přístupy pro redukování podob-

Velikost okna/Dataset	Semeval (20)	ČTK-raw (319)
N=2	44.76%	**27.64**%
N=5	45.86%	26.92%
N=8	47.05%	25.52%
N=10	**47.29**%	25.37%

Tabulka 7.1: Metrika f-score metody TextRank v závislosti na nastavené velikosti okénka. Číslo v závorce reprezentuje počet dokumentů, pro které metoda určovala klíčová slova.

ných klíčových slov ve výsledné množině – tedy metodika MMR (*Maximal Marginal Relevance*) i metodika MaxSum. Z tabulky 7.2 si můžeme povšimnout, že obě tyto redukční metodiky, které KeyBert nabízí značně zhoršili výslednou F míru. To může být způsobeno tím, že články obvykle obsahují podobná klíčová slova. Například v článku o Evropské unii mohou být klíčová slova: Evropská unie; Evropa; Europoslanec; Europarlament; Eurozóna, kde následně díky těmto redukčním metodám je ve výsledné množině určené jen jedno z těchto slov jako klíčové. I vzhledem k navýšení času (7.3) potřebném pro určení klíčových slov v konečném testování žádná z těchto redukčních metod použita nebyla.

Funkce/Dataset	Semeval (20)	ČTK-raw (319)
KeyBert	**31**%	**27**%
KeyBert_MMR	18%	21%
KeyBert_MaxSum	20%	22%

Tabulka 7.2: Metrika f-score metody KeyBert v závislosti na použité redukční metodě. Číslo v závorce reprezentuje počet dokumentů, pro které metoda určovala klíčová slova.

V tabulce 7.3 je poté zobrazen čas, který metoda potřebovala pro určení množiny klíčových slov. Vyplývá z ní například, že pořadí metod v rychlosti určování je nezávislé na délce dokumentu. Jelikož výsledné pořadí je stejné pro anglický korpus, kde průměrná délka dokumentů se blíží 8000 slov i pro český, kde je průměrná délka přibližně 440 slov. Dále si můžeme všimnout že metody TF-IDF a YAKE! jsou v určování klíčových slov nejrychlejší. Výsledná metrika *F-score* je poté zobrazena v tabulce 8.1.

7.2 Návrh nové metody

Díky tomu, že při spolupráci s ČTK byla získána dostatečně velká datová sada, může nová metoda využívat *supervised* přístup. To není úplně obvyklé

Metoda/Dataset	Semeval (20)	ČTK (319)	ČTK-filtrované (225)
TF-IDF	0:00:17	0:00:17	0:00:12
TextRank	0:00:40	0:01:11	0:01:01
YAKE!	0:00:22	0:00:48	0:00:21
KeyBert	0:00:53	0:03:19	0:02:54
KeyBert_MMR	0:02:16	0:04:36	0:04:08
KeyBert_MaxSum	0:33:08	0:41:54	0:31.12

Tabulka 7.3: Časy běhů jednotlivých metod. Číslo v závorce reprezentuje počet dokumentů, pro které metoda určovala klíčová slova.

u metod zabývajících se extrakcí klíčových slov, právě z důvodu nedostatku trénovacích dat. Cílem nové metody bylo pokusit se překonat výsledky implementovaných metod tzv. *baseline* a dokázat, že využití BERT modelů je pro extrakci klíčových slov výhodné i přes neúspěch KeyBert.

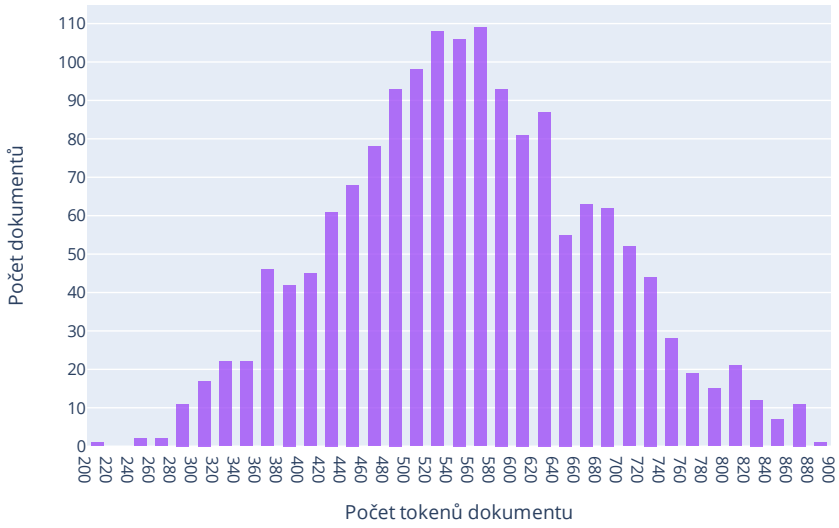
7.2.1 Metoda založená na BERT modelu

Nová metoda tedy využívá algoritmus učení s učitelem (*supervised*) a je založená na *fine tuning* BERT modelu, respektive Czert [20] modelu. To znamená, že model je přetrénován pro problematiku extrakce klíčových slov v textu. Metoda používá model *BertForTokenClassification*, který je součástí knihovny Transformers od HuggingFace [25]. Tento model umožňuje provádět klasifikaci na úrovni tokenu.

