FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Assignment of bachelor's thesis

Title:	Topic Modeling for Corpus of Czech Verse
Student:	Anna Tesaříková
Supervisor:	Ing. Magda Friedjungová, Ph.D.
Study program:	Informatics
Branch / specialization:	Knowledge Engineering
Department:	Department of Applied Mathematics
Validity:	until the end of summer semester 2022/2023

Instructions

The aim of this thesis is to analyze and apply methods of topic modeling with focus on poems to extract themes and motifs from a corpus of Czech poetry. This corpus is publicly available and contains 1305 books (annotated poetic meters, rhymes, tokenized, lemmatized, etc.) of Czech poetry from the 19th and the beginning of the 20th century.

1) Get familiar with the Corpus of Czech Verse publicly available at https://github.com/versotym/corpusCzechVerse/ and with specifics of NLP for the Czech language in general.
2) Survey state of the art methods and tools for poetic topic modeling including possible word embeddings.
3) Apply selected methods (including those in references [1] and [2] below) on the Corpus of Czech Verse, extract a set of representative poetic "topics" and evaluate results.

References

[1] Plechac, Haider: Mapping Topic Evolution Across Poetic Traditions. 2020.
[2] Devlin et al.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 2018.

Electronically approved by Ing. Karel Klouda, Ph.D. on 26 October 2021 in Prague.

FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Bachelor's thesis

Topic Modeling for Corpus of Czech Verse

Anna Tesaříková

Department of Applied Mathematics
Supervisor: Ing. Magda Friedjungová, Ph.D.

May 10, 2022

Acknowledgements

I would like to thank my thesis supervisor Ing. Magda Friedjungová, Ph.D. for guidance during writing this thesis and my three cats for support.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended. In accordance with Article 46 (6) of the Act, I hereby grant a nonexclusive authorization (license) to utilize this thesis, including any and all computer programs incorporated therein or attached thereto and all corresponding documentation (hereinafter collectively referred to as the "Work"), to any and all persons that wish to utilize the Work. Such persons are entitled to use the Work in any way (including for-profit purposes) that does not detract from its value. This authorization is not limited in terms of time, location and quantity. However, all persons that makes use of the above license shall be obliged to grant a license at least in the same scope as defined above with respect to each and every work that is created (wholly or in part) based on the Work, by modifying the Work, by combining the Work with another work, by including the Work in a collection of works or by adapting the Work (including translation), and at the same time make available the source code of such work at least in a way and scope that are comparable to the way and scope in which the source code of the Work is made available.

In Prague on May 10, 2022 . . . . . .

Czech Technical University in Prague
Faculty of Information Technology

Citation of this thesis

Tesaříková, Anna. *0.0.0.* Bachelor's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2022.

Abstrakt

Tato bakalářská práce zkoumá současné metody pro modelování témat s důrazem na jejich využití v modelování témat poezie. Metody jsou aplikovány na Korpus českého verše. Tento korpus obsahuje 1 305 básnických sbírek z 19. a počátku 20. století, které jsou lemmatizované a foneticky, morfologicky, metricky a stroficky anotované. Práce vyhodnocuje výsledky jednotlivých metod a vzájemně je porovnává.

Klíčová slova modelování témat, poezie, český jazyk, zpracování přirozeného jazyka, strojové učení

Abstract

This bachelor thesis surveys state-of-the-art methods of topic modeling with an emphasis on their use in topic modeling of poetry. The methods are applied on the Corpus of Czech Verse. This corpus contains 1,305 collections of poetry from the 19th and the beginning of 20th century which are lemmatized and annotated phonetically, morphologically, metrically, and strophically. The thesis evaluates the results of the methods and compares them with each other.

Keywords topic modeling, poetry, Czech language, natural language processing, machine learning

Contents

List of Figures

List of Tables

Introduction

Topic modeling is an important field of machine learning and natural language processing; it provides methods to organize, understand and summarize large collections of data. Its aim is to automatically extract abstract topics from collections of documents. It is a field of unsupervised machine learning; it does not require training data which have been previously classified by humans.

The techniques of topic modeling are usually used with text data, especially with non-fiction scientific texts; this work focuses on use of topic modeling methods for extracting topics from poetry. The dataset used in this thesis is the Corpus of Czech Verse which is being built at the Institute of Czech Literature of the Czech Academy of Sciences.

The goal of this thesis is to survey state-of-the-art methods of topic modeling, to apply them on the Corpus of Czech Verse and to evaluate and discuss the results. The thesis consists of three chapters. In chapter 1, the theoretical background of topic modeling is described. In chapter 2, the Corpus of Czech Verse dataset is described as well as the implementation of topic modeling methods, experiments and their evaluation. In chapter 3, the performance and results of different methods are discussed.

This thesis is written with the assumption that the reader has a working knowledge of machine learning. For more information about machine learning, see e.g. K. P. Murphy. Machine learning: a probabilistic perspective. MIT press, 2012.

Chapter 1

Theoretical background

In this chapter, I will explain the basic terms of topic modeling and introduce the reader to existing research in this area. I will describe the state-of-the-art methods of topic modeling such as the specifics of Czech language processing.

1.1 Topic Modeling

Topic modeling is a statistical method for extracting the abstract topics from collections of documents. It is well suited for use with text data; however, it also has applications in other fields such as bioinformatics, social sciences or computer vision [1].

Topic modeling was originally developed in the 1980's as a part of the subject area of generative probabilistic modeling [1]. This type of modeling describes how observed variables interact with unobserved parameters in terms of a probabilistic model. This specific probabilistic relationship generates the data within a dataset. The developement of these models came with the need to briefly describe the elements within increasing large collections of data without loosing statistical relationships which are important for more straightforward analyses such as classification and summarization.

A topic is the theme, matter or subject of a text. Topics are often cosidered as discrete values, such as *politics*, *science* or *religion*, but any of these topics can be further divided into many sub-topics. Aditionally, the topics can overlap; for example the topic of *politics* can overlap with the topic of *health* and both of these topics can share the sub-topic of *health care*. Any of these topics can be described as clusters of weighted words. We suppose that topics are continuous, as there are infinitely many combinations of weighted words which can represent a topic; we also assume that any document has its own topic with a value in this continuum. A document typically consists of multiple

topics in different proportions [2].

A useful topic model should find topics which represent a high-level summary of the information present in the text. Each topic's cluster of words should describe information contained in the documents; for example, we can decide that the words *warming*, *global*, *temperature* and *environment* refer to the topic of *global warming*. Topic modeling is defined to be the process of finding the best-representing topics as weighted sets of words [2].

1.1.1 State of the art

In this section, I will give an overview of the present state of the art in the field of topic modeling with an emphasis on techniques which can be used for modeling motivic clusters of literary and poetic texts.

Kherwa and Bansal [3] created a classifaication strategy of topic modeling. In this text, topic modeling approaches are divided into probabilistic models and non-probabilistic models (algebraic models).

One of the algebraic approaches is Latent Semantic Analysis (LSA) [4]. LSA is a statistical method for analyzing a piece of text and for capturing relationships between terms in the documents and between the documents in the corpus by aggregating the different word usage in document collection. In LSA, a vector space model is created from singular value decomposition of a term-by-document matrix. The extracted semantic relationships are useful for making hybrid models for many NLP tasks including topic modeling [4]. Mohammed and Al-augby [5] applied LSA on a collection of e-books.

The study of Mohammed and Al-augby [5] uses a dataset consisting of 300 books which contains about 23 million words. The data are collected from the bookboon library, which is a platform for downloading e-books. The data were preprocessed and two topic modeling methods were then applied: LSA and LDA. The results were evaluated using the topic coherence measure [6]. The results show that LSA gives worse results than LDA.

Another algebraic approach is Non-Negative Matrix Factorization (NMF). NMF is a linear-algebraic optimization algorithm used for dimensionality reduction and data analysis. The NMF-based models determine the topics by directly decomposing the term-by-document matrix, which is factorized into a document-topic matrix and a topic-term matrix [7]. Young and Johnson [7] used NMF to learn the topics of works of two classical authors, Mark Twain and Edgar Allan Poe, downloaded from Project Gutenberg, which is a website that provides free e-books.

Young and Johnson's hypothesis [7] is that there are certain literary themes which are present in multiple works of classical authors. They chose two authors to compare their works: Mark Twain as a novelist and Edgar Allan Poe as a writer of short stories and poetry. The aim of the work was to compare topic models produced by NMF and LDA. The opinion of Young and Johnson is that the topics produced by NMF provide better understanding than those given by LDA for both authors.

One of the probabilistic methods is Probabilistic Latent Semantic Analysis (PLSA). It is a statistical technique for the analysis of a collection of documents based on the bag of words. It is evolved from LSA; in comparison to LSA which originates from linear algebra and uses a singular value decomposition to reduce term-by-document matrix, PLSA is based on decompositions derived from a latent class model [8].

A well-known probabilistic approach is Latent Dirichlet Allocation (LDA). The basic idea of LDA is that a particular document contains a random mixture of latent topics, where each topic is represented by a distribution over words [9]. Plechac and Haider [10] applied LDA to four corpora of Czech, Russian, German and English poems. Similar application of LDA was done by Navarro-Colorado who used this method for modeling a corpus of Spanish poetry [11] and by Haider who applied LDA to a corpus of New High German poetry [12]. Another use of LDA with poetic language was described by Lisa M. Rhody [13].

The aim of the Plechac and Haider's work [10] is to describe the evolution of poetic topics across four languages: Czech, Russian, German and English. They interpret their trend over time (1600–1925 A.D.), show similarities and differences between poetic traditions with a few selected topics and use their trajectories over time to determine specific literary epochs. The poems come from Corpus of Czech Verse, poetic subcorpus of Russian National Corpus, German Poetry Corpus and Project Gutenberg. The authors have found that some of the topics are shared by all languages, sometimes with temporal delay, while other topics did not appear so frequently in other poetic discourses.

Navarro-Colorado [11] applied LDA on the Corpus of Spanish Golden Age Sonnets, which consists of 5,078 sonnets of 52 poets written during the 16th and 17th century. The generated topics have been automatically evaluated and then manually analyzed. The author has drawn two main conclusions: the lemmatization process is not appropriate because of many poetic features disappearing and the standard LDA algorithm is more suitable for this type of text than the LF-LDA algorithm specifically developed for short texts.

Haider [12] chose the corpus of New High German poetry from the TextGrid

Repository for his research. The corpus contains 51,000 poems and covers a time period from the mid 16th century up to the first decades of the 20th century. Haider applies LDA on this corpus, interprets extracted topics such as their trend over time and uses the distribution of topics over documents to classify poems into time periods and to determine the poems' autorship. The author has found that most topics are clear and easily interpretable while others are quite noisy. The classification into time slots and the authorship determination seems to be very promising, however far from perfect.

The aim of Rhody's work [13] was to understand how topic models handle figurative language in comparison to the language of non-fiction texts. For her research, Rhody chose a genre of poetry called ekphrasis – poems written to, for, or about visual arts. The research has shown that poetic topics have lesser thematic clarity than those describing non-fiction texts. In addition, the traditional methods of evaluation are not very effective, because the expectations of how language should operate in poetry are different. In conclusion, the author says that understanding the generated topics should be accompained by thorough reading of the poems.

Attention is also drawn to neural models, especially transofrmers. Bidirecitonal Encoder Representations from Transformers (BERT) [14] is a language representation model designed for pretraining deep bidirectional representations from unlabeled text. BERT is pre-trained using two unsupervized tasks: masked language model, which means masking some percentage of input tokens at random and predicting them, and next sentence prediction, which is based on understanding the relationship between two sentences. For the pre-training corpus BERT uses the BooksCorpus and English Wikipedia; it is critical to use a document-level corpus in order to extract long contiguous sequences. A common pattern of fine-tuning is to independently encode text pairs before applying bidirectional cross attention. Another example of the application of transformers in the field of topic modeling is the work of Hoyle et al. [15].

Hoyle [15] uses knowledge distillation to combine the best attributes of probabilistic topic models and pretrained transformers. This method can be applied with any neural topic model to improve topic quality. The model is validated using three datasets which vary widely in domain, size and document length: 20 Newsgroups, Wikitext-103 and IMDb movie reviews. The method produces improvement in the aggregate and its results can be interpreted more specifically.

The Top2Vec model [2] produces topic, document and word vectors such that distance between them represents semantic similarity. A major advantage over traditional methods is that removing stop-words, lemmatization, stemming

and a priori knowledge of the number of topics is not required. Whereas probabilistic generative models model topics as distribution of words which are used to recreate the original document word distributions, a Top2Vec topic vector represents a prominent topic shared among documents. In his work, D. Angelov trained the Top2Vec model on the 20 Newsgroups dataset and the Yahoo Answers dataset and compared it with LDA and PLSA trained on the same datasets; all the models were evaluated using the probability-weighted amount of information (PWI). The results have shown that the Top2Vec topics are more localized in the semantic space and therefore more informative.

1.2 Preprocessing

Preprocessing is an important step for natural language processing tasks, since the characters, words and sentences are the fundamental units passed to all further stages. Its importance lies in the removal of redundant parts which do not contain valuable information.