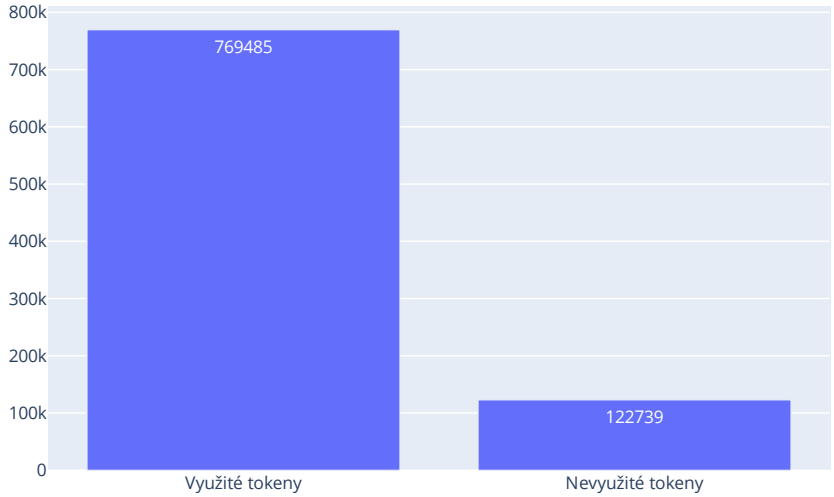
Nejprve bylo zapotřebí načíst samotné dokumenty a k nim anotátorem označená klíčová slova do slovníků. Jeden slovník tedy obsahuje soubor slov jednotlivých článků a ve druhém slovníku se nachází množina 0 a 1, značící, zda odpovídající slovo je klíčové či nikoli. Jelikož jsou modely typu BERT založeny na *WordPiece* tokenizaci, je zapotřebí převést všechna slova na tokeny za pomoci tokenizéru. Zde je důležité udržet mapování jednotlivých *labelů*. To znamená, že pokud dané slovo bylo tokenizérem rozděleno na více tokenů, je třeba jednotlivé části slov označit stejně jako tomu bylo u původního slova.

Po získání tokenů a k nim správně namapovaných *labelů* byly jednotlivé tokeny převedeny na jejich ID, aby s nimi mohla pracovat neuronová síť. Vstupní sekvence do BERT modelu byla nastavena na délku 512 tokenů. V případě, že jich daný článek obsahoval více než 512, byly tokeny, které tuto hodnotu přesáhli odstraněny. Na druhou stranu pokud článek obsahoval méně než 512 tokenů, byly doplněny hodnotou 0. Histogram 7.1 zobrazuje distribuci počtu tokenů dokumentů. Lze vidět, že více než u poloviny článku bylo zapotřebí nějaké tokeny odebrat a přibližně u jedné třetiny bylo

nutné doplnit hodnotu 0. Graf 7.2 dále ukazuje, že celkem bylo odebráno (nevyužito) 13,76% tokenů.



Obrázek 7.1: Počet tokenů dokumentu



Obrázek 7.2: Počet využitých a nevyužitých tokenů

Dále bylo zapotřebí připravit model typu *BertForTokenClassification*, a zatížit jej předem připravenými váhami modelu *UWB-AIR/Czert-B-base-cased* [20]. Jediné, co muselo být dodatečně specifikováno byl počet klasifikovaných tříd – v našem případě dvě, zda je slovo klíčové či nikoli.

Pro ověření kvality učení modelu, byla vstupní data rozdělena na dvě části – trénovací a validační v poměru 80:20. Při trénování a evaluaci modelu byly tedy využity tři části datové sady, a to *train* – trénovací data (80% z původní datové sady); *test* – evaluační (gold) část (20% z původní datové sady); a *dev* – část pro testování v průběhu tréninku (20% z trénovací části). Poté už se daný model trénoval po několik epoch, více k experimentům se nachází v dalších sekcích.

Výsledkem je tedy model, který byl pomocí *fine tuning* trénován pro problematiku extrakce klíčových slov. Na závěr metoda pomocí daného modelu vytvoří stejný `tsv` soubor, jako tomu bylo u metod *baseline* na stejné testovací datové sadě.

7.2.2 Nastavení hyperparametrů

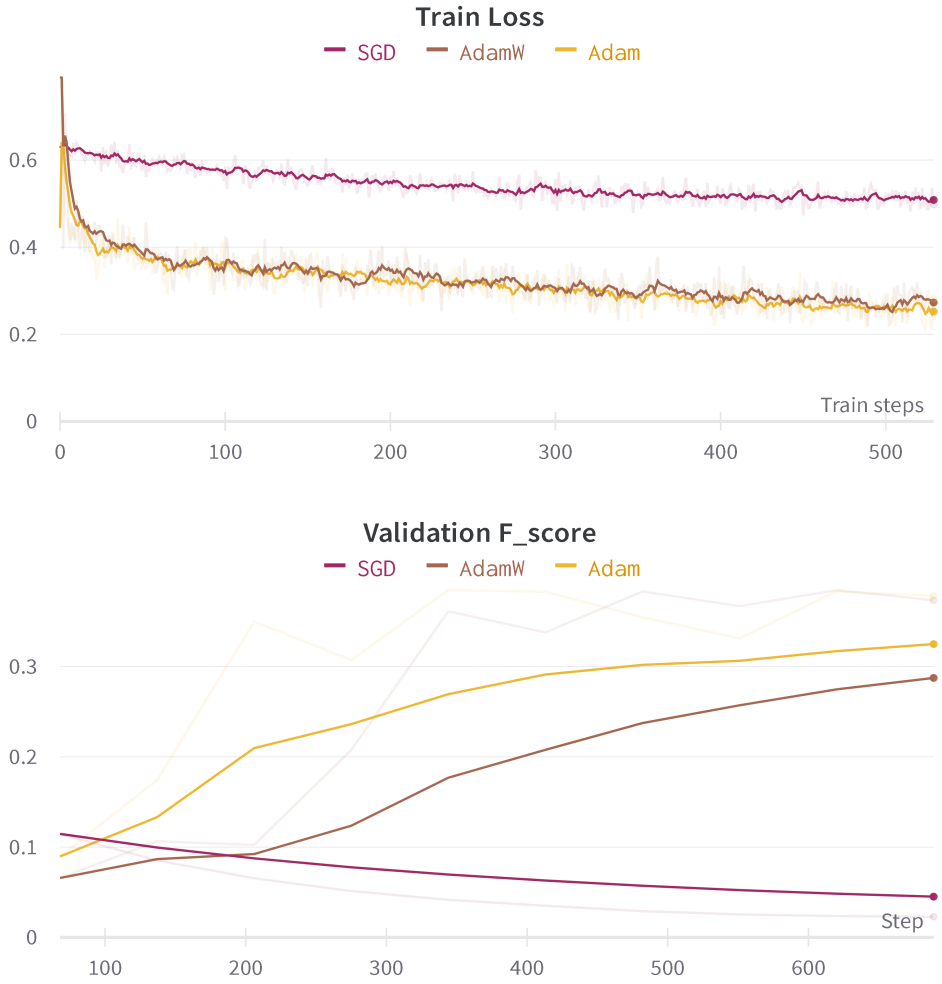
Pro nalezení optimálních hyperparametrů neuronové sítě byl zvolen experimentální přístup, kdy jsou ručně nastaveny počáteční hodnoty a na základě

výsledku je určen optimální hyperparametr. Cílem tohoto přístupu je postupné nalezení závislostí a minimalizace nevhodných parametrů. Všechny experimenty byly spuštěny opakovaně, aby nedošlo k chybnému posouzení na základě jednoho výsledku. Experimenty spojené s trénováním *supervised* metody probíhaly na Metacentru, což je jedna z virtuálních organizací české Národní Gridové Iniciativy MetaCentrum NGI a je otevřená všem akademickým pracovníkům a studentům členů sdružení CESNET.

Dle původní studie modelů typu BERT [4] by měla při *fine tuning* většina hyperparametrů zůstat podobná jako při původním tréninku BERT modelu. Výjimkou jsou hyperparametry *batch size*, *learning rate*, *optimizer* a počet trénovacích epoch.