The preprocessing of text documents consists of these following activities:

Tokenization is the process of dividing a stream of text into words, phrases, symbols and other meaningful elements. These components are called tokens. The process of tokenization is intended for the exploration of the words in a sentence; the resulting list of tokens becomes the input for further processing [16].

Stop words removal. Many words appear very frequently in the text documents, but they are essentially meaningless as they are only used to link other words in the sentence. These words are called stop words. Stop words do not affect the context or content of text documents; due to their frequent occurence, their presence is an obstacle in understanding the text. The removal of these stop words depends on a stop words list. This process also highly reduces the text data [16].

Stemming is a word normalization technique. It is the process of merging different forms of a word into a common representation, a stem. For example, the words *presentation*, *presented* and *presenting* share a common stem *present* [16].

Lemmatization is another word normalization approach. It is the process of assembling the inflected parts of a word such as they can be recognized as a single element, which is the word's vocabulary form – a lemma. It is a similar process to stemming, but the particular words have meaning. Either lemmatization or stemming can be used [17].

1.2.1 Czech language specifics

Czech is a West Slavic language belonging to the Czech-Slovak group. It is a fusional language with a rich morphology system and relatively flexible word order; its vocabulary has been significantly influenced by Latin and German [18].

Czech contains thirteen vowel phonemes; three of them can be found only in loanwords. They are ten monophthongs and three diphthongs. Czech also uses twenty five consonants; words may contain complicated consonant clusters or lack vowels at all. Czech has a raised alveolar trill, which is considered unique to it, represented by the phoneme *ř*. Each word usually has stress on its first syllable and the stress is unrelated to vowel length [18].

Czech grammar is fusional – its nouns, verbs and adjectives are inflected in order to modify their meanings and grammatical functions. Czech inflection is complex and pervasive [18]. Czech traditionally distinguishes ten parts of speech: nouns, adjectives, pronouns, numbers, verbs, adverbs, prepositions, conjunctions, particles and interjections [19]. Adverbs are usually formed from adjectives by taking the final *-ý* or *-í* of the base form and replacing it with *-e*, *-ě* or *-o* [18]. Negative words are formed by adding the affix *ne-* to a word. Czech grammar allows more than one negative word to occur in a sentence [20].

Czech word order is flexible, because it uses grammatical case to convey word function in a sentence instead of relying on word order typical for English; it is used for topicalization and focus. A word at the end of the clause is typically emphasized. Czech is a pro-drop language; an intrasistive sentence can consist only of a verb, because information about its subject is already expressed by the verb form. Relative clauses are introduced by relativizers such as the word *který*, which are analogous to English relative pronouns *which*, *that*, *who* and *whom* [18].

Nouns and adjectives are declined into one of seven grammatical cases. The inflection of nouns indicates their use in a sentence – subject nouns are marked with nominative case and object nouns with accusative case. The genitive case indicates possesive nouns and some types of movement; the remaining cases (instrumental, locative, vocative and dative) express semantic relationships such as secondary objects, movement, position or accompainment. The case of an adjective agrees with the noun that it describes. Some prepositions require the modified noun to take a particular case; other prepositions take one of several cases and their meaning depends on the case [18].

In Czech, three genders are distinguished: masculine, feminine and neuter. The masculine gender is subdivided into animate and inanimate. Feminine

nouns usually end in *-a*, *-e* or consonant, neuter nouns in *-o*, *-e* or *-í*, masculine nouns in a consonant (with few exceptions). Adjectives agree in gender and animacy with the nouns they belong to. In addition to the difference in noun and adjective declension, the gender also affect past-tense verb endings [18].

Czech grammatical number distinguishes between singular and plural. However, several residuals of dual forms remain; in some cases, some nouns for paired body parts use a historical dual form to express plural [18].

Czech verb conjugation is less complex than noun and adjective declination. Verbs agree with their subjects in person (first, second or third) and number (singular or plural) and are conjugated for tense (past, present or future). Verbs are marked for one of two grammatical aspects: perfective and imperfective. Most verbs are parts of inflected aspect pairs – their meaning is similar, but in the perfective form the action is completed and in the imperfective form the action is ongoing. The verbs of most aspect pairs differ in prefix or in suffix. The most common prefixes used for creating perfective forms are *na-*, *o-*, *po-*, *s-*, *u-*, *vy-*, *z-*, *za-*; in suffix pairs, a different ending is added to the perfective stem (for example *koupit – kupovat*). Many verbs have only one aspect. Three grammatical moods are distinguished: indicative, imperative and conditional [18].

Czech orthography is one of the most phonemic orthographies of all European languages. It has thirty-one graphemes representing thirty sounds (*i* and *y* sound the same) and it contains one digraph *ch*. The characters *q*, *w* and *x* appear only in foreign words. Some letters can form new characters with the use of caron: these are *š*, *ž* and *č*, as well as *ň*, *ě*, *ř*, *ť* and *ď* (the latter five are uncommon outside Czech). Czech also distinguishes vowel length; long vowels are marked with acute accent (*á*) or occasionally with a ring (*ů*) [18].

For language processing, some Czech corpora can be used. The Czech National Corpus [1] is an academic project founded in 1994 by the Faculty of Arts at the Charles University. Its aim is to systematically map Czech and other languages in comparison to it. The Czech Academic Corpus [2] was created by the team from the Institute of the Czech Language, of the Academy of Sciences of the Czech Republic. Its original purpose was to build a frequency dictionary of the Czech language. Some Czech language processing tools are also provided by the Natural Language Processing Centre [3] at the Faculty of Informatics of the Masaryk University in Brno.

[1] https://www.korpus.cz/
[2] https://ufal.mff.cuni.cz/cac
[3] https://nlp.fi.muni.cz/en/NLPCentre

1.2.2 Bag-of-words

The bag-of-words model is a simplyfiing text representation. In this model, a document is represented as a vector of terms which occur in the document. A term can be a single word (1-gram) or it can consist of multiple words (n-gram) that are present in the document [21].

Term frequency – tf – measures number of occurences of a term t_i in a document d_i. It is defined by

$$tf(t_i, d_i) = \#(t_i, d_i),$$

where $\#(t_i, d_i)$ is the number of occurences of t_i in d_i. Terms with high frequency usually do not contain useful information for discriminating the documents; inverse document frequency – idf – is a measure which favors terms that are present in few documents of the collection. It is defined by

$$idf = \log \frac{n}{\#D_{t_j}},$$

where n is the total number of documents in the collection D and $\#D_{t_j}$ is the number of documents in D where the term t_j occurs at least once. Another aspect that should be considered is that larger documents usually have higher probability of being relevant than smaller documents. Independently of their size, all relevant documents should be considered equally important and therefore a normalization factor should be incorporated; the $tfidf$ measure includes the normalization factor and is defined by

$$tfidfn(t_j, d_i) = \frac{tfidf(t_j, d_i)}{\sqrt{\sum_{s=j}^{n}(tfidf(t_s, d_i))^2}} \text{ [21].}$$

1.2.3 Word embeddings

Word embedding is a term used for NLP techniques where meaning of words and phrases is represented as vector of real numbers. Research has shown that these word vectors can significantly improve and simplify many NLP applications.

The backbone principle of word embedding methods is that the meaning of a word is related to the context where it usually occurs; it is possible to compare the meanings of two words by statistical comparison of their contexts. A vector representing the word meaning in distributional semantic models reflects the contextual information of the word across the training corpus. Each word $w \in W$ (where W stands for the word vocabulary) is associated with a vector of real numbers $\boldsymbol{w} \in \mathbb{R}^k$. Geometrically interpreted, the word meaning is a

point in a high-dimensional space; the words with closely related meaning are closer in the space [22].

CBOW (Continuous Bag-of-Words) representation predicts the current word by a small context window around the word. The architecture is similar to feed-forward Neural Network Language Model; while NNLM is computationally expensive between the projection and the hidden layer, the hidden layer of CBOW is removed and the projection layer is shared among all words. The projection is not influenced by word order in the context; this architecture also shows low computational complexity [22].

Another architecture similar to CBOW is Skip-gram. Instead of predicting the current word by its context, it tries to predict the word context based on the word itself; the Skip-gram model is intended to find word patterns that are useful for prediction of the surrounding words. Skip-gram model is slightly worse at estimating syntactic properties of words than the CBOW model, but does better at modeling the word semantic. Its training is also efficient [22].

Another approach is the GloVe model. This model is more focused on the global statistics of the data. It is based on the analysis of log-bilinear regression models which effectively capture global statistics and word analogies. The authors come up with a weighted least squares regression model which is trained on global word-word co-occurence counts. The basis of the model is the observation that ratios of word-word co-occurence probabilities are able to encode meaning of words [22].

1.3 Methods

In this section, I will introduce some of the methods of topic modeling.

1.3.1 Latent Semantic Analysis

Latent Semantic Analysis (LSA) is a fully automatic statistical technique for extracting relations of expected usage of words in passages of discourse [23]. LSA was created at Bellcore labs as a tool for information retrieval and was followed by psychological work in discourse processing [24].

The LSA algorithm is as follows:

1. Get a corpus of text files (documents).

2. Create a term-by-document matrix. Each cell of this matrix marks the number of occurence of a particular term in a particular document.

3. Lower the effect of the most commonly used words by applying a weighting method.

4. Apply singular value decomposition and reduce its output to the desired number of dimensions [4].

LSA is based on the Vector Space Model (VSM), which is an algebraic model for representing terms and documents in a high-dimensional space. VSM represents a collection of d documents in a space of t terms as a $t \times d$ matrix $\boldsymbol{X}$. The term dimensionality of the matrix $\boldsymbol{X}$ is reduced by the use of preprocessing techniques such as stop words removal, stemming or lemmatization. Each cell in $\boldsymbol{X}$ contains the frequency count of occurence of term i in document j; all cell entries are subjected to a preliminary transformation where more frequent terms are discounted and less frequent terms are promoted. The common transformation methods include the term frequency–inverse document frequency (TF–IDF) and the log-entropy transformation [24].

Because documents are represented as vectors in the term space as well as terms are represented as vectors in the document space, the quantification of a document collection as the term-by-document matrix $\boldsymbol{X}$ allows to calculate the term-to-term and document-to-document similarities. A commonly used similarity metric is the cosine similarity, which is defined as the cosine of the angle formed by two vectors. Maximum similarity is indicated by the cosine of 0° which is equal to 1; cosines of small vectors which are close to 1 indicate that the vectors have large degree of similarity. In linear algebra, the cosine can be expressed as the inner product of the two vectors divided by the product of their lengths; when the vectors are normalized, the cosine is equal to the inner product. For a set of q documents represented in the term space by the normalized matrix $\boldsymbol{Q}$, the pairwise cosine similarities of d documents represented by $\boldsymbol{X}$ are acquired as the $q \times d$ matrix $\boldsymbol{R}$:

$$\mathrm{Sim}(\boldsymbol{Q}, \boldsymbol{X}) = \boldsymbol{R} = \boldsymbol{Q}^\intercal \boldsymbol{X} \text{ [24].}$$

The term frequency matrix $\boldsymbol{X}$ is then decomposed using a matrix operation called singular value decomposition (SVD). In SVD, $\boldsymbol{X}$ is decomposed into the product of three other matrices: term eigenvectors $\boldsymbol{U}$, document eigenvectors $\boldsymbol{V}$ and singular values Σ. The resulting equation looks as follows:

$$\boldsymbol{X} = \boldsymbol{U}\Sigma\boldsymbol{V}^\intercal.$$

Using SVD, $\boldsymbol{X}$ is converted to a space of latent semantic dimensions; the elements of the diagonal matrix Σ, called singular values, describe the relative importance of these dimensions. Only the k most important dimensions can be kept, which produces the truncated version of the term frequency matrix

$\boldsymbol{X}_k$. The matrix $\boldsymbol{X}_k$, a least-squares best approximation of the original matrix $\boldsymbol{X}$, takes into account a hidden topic structure and thus modifies the original term frequencies. A smaller k value may associate certain terms with a broader context, while a larger k value may lead to a finer distinction of different fields the terms are associated with. The optimal k is usually chosen empirically [24]. The SVD can sufficiently uncover the underlying semantic structure of the document collection; the computation is also manageable for large datasets [4].

1.3.2 Probabilistic Latent Semantic Analysis

Probabilistic Latent Semantic Analysis (PLSA) is a probabilistic variant of standard LSA; it has a sound statistical foundation and provides a proper generative model of the data. The statistical model of PLSA is called aspect model. The aspect model is a latent variable model which links an unobserved class variable $z \in Z = \{z_1, ..., z_K\}$ with each observation. A joint probability model over $D \times W$, where $D = \{d_1, ..., d_N\}$ is a collection of documents and $W = \{w_1, ..., w_M\}$ are terms from the vocabulary, is defined by the mixture

$$p(d, w) = p(d)p(w \mid d),$$

where $p(w \mid d) = \sum_{z \in Z} p(w \mid z)p(z \mid d)$ [25].