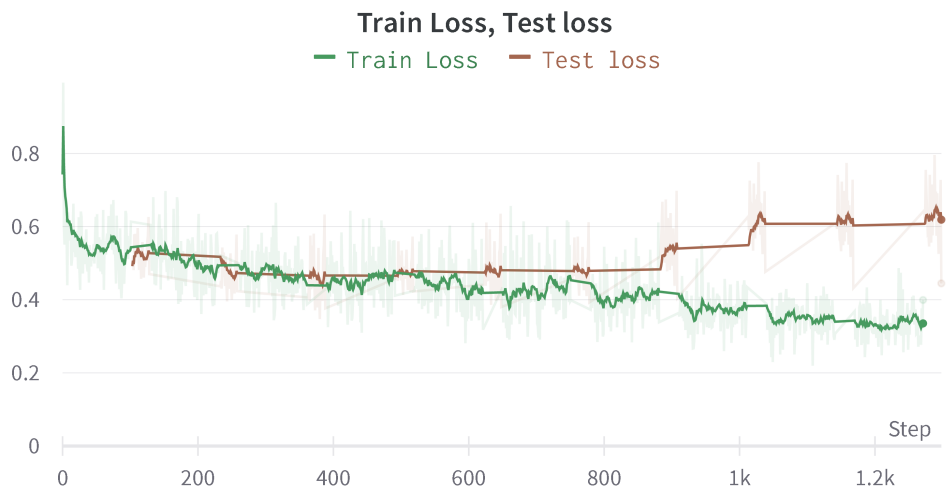
Prvním hledaným hyperparametrem byl *optimizer*. Optimizer je funkce nebo algoritmus, který upravuje atributy neuronové sítě, jako jsou její váhy nebo rychlost učení. Pomáhá tedy redukovat celkovou ztrátu a zlepšit přesnost. Problém výběru správných vah pro daný model je obtížný úkol, protože model hlubokého učení se obecně skládá z milionů parametrů. To vyvolává potřebu zvolit vhodný optimalizační algoritmus pro danou aplikaci. Mezi testovanými optimizery byly *SGD*, *Adam* a *AdamW*. Při hledání nejvýhodnějšího optimizeru zůstaly ostatní hyperparametry, kromě batch size, stejné jako u původní studie [4] – tedy learning rate: 3e-5, počet epoch: 3 a hyperparametr batch size byl nastaven na hodnotu 10 z důvodu výpočetní náročnosti.

Na obrázku 7.3 lze vidět chybovost při trénování modelu a *F-score* pro jednotlivé optimizery. *F-score* byla určena na *dev* části trénovacího korupusu, čili na 20% z trénovací části, která při tréninku použita nebyla. Z obou grafů vyplývá že jednoznačně nejhorší výsledky má optimizer SGD, na druhou stranu zbylé dva optimizery mají velmi podobné výsledky. Nepatrně lepších výsledku dosáhl optimizer Adam, ale vzhledem k *F-score* spočítané na testovací (*gold*) části korpusu, kde AdamW dosáhl pokaždé o 1% až 2% lepších výsledků, byl jako nejvhodnější optimizer vybrán AdamW, se kterým byly následně prováděny další experimenty.



Obrázek 7.3: Chybovost při tréninku a F-score použitých optimizerů

Z obrázku 7.4 si můžeme poté povšimnout, že při trénování po více epoch u modelu dochází k tzv. *overfittingu*. Overfitting je v podstatě přeučení modelu pro trénovací část dat. To znamená, že se model snaží naučit příliš mnoho detailů z trénovacích dat. V důsledku toho je výkon modelu u testovacích datových sad velmi nízký. Síť tedy nedokáže zobecnit vlastnosti nebo vzory přítomné v trénovací datové sadě.



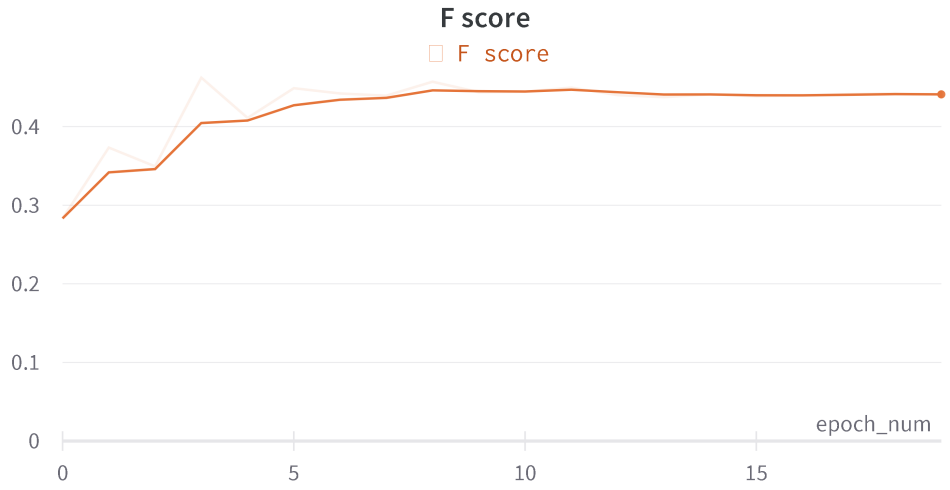
Obrázek 7.4: Porovnání chybovosti na trénovací a dev části korpusu

Aby k *overfittingu* nedocházelo aplikuje se na model tzv. regularizace, která může mít několik podob. V případě této práce byla použita tzv. *dropout* funkce. Aplikováním *dropout* na vrstvy neuronové sítě, model ignoruje podmnožinu jednotek sítě s nastavenou pravděpodobností, která je v našem případě 20%. *Dropout* funguje tak, že náhodně nastaví výstupy skrytých jednotek (neuronů, které tvoří skryté vrstvy) na 0 při každé aktualizaci tréninkové fáze. Díky tomu lze omezit vzájemné učení mezi jednotlivými neurony, což mohlo vést k *overfittingu*. Na druhou stranu při použití *dropout* je občas zapotřebí model trénovat po více epoch k důkladnému naučení. Na obrázku 7.5 je poté vidět vývoj chybovosti při použití regularizace.