The aspect model states that d and w are independent conditioned on the state of the associated latent variable. The model can be equivalently described as

$$p(d, w) = \sum_{z \in Z} p(z)p(d \mid z)p(w \mid z),$$

which is perfectly symmetric in both documents and words [25].

The procedure for estimating the parameters of PLSA is the iterative Expectation-Maximization (EM) algorithm. Its aim is to fit a training corpus D by maximizing the log-likelihood function

$$\mathcal{L} = \sum_{d \in D} \sum_{w \in d} f(d, w) \log(p(d, w)),$$

where $f(d, w)$ is the frequency of word w in document d. The EM algorithm alternates two steps:

1. the E-step, where posterior probabilities are computed for the latent variables as
$$p(z \mid w, d) = \frac{p(w \mid z)p(z \mid d)}{\sum_{z'} p(w \mid z')p(z' \mid d)}$$

2. the M-step, where parameters are updated to maximize $\mathcal{L}$:

$$p(w \mid z) = \frac{\sum_d f(d,w)p(z \mid w,d)}{\sum_{w'}\sum_d f(d,w')p(z \mid w',d)}$$

$$p(z \mid d) = \frac{\sum_w f(d,w)p(z \mid w,d)}{\sum_{z'}\sum_w f(d,w)p(z' \mid w,d)} [26]$$

The training process is followed by the "folding-in" process. In this process, the estimated $p(w \mid z)$ parameters are used to estimate $p(z \mid q)$ for the new documents q. The use of the EM algorithm is similar to the training process; the E-step is identical and the M-step recalculates $p(z \mid q)$ while all the $p(w \mid z)$ are kept constant. The folding-in process usually needs only a small number of iterations [26].

1.3.3 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) is a generative probabilistic model of a text corpus. According to its basic idea, the documents in the collection are represented as random mixtures over latent topics, where each topics is characterized by distribution over words. LDA was first introduced by Blei, Ng and Jordan in 2003 and it is one of the most popular methods for topic modeling [27].

The generative process for each document w in a corpus D is as follows:

1. Choose $N \sim \text{Poisson}(\xi)$.
2. Choose $\theta \sim \text{Dir}(\alpha)$.
3. For each of the N words w_n:
   a) Choose a topic $z_n \sim \text{Multinomial}(\theta)$.
   b) Choose a word w_n from $p(w_n \mid z_n, \beta)$ – a multinomial probability conditioned on the topic z_n [9].

In this generative process, words in documents are only observed variables while others are latent variables (θ) or hyperparameters (α and β). The probability of observed data D is obtained from the corpus as a product of the marginal probabilities of single documents:

$$p(D \mid \alpha, \beta) = \prod_{d=1}^{M} \int p(\theta_d \mid \alpha)(\prod_{n=1}^{N_d} \sum_{z_{d_n}} p(z_{d_n} \mid \theta_d)p(w_{d_n} \mid z_{d_n}, \beta))d\theta_d,$$

where α is the parameter of a Dirichlet prior on the per-document topic distribution, β is the parameter of a Dirichlet prior on the per-topic word distribution, M is the number of documents and N is the size of vocabulary [9].

The hyperparameters of LDA can be estimated by various methods.

Gibbs sampling is a Monte Carlo Markov-chain algorithm. This method creates a sample from a joint distribution when only conditional distributions of each variable can be efficiently computed [27].

Expectation-maximization (EM) finds the parameters α and β by maximizing the log-likelihood of the data. It consists of two steps: the E-step, where the optimizing values of variational parameters are found, and the M-step, where the log-likelihood is maximized with respect to the parameters α and β [9].

Variational Bayes inference (VB) is a type of EM extension which optimizes the fit to the data by using a parametric approximation to the posterior distribution of both parameters and other latent variables [27].

A new document can contain words which are not present in any of the documents in the training corpus. This leads to assigning zero probability to such words, and thus zero probability to new documents. This problem can be solved by "smoothing" the multinomial parameters – assigning positive probability to all words regardless of whether they are present in the training corpus or not. A commonly used method is the Laplace smoothing; it is often implemented in practice, although it is no longer considered a maximum a posteriori method. The solution proposed by Blei et al. is applying variational inference methods to the extended model which includes Dirichlet smoothing on the multinomial parameter [9].

1.3.4 Non Negative Matrix Factorization

Non-negative matrix factorization (NMF) determines topics by directly decomposing the term-by-document matrix. The term-by-document matrix $\boldsymbol{A} \in \mathbb{R}_{+}^{M \times N}$, where N is the number of documents and M is the number of distinct words in the vocabulary, represents the corpus. Each column vector $\boldsymbol{A}_{(:,j)} \in \mathbb{R}_{+}^{M \times 1}$ marks a bag-of-words representation of a document j described by M terms [28].

The goal is to find two lower-rank matrices $\boldsymbol{W} \in \mathbb{R}_{+}^{M \times K}$ and $\boldsymbol{H} \in \mathbb{R}_{+}^{N \times K}$,

where K is the number of latent factors (topics), such as

$$\boldsymbol{A} \approx \boldsymbol{W}\boldsymbol{H}^{\intercal}.$$

The column vector $\boldsymbol{W}_{(:,k)} \in \mathbb{R}_{+}^{M\times 1}$ represents the k-th topic described by M terms and the row vector $\boldsymbol{H}_{(j,:)} \in \mathbb{R}_{+}^{1\times K}$ describes a document j in terms of K topics. The NMF model ha s great performance in clustering high-dimensional data [28].

1.3.5 BERT

Bidirectional Encoder Representations from Transformers (BERT) is a language representation model which pretrains deep bidirectional representations from unlabeled text. The pre-trained model can be then fine-tuned and perform many natural language processing tasks including topic modeling [14].

While context-free models such as GloVe or word2vec produce a single word embedding for each word in the vocabulary, BERT as a contextual model provides a representation of each word depending on the context, the other words in the sentence. The main goal is to create a fixed-length vector representation of a sentence based on the meaning and order of words in it. Different languages need different BERT models [29].

As indicated, the BERT model consists of two steps: pre-training and fine-tuning. During the pre-training phase, unlabeled data are used to train the model over different pre-training tasks. In their work, Devlin et al. [14] decided not to use traditional left-to-right or right-to-left language models, but two unsupervised tasks: Masked Language Model and Next Sentence Prediction.

Masked Language Model masks some amount of input tokens at random and then predicts them. The final hidden vectors of the mask tokens are then sent to an output softmax over the vocabulary. A disadvantage of this procedure is that the mask token is not present during fine-tuning, which creates a mismatch between pre-training and fine-tuning. A way to avoid it is to not always replace "masked" token with the actual mask token.

Next Sentence Prediction is a procedure designed to understand the relationship between two sentences. It is a binarized task that can be generated from any monolingual corpus. When a pair of sentences is chosen for each pre-training example, 50% of the time the second sentence is the actual following sentence and 50% of the time it is a random sentence from the corpus.

For the pre-training corpus, Devlin et al. [14] use the BooksCorpus with 800M words and the English Wikipedia with 2,500M words. Because extracting of

long contiguous sentences is needed, it is important to choose a document-level corpus rather than a shuffled sentence-level corpus.

Next step of the BERT model is fine-tuning. For similar applications, it is common to independently encode text pairs and then apply bidirectional cross attention; BERT unifies these two stages by using a self-attention mechanism. For each task, the task-specific inputs and outputs are fed into BERT and all the parameters are finetuned end-to-end. The procedure of fine-tuning is relatively inexpensive compared to pre-training [14].

1.3.6 Top2Vec

The Top2Vec model [2] is a topic model which computes jointly embedded topic, document and word vectors; the semantic similarity is represented by the distance between them. A major advantage over traditional methods is that it does not require text preprocessing such as removing stop-words, stemming or lemmatization, as well as a priori knowledge of the number of topics. The results provided by Angelov [2] show that the topics determined by Top2Vec are significantly more informative and representative than those created by traditional methods such as PLSA or LDA.

A starting point for the Top2Vec model is the idea that the semantic space is a continuous representation of topics; each point in this representation is a different topic which is best described by its nearest words. To extract topics, jointly embedded document and word vectors are needed. The distance between document vectors and word vectors should represent semantic association; semantically similar documents should be placed close together while dissimilar documents should be placed further from each other. Words should be also close to documents they best describe. With these vectors, topic vectors can be computed. To learn these vectors, the Doc2Vec algorithm is used [2].

In such semantic space, a dense area of documents found there can be interpreted as a cluster of highly similar documents; existence of this area indicates an underlying topic common to the documents. The number of prominent topics is determined by the number of these dense areas of documents. The topic vectors are then computed as the centroids of each of these areas; they can be thought of as the average documents which represent their areas best. The most representative words of each topics are found by determining the closest word vectors to each topic vector [2].

There are multiple ways of how the topic vector can be calculated from the document vectors. The simplest method is to find the centroid of the document cluster, which is the arithmetic mean of the documents. Other options to get

the topic vector are calculating the geometric mean or using probabilities from the confidence of clusters. The experiment done by Angelov [2] showed that all of these techniques created very similar topic vectors with almost identical nearest-neighbor word vectors.

As mentioned, every topic is best described semantically by its nearest word vectors. The semantical similarity of a word and a topic is indicated by the distance of the word vector to the topic vector. The words which are placed closest to the topic vector can be considered as the words that are most similar to all documents in the cluster. It was shown that common words which occur in most documents are often equally distant from all documents, so the words closest to a topic vector are rarely stopwords [2].

An advantage of the Top2Vec model is that it allows a hierarchical reduction of the number of topics. This is achieved by iterative merging of the smallest topic into its most semantically similar topic until the desired number of topics is obtained. The merging itself is done by calculating a weighted arithmetic mean of the topic vectors of these two topics; each topic vector is weighted by its topic size. For each topic, the topic sizes are then recalculated [2].

Chapter 2

Experiments

In this chapter, I will describe the Corpus of Czech Verse dataset, the preprocessing of the data and the implementation of some of the topic modeling methods mentioned in the previous chapter. The methods which I chose are Latent Semantic Analysis, Latent Dirichlet Allocation, BERT and Top2Vec. All methods are implemented in Python.

2.1 Dataset

The dataset used for my work is the Corpus of Czech Verse [4]. The Corpus of Czech Verse is being built at the Institute of Czech Literature of the Czech Academy of Sciences and is lemmatized, phonetically, morphologically, metrically and strophically annotated. The texts of the corpus come from the Czech electronic library [5]; because a number of duplicates occurs there, only the oldest occurence of each poem was included into the corpus to avoid misrepresentation of statistical data. The corpus contains 1,689 books of poetry with 76,699 poems.

The GitHub repository of the corpus [6] contains 1,305 books of poetry processed as JSON files; 384 books are still under the copyright protection. Each JSON file holds poems from a single book of poetry. The files contain the text of the poems themselves and their metadata, as well as thorough annotation of poetic meters, annotation of rhymes, phonetic transcription, tokenization, lemmatization and morphological tagging.

[4] https://versologie.cz/en/kcv.html
[5] http://www.ceska-poezie.cz/cek/
[6] https://github.com/versotym/corpusCzechVerse/

2.2 Data preprocessing

The preprocessing of the data is relatively simple while the JSON files from the GitHub repository of the Corpus of Czech Verse contain lemmatized forms of all words occuring in poems, so extra tokenization and lemmatization is not needed. The JSON files are read using the `pandas` library and the lemmatized words are extracted. While each file contains a collection of poetry, the extracted words must be grouped into individual poems.

An important part of data preprocessing is stop-words removal. The list of stop-words was created by merging lists from countwordsfree.com [7], GitHub [8] and a list provided by the supervisor of my thesis, which takes into account stop-words occuring in processed poems. Some of the most frequent verbs were also added, which were obtained from the list of the most frequent words of the Czech National Corpus [9].

To preprocess data, three functions were created: `read_stopwords`, which extracts a list of stop-words from a text file, `get_filenames`, which obtains a list of filenames from the GitHub repository, and `preprocess_data`, which loads files of these names and performs the data preprocessing using the list of stop-words.

The output of the data preprocessing is a list of lists of lemmatized tokens without stop-words; each list represents a single poem.

Figure 2.1: An example of a preprocessed poem

```
['ptáček',
 'bůh',
 'chvalozpěv',
 'zpívat',
 'potřebný',
 'zob',
 'mívat',
 'stůl',
 'modlívat',
 'poděkování',
 'najíst',
 'říhat']
```