Obrázek 7.5: Porovnání chybovosti na trénovací a *dev* části korpusu s regularizací.

Dalším testovaným hyperparametrem byl *learning rate*. Podle původní studie BERT modelů [4] by se měl při *fine tuning* rozsah tohoto parametru pohybovat v rozmezí 5e-5 až 2e-5. Trénování modelu probíhalo postupně se všemi těmito hodnotami. Výsledná chybovost všech tří hodnot byla podobná, rozdíl mezi krajními hodnotami jednotlivých nastavení bylo 8%. Hlavním rozdílem nebyla tedy výsledná chybovost ale doba, po kterou se model trénoval. Jako finální hodnota hyperparametru *learning rate* byl nakonec zvolen kompromis mezi dobou trénování a výslednou chybovostí, tedy hodnota 3e-5.

Posledním hyperparametrem pro nastavení je počet epoch. Dle autorů BERT [4] stačí pro *fine tuning* modelů typu BERT 2 až 4 epochy. Nicméně experimentálně byl vyzkoušen trénink až po 20 epoch a byla pozorována metrika *F-score* na testovací části datové sady. Na obrázku 7.6 je viditelné, že přibližně po desáté epoše se *F-score* pro testovací gold data již nezvyšuje, proto bylo u dalších experimentů zvoleno pro trénování 10 epoch.

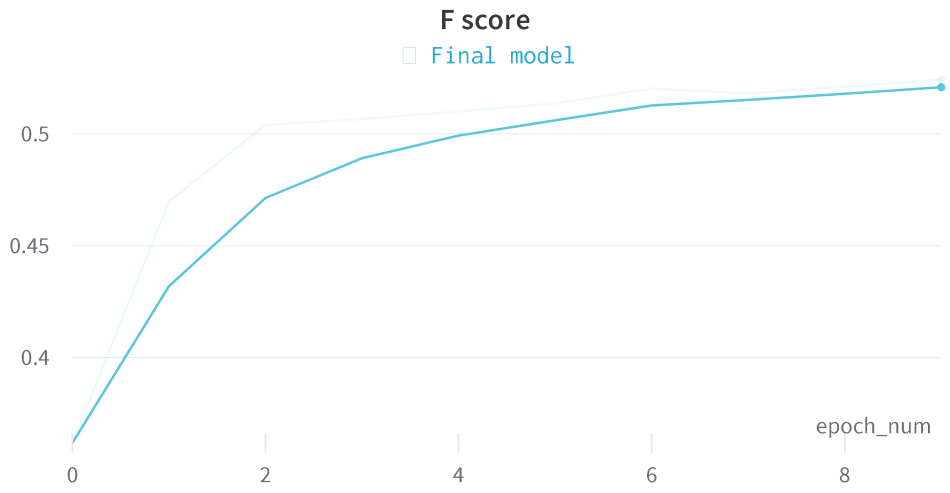


Obrázek 7.6: Vývoj metriky F-score po jednotlivých epochách

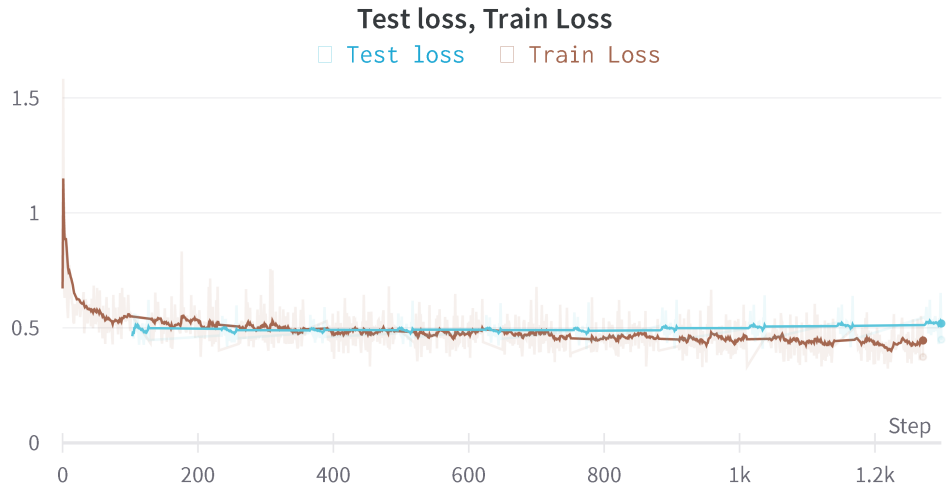
Devlin (2018) v jeho původní studii BERT modelů [4] doporučuje při *fine tuning* měnit pouze hyperparametry learning rate, počet epoch, optimizer a batch size. V této *supervised* metodě zabývající se problematikou automatické extrakce klíčových slov byly nakonec jako optimální hyperparametry zvoleny následující hodnoty:

- optimizer – AdamW
- learning rate – 3e-5
- batch size – 10
- počet epoch – 10

S těmito hyperparametry byl model 8 krát nezávisle trénován a byl vypočten výsledný 95% interval spolehlivosti pro *F-score* na testovací části všech vytvořených českých datových sad 8.1. Vývoj metriky *F-score* takto natrénovaného modelu je znázorněn na obrázku 7.7. Graf 7.8 poté zobrazuje vývoj chybovosti na trénovací i dev části datové sady výsledného modelu. Jeden průběh *fine tuning* s tímto nastavením zabere přibližně 20 minut času pro nefiltrovanou datovou sadu a 15 minut pro filtrovanou, testování probíhalo na clusteru *adan*.



Obrázek 7.7: Vývoj metriky F-score po jednotlivých epochách výsledného modelu



Obrázek 7.8: Porovnání chybovosti na trénovací a *dev* části korpusu výsledného modelu

8 Diskuze

V následující tabulce lze nalézt všechny výsledky jednotlivých metod v závislosti na testované datové sadě. Pro anglickou datovou sadu nebyla aplikována *supervised* metoda založená na *fine tuning* BERT modelu z důvodu absence trénovacích dat.

Metoda/Dataset	Semeval (80 + 20)	ČTK-raw (1274 + 319)	ČTK-lemma (1274 + 319)	ČTK-raw filtrované (899 + 225)	ČTK-lemma filtrované (899 + 225)
TF-IDF	33%	45%	51%	35%	43%
TextRank	**47%**	28%	33%	22%	26%
YAKE!	31%	33%	45%	30%	39%
KeyBert	30 %	27%	34%	22%	26%
Czert fine tuning	—	**48.28% ±0.22**	**52.34% ±0.18**	**45.97 ±0.43**	**50.44% ±0.32**

Tabulka 8.1: Metrika F score jednotlivých datových sad a metod spočtená na testovací (gold) části datových sad. Číslo v závorkách reprezentuje počet trénovacích respektive testovacích dat (train + test)

Z tabulky 8.1 lze vidět, že nejlepší *unspuervised* metoda pro všechny české datové sady je jednoznačně TF-IDF, na rozdíl oproti anglické datové sadě, kde vládne TextRank, který u českého korpusu příliš neuspěl. Veliký rozdíl v závislosti na datové sadě, který se u TextRank vyskytuje je pravděpodobně způsoben markantním rozdílem v rozsahu dokumentů v anglické a české datové sadě. TextRank určuje klíčová slova na základě počtu výskytů slova v odlišném kontextu. Tudíž v dokumentu, který obsahuje například pouze 300 výrazů, se počet rozdílných kontextů klíčového slova nemusí tolik lišit od počtu kontextů slova neklíčového.

Na druhou stranu nejméně uspokojivé výsledky má metoda založená na kosinové podobnosti – KeyBert. Ten má porovnatelně horší výsledky než ostatní metody nezávisle na datové sadě. To může být způsobeno tím, že KeyBert je založen na tzv. *sentence-transformers*, který na vstupu očekává množinu vět. Avšak KeyBert na vstup *sentence-transformers* posílá obsah celého dokumentu jako jednu větu. *Sentence-transformers* byl původně navržen pro porovnání sémantické podobnosti celých vět, a při použití pro problematiku extrakce klíčových slov nedosahuje takových výsledků.

Celkovým vítězem pro českou datovou sadu je *supervised* metoda založená na *fine tuningu* modelů typu BERT. Metody extrakce klíčových slov nebývají často řešeny pomocí *supervised* přístupů, protože obvykle nebývá dostatek trénovacích dat. Nicméně díky těmto výsledkům je vidět, že *supervised* metoda dominuje i přes to, že získaná datová sada by mohla mít lepší podobu. Například v současné době připravovaná druhá datová sada, ve

které bude každý článek označen více anotátory. Díky tomu lze získat mnohem přesnější testovací data na základě překryvu jednotlivých anotátorů, tedy označit slovo jako klíčové jen v případě, že se na něm shodlo určité procento osob. Na druhou stranu je potřeba si položit otázku, zda výsledky *supervised* metody jsou natolik odlišné od ostatních *unsupervised* přístupů, aby se vyplatilo ručně anotovat velké množství článků.

Dále je patrné, že po provedení filtrování na základě počtu klíčových slov se veškeré metody potýkají se zhoršením výsledků. To je pravděpodobně způsobeno tím, že v originální české datové sadě, se nachází spousta článků, které mají anotováno více než je běžný počet klíčových slov (viz 6.1), čili jako klíčová slova jsou označena i slova, která ve skutečnosti klíčová nejsou (například stop slova). Ku příkladu pokud má článek skládající se ze 400 slov anotátorem označeno 200 slov jako klíčových, má metoda výrazně vyšší šanci, že „správně“ určí klíčové slovo než u stejně dlouhého článku, který má anotováno slov 50.