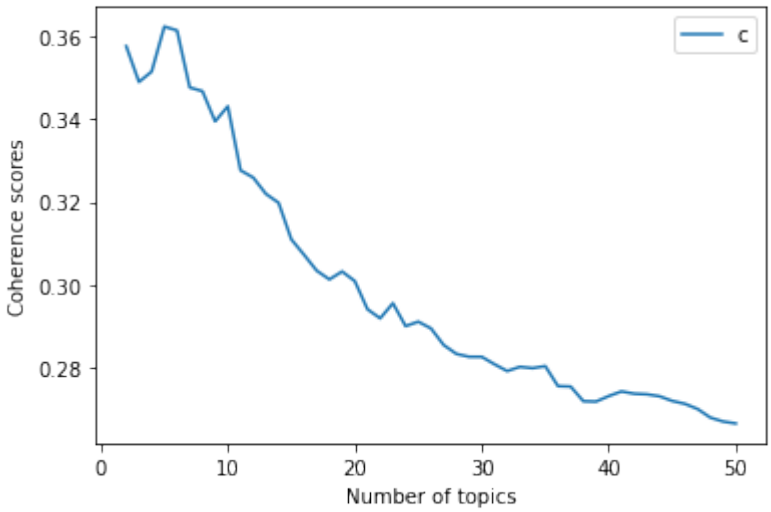
[7] https://countwordsfree.com/stopwords/czech
[8] https://github.com/stopwords-iso/stopwords-cs
[9] https://wiki.korpus.cz/doku.php/seznamy:srovnavaci_seznamy

2.3 Latent Semantic Analysis

For the implementation of Latent Semantic Analysis, the `LsiModel` from the `gensim` library is used. At first, a dictionary is created from the output of the data preprocessing. This dictionary performs the mapping between words and their integer IDs; it is then used to create a bag-of-words representation of the corpus. This representation is passed to the LSA model together with the dictionary.

While the LSA model needs an a priori konwledge of the requested number of topics, the key task is to determine it. In my work, numbers in range from 2 to 50 are passed to the algorithm and the results are evaluated using the topic coherence measure. The highest coherence score was achieved for five topics and the value of it is 0.362.

Figure 2.2: Topic coherence score of LSA model depending on number of topics



2.4 Latent Dirichlet Allocation

The implementation of Latent Dirichlet Allocation is also taken from the `gensim` library; the `LdaMulticore` model is chosen. This implementation is able to parallelize the task and therefore speed up model training.

The process of the creation of dictionary and bag-of-words representation is the same as that used in the implementation of LSA; both of these are alo required to train the LDA model. A fixed random state of the model was set in order to get a better reproducibility of results. With regard to the work of Plechac and Haider [10], who applied LDA on the Corpus of Czech Verse, the

Figure 2.3: Topics generated by LSA

	Topic 01	Topic 02	Topic 03	Topic 04	Topic 05
0	srdce	Karel	láska	bůh	láska
1	bůh	král	bůh	pán	srdce
2	oko	Roland	sláva	hvězda	Žižka
3	ruka	císař	starý	pan	duše
4	duše	pravit	srdce	země	země
5	hlava	velký	král	člověk	bůh
6	láska	srdce	smrt	noc	pán
7	země	svět	vlast	vidět	sen
8	svět	duch	ruka	Karel	Jan
9	tvář	láska	muž	muž	meč

hyperparameter `passes`, which determines the number of passes through the corpus during training, was set to 100.

As well as LSA, the LDA model requires an a priori knowledge of the number of topics. Plechac and Haider trained a LDA model on this corpus using 100 topics which they chose as an optimal number according to their previous research; I decided to pass numbers in range from 2 to 105 with the step of 5 to the algorithm and evaluate the models using the topic coherence measure. The highest coherence score was achieved for 47 topics and the value of it is 0.432.

2.5 BERT

Five BERT models were created; the first of them was implemented using the `SentenceTransformers` framework, UMAP and HDBSCAN, and four of them were created using the `BERTopic` library. For all models, the data was preprocessed and the tokens of each poem were joined to a single list.

2.5.1 BERT with UMAP and HDBSCAN

At first, an embedding model was created; the aim of it is to convert the documents to numerical data. For this purpose, the `sentence-transformers` package was used and the pretrained model `'distiluse-base-multilingual-cased-v2'` was chosen.

Uniform Manifold Approximation and Projection (UMAP) [30] is a manifold

Figure 2.4: Topic coherence score of LDA model depending on number of topics

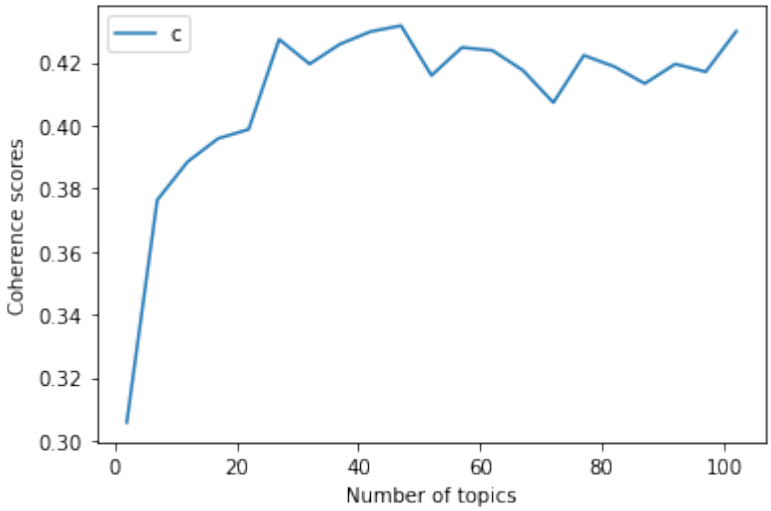


Figure 2.5: The first ten topics generated by LDA

	Topic 01	Topic 02	Topic 03	Topic 04	Topic 05	Topic 06	Topic 07	Topic 08	Topic 09	Topic 10
0	hora	svět	moře	sníh	národ	člověk	oko	láska	bůh	bůh
1	hlava	duch	vlna	vítr	vlast	pan	ruka	srdce	svět	svatý
2	bůh	člověk	břeh	mlha	sláva	muž	vidět	slza	srdce	boží
3	srdce	život	voda	země	lid	celý	hlava	drahý	krása	chrám
4	zas	věčný	loď	kraj	boj	dáma	chvíle	milovat	láska	kříž
5	oko	bůh	plout	zima	síla	paní	slovo	oko	sláva	duše
6	druh	lidský	řeka	bílý	bratr	velký	duše	svět	oko	nebe
7	mlýn	velký	proud	padat	věk	řeč	zrak	krásný	smrt	Kristus
8	les	lidstvo	bouře	mráz	krev	rád	zas	milý	sy	země
9	milý	žití	plachta	smutný	duch	trochu	náhle	duše	otec	anděl

learning technique for dimensionality reduction; it builds upon Riemannian geometry and algebraic topology. Its advantage is that it has no computational restrictions on embeddings dimension. In my work, the implementation of UMAP is taken from the `umap-learn` package [10] and dimensionality of the embeddings is reduced to 10 while keeping the size of local neighborhood at 20.