V neposlední řadě si můžeme povšimnout, že datové sady využívající při předzpracování lemmatizaci si vedou podstatně lépe než datové sady, které nikoliv. Děje se tak proto, že v českém jazyce se nachází spousta tvarů slov (skloňování, časováni apod.) a po lemmatizaci jsou tyto tvary převedeny na jednu stejnou podobu. Pro metodu je poté jednodušší určit správné klíčové slovo. Nicméně u metody založené na *fine tuning* BERT modelů nebylo toto chování očekávané, protože model Czert [20], se kterým byly prováděny všechny experimenty, byl původně trénován na nezlemmatizovaném textu. Toto chování je pravděpodobně způsobeno nedostatečně velkou trénovací částí. V případě, že by existovalo více trénovacích dat byl by efekt nejspíše obrácený, tedy korpus, který nevyužívá lemmatizaci by dosáhl lepších výsledků.

9 Závěr

Na začátku této práce byly vysvětleny základní pojmy z oblasti extrakce klíčových slov z textu. Byla představena existující datová sada obsahující anglicky psané dokumenty, která byla použita i při soutěži SemEval 2010, představena byla rovněž nově vzniklá, nikdy netestovaná datová sada v českém jazyce získaná při spolupráci s Českou tiskovou kanceláří. Dále byly analyzovány jednotlivé přístupy pro řešení problematiky získávání klíčových slov a z každé kategorie byla jedna metoda vybrána: TF-IDF, TextRank, YAKE! a KeyBert a tyto přístupy byly následně podrobně vysvětleny.

Dále byla česká datová sada transformována do stejné podoby, jako tomu bylo u anglické datové sady a následně byly vytvořeny čtyři nové korpusy na základě použitého typu předzpracování. Podle nastudovaných algoritmů byly vyjmenované metody implementovány a následně otestovány na obou datových sadách. Díky velikosti korpusu byla následně vytvořena *supervised* metoda založená na *fine tuning* modelů typu BERT.

V rámci experimentů byla otestována tato navržená metoda na české datové sadě a byly nalezeny optimální hodnoty jednotlivých hyperparametrů pro *fine tuning* modelu. S touto konfigurací byl model natrénován a následně porovnán s ostatními přístupy. Tento *supervised* postup se ukázal jako velice efektivní a na všech českých datových sadách, nezávisle na typu předzpracování, překonal všechny *unsupervised* metody. Kvůli nedostatku trénovacích dat u anglického korpusu tato metoda nebyla na této datové sadě testována.

Vzhledem k tomu, že je v přípravě druhá česká datová sada, která bude umožňovat vytvoření testovacích dat na základě překryvu anotátorů, mohl by být tento *supervised* přístup v budoucnu otestován na tomto korpusu a dosáhnout přesněji naměřených výsledků.

Závěrem lze říci, že metoda využívající přístup učení s učitelem překonala metody využívající učení bez učitele. Na druhou stranu získání dostatku trénovacích dat je velmi časově i finančně náročné. Dalším rozšířením práce by tedy mohlo být vylepšení metody KeyBert, která je *unsupervised* a zároveň využívá BERT modely, které se ukázaly jako velmi efektivní pro problematiku extrakce klíčových slov.

Literatura

[1] Ali, K. – Zahedi. An Efficient Approach for Keyword Selection ; Improving Accessibility of Web Contents by General Search Engines. *International journal of Web & Semantic Technology*. 10 2011, 2, s. 81–90. doi: 10.5121/ijwest.2011.2406.

[2] BELIGA, S. Keyword extraction: a review of methods and approaches. *University of Rijeka, Department of Informatics*. 2014, s. 1–9. doi: 10.1.1.704.9230. Dostupné z: `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.704.9230&rep=rep1&type=pdf`.

[3] Campos, R. et al. YAKE! Keyword extraction from single documents using multiple local features. *Information Sciences*. 2020. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2019.09.013. Dostupné z: `https://www.sciencedirect.com/science/article/pii/S0020025519308588`.

[4] Devlin, J. et al. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*. 2018.

[5] Florescu, C. – Caragea, C. PositionRank: An Unsupervised Approach to Keyphrase Extraction from Scholarly Documents. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, s. 1105–1115, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1102. Dostupné z: `https://aclanthology.org/P17-1102`.

[6] Grootendorst, M. KeyBERT: Minimal keyword extraction with BERT., 2020. Dostupné z: `https://doi.org/10.5281/zenodo.4461265`.

[7] *Getting started with keyword extraction. text mining online.* [online]. 2021. [cit. 2021/10/17]. Dostupné z: `http://textminingonline.com/getting-started-with-keyword-extraction,2014`.

[8] Jayakodi, K. et al. WordNet and Cosine Similarity based Classifier of Exam Questions using Bloom's Taxonomy. *International Journal of Emerging Technologies in Learning*. 2016, 11, 4.

[9] Jivani, A. G. – others. A comparative study of stemming algorithms. *Int. J. Comp. Tech. Appl*. 2011, 2, 6, s. 1930–1938.

[10] Kim, S. N. et al. Semeval-2010 task 5: Automatic keyphrase extraction from scientific articles. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, s. 21–26, 2010.

[11] Korenius, T. et al. Stemming and Lemmatization in the Clustering of Finnish Text Documents. In *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management*, CIKM '04, s. 625–633, New York, NY, USA, 2004. Association for Computing Machinery. doi: 10.1145/1031171.1031285. Dostupné z: https://doi.org/10.1145/1031171.1031285. ISBN 1581138741.

[12] Liu, F. et al. *Unsupervised Approaches for Automatic Keyword Extraction Using Meeting Transcripts*. Association for Computational Linguistics, 2009. ISBN 9781932432411.

[13] Machado, D. et al. Universal Mobile Information Retrieval. In Stephanidis, C. (Ed.) *Universal Access in Human-Computer Interaction. Intelligent and Ubiquitous Interaction Environments*, s. 345–354, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. ISBN 978-3-642-02710-9.

[14] MIHALCEA, P. R. T. Textrank: Bringing order into text. *Proceedings of the 2004 conference on empirical methods in natural language processing*. July 2004, s. 404–411. Dostupné z: https://aclanthology.org/W04-3252.pdf.

[15] Nasar, Z. – Jaffry, S. W. – Malik, M. K. Textual keyword extraction and summarization: State-of-the-art. *Information Processing & Management*. 2019, 56, 6, s. 102088. ISSN 0306-4573. doi: https://doi.org/10.1016/j.ipm.2019.102088. Dostupné z: http://doi.acm.org/10.1145/366622.366644.

[16] Page, L. et al. The PageRank Citation Ranking: Bringing Order to the Web. Technical Report 1999-66, Stanford InfoLab, November 1999. Dostupné z: http://ilpubs.stanford.edu:8090/422/. Previous number = SIDL-WP-1999-0120.

[17] Pedregosa, F. et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*. 2011, 12, s. 2825–2830.

[18] Plisson, J. et al. A rule based approach to word lemmatization. In *Proceedings of IS*, 3, s. 83–86, 2004.

[19] Rose, S. et al. Automatic keyword extraction from individual documents. *Text mining: applications and theory*. 2010, 1, s. 1–20.

[20] Sido, J. et al. Czert–Czech BERT-like Model for Language Representation. *arXiv preprint arXiv:2103.13031*. 2021.

[21] Straka, M. UDPipe 2.0 Prototype at CoNLL 2018 UD Shared Task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from*

Raw Text to Universal Dependencies, s. 197–207, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/K18-2020. Dostupné z: https://www.aclweb.org/anthology/K18-2020.

[22] TURNEY, P. D. Learning to extract keyphrases from text. *National Research Council Canada. Institute for Information Technology*. 1999. Dostupné z: https://arxiv.org/ftp/cs/papers/0212/0212013.pdf.

[23] VOITA, E. et al. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. *arXiv preprint arXiv:1905.09418*. 2019.

[24] WITTEN, I. et al. KEA: Practical automatic keyphrase extraction. *ACM DL*. 1999. doi: 10.1145/313238.313437. Dostupné z: https://doi.org/10.1145/313238.313437.

[25] WOLF, T. et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*. 2019.

[26] ZHANG, C. Automatic keyword extraction from documents using conditional random fields. *Journal of Computational Information Systems*. 2008, 4, 3, s. 1169–1180.

[27] ZHANG, Y. – JIN, R. – ZHOU, Z.-H. Understanding bag-of-words model: a statistical framework. *International Journal of Machine Learning and Cybernetics*. 2010, 1, 1-4, s. 43–52.

Přílohy

Popis adresářové struktury

- **Aplikace_a_knihovny** – obsahuje všechny vytvořené skripty v jazyce python a další potřebné soubory ke spuštění. Obsahuje rovněž README s návodem pro spuštění.
- **Text_prace** – Obsahuje všechny zdrojové soubory dokumentace.
- **Vstupni_data** – Obsahuje tři adresáře s datovými sadami
  - **CTK** – obsahuje českou datovou sadu určenou pro extrakci klíčových slov
  - **CTK-prekryv** – obsahuje českou datovou sadu s překryvem anotátorů
  - **SemEval2010** – obsahuje anglickou datovou sadu určenou pro extrakci klíčových slov
- **Výsledky** – obsahuje adresář s jednotlivými běhy skriptů