The next step is document clustering; this part was performed using HDB-

[10] https://umap-learn.readthedocs.io/en/latest/

SCAN. Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) [31] is a clustering algorithm which can handle noise and mark them as outliers. It is able to work with data with variable density, but it does not perform well on clustering high-dimensional data, so a dimensionality reduction method should be applied before. The implementation of HDBSCAN in my work comes from the `hdbscan` library [11]. The minimal cluster size was set to 10 and the embeddings reduced with UMAP were passed to it.

To extract topics, a class-based TF-IDF is used; the standard TF-IDF is applied on all documents in cluster, which are treated as a single document. The poems are grouped by the cluster they belong to and a single document for each cluster is created. Then, the TF-IDF score is computed to find the words which describe each topic best and the top ten words and sizes of each topic are obtained.

This model extracted 209 topics from the Corpus of Czech Verse and its coherence score is 0.371.

Figure 2.6: An example of a topic generated by BERT with UMAP and HDBSCAN

```
[('básník', 0.013635621033562832),
 ('kniha', 0.012316431889750765),
 ('poezie', 0.008506781092294769),
 ('psát', 0.007903220352570432),
 ('báseň', 0.007644980623035192),
 ('číst', 0.006511502409980026),
 ('verš', 0.00528441626234104),
 ('člověk', 0.004859141765209119),
 ('sen', 0.004765052106917316),
 ('starý', 0.004705822288586273)]
```

2.5.2 BERTopic

The `BERTopic` library [12] provides topic modeling technique which uses transformers and class-based TF-IDF to create easily interpretable topics. For my experiments with this library, four different models were created, which vary in selected embedding models and clustering methods of HDBSCAN.

Two embedding models were created: a `SentenceTransformer` model as an example of sentence embedding and a model from the `gensim` library as an example of word embedding. For the `SentenceTransformer` model, the

[11] https://hdbscan.readthedocs.io/en/latest/
[12] https://maartengr.github.io/BERTopic/index.html

pretrained model `'paraphrase-multilingual-mpnet-base-v2'` was chosen; from the models offered by the `gensim` library, a pretrained Word2Vec model was selected, namely the `'word2vec-ruscorpora-300'`.

Next, two HDBSCAN models were created that differ in selected clustering method: the first one uses the default option of Excess of Mass method [32], which usually picks one or two large clusters and then a number of extra small clusters, and the second one uses Leaf clustering [32], which is supposed to provide many small homogenous clusters. The minimal cluster size was set to 10 in both cases.

Also an UMAP model was provided; it reduces the dimensionality to 10 while preserving the size of local neighborhood at 10. A `CountVectorizer` from the `scikit-learn` library was chosen to perform the conversion of a document collection to a matrix of token counts; in this case, it allows to create unigrams as well as bigrams. All of these models along with the preprocessed data were passed to the `BERTopic` models.

The first `BERTopic` model was trained using the `SentenceTransformer` and HDBSCAN with Excess of Mass clustering. It extracted only 2 topics from the Corpus of Czech Verse; most of the poems fell into the first cluster. Its coherence score is 0.335.

The second `BERTopic` model was created with the use of the `SentenceTransformer` and HDBSCAN with Leaf clustering. It found 426 topics in the Corpus of Czech Verse and its coherence score is 0.359.

The third `BERTopic` model uses the Word2Vec embedding and HDBSCAN with Excess of Mass clustering. It extracted 1,190 topics from the Corpus of Czech Verse and its coherence score is 0.375.

The fourth `BERTopic` model was trained using the Word2Vec embedding and HDBSCAN with Leaf clustering. It found 1,250 topics in the Corpus of Czech Verse and its coherence score is 0.376.

2.6 Top2Vec

The Top2Vec method is implemented using the `Top2Vec` model from the `top2vec` library [13]. At first, the data was preprocessed as usual; the tokens of each poem were then joined to a single string, because the `Top2Vec` model implementation requires this format of input data. The data was then passed to the `Top2Vec` model and the hyperparameter `speed` was set on the `'deep-learn'` option, which is supposed to learn the best-quality vectors.

[13] https://github.com/ddangelov/Top2Vec

Figure 2.7: An example of a topic generated by BERTopic with Sentence-Transformer and Excess of Mass clustering

```
[('srdce', 0.014664731562888719),
 ('duše', 0.012117553362762367),
 ('láska', 0.010830647563429557),
 ('oko', 0.010827153736099847),
 ('svět', 0.010784478195352434),
 ('bůh', 0.009931108283395284),
 ('země', 0.009833598938901128),
 ('ruka', 0.009714024607226123),
 ('hlava', 0.009163408992479747),
 ('zas', 0.008158090938638517)]
```

Figure 2.8: An example of a topic generated by BERTopic with Sentence-Transformer and Leaf clustering

```
[('smrt', 0.009479300235142173),
 ('život', 0.00858113427612705),
 ('žít', 0.007128217318205433),
 ('žití', 0.006112677060771272),
 ('hrob', 0.005854100527521109),
 ('umřít', 0.0049902378123900136),
 ('mrtvý', 0.004532485150917112),
 ('smrt bát', 0.004406867155247361),
 ('zemřít', 0.004120685514618708),
 ('rok dvaadvacet', 0.0038131517055894153)]
```

Figure 2.9: An example of a topic generated by BERTopic with Word2Vec and Excess of Mass clustering

```
[('ušák', 0.0019329083988563072),
 ('šmaha', 0.001888319125330691),
 ('bergov', 0.0014721119441803733),
 ('šárka', 0.0012573223600036208),
 ('šedíka', 0.0012562718650338753),
 ('truldoslav', 0.0012562718650338753),
 ('střevíček modrý', 0.0012562718650338753),
 ('vlasta', 0.001249595833501113),
 ('brčko', 0.0012267599534836445),
 ('vězet', 0.0012217197686915746)]
```

The `Top2Vec` model creates jointly embedded document and word vectors at first; for this task, the default option of Doc2Vec model was chosen. This task is followed by the dimensionality reduction using UMAP and searching for dense areas of documents using HDBSCAN. The topic vectors are then

Figure 2.10: An example of a topic generated by BERTopic with Word2Vec and Leaf clustering

```
[('hmota', 0.002224092736011176),
 ('peklo', 0.002213451185818327),
 ('modlit děvče', 0.001833122191301305),
 ('děvče nevinný', 0.0018130312908659024),
 ('mrtvola živý', 0.0018130312908659024),
 ('nenávidět mrtvola', 0.0017435695218976143),
 ('nenávidět', 0.001665418015979382),
 ('prokop', 0.0016531787121768157),
 ('chechtat', 0.001523896429321223),
 ('jimram', 0.0015108594090549187)]
```

calculated as the centroids of the document clusters and the closest word vectors are found.

The `Top2Vec` model found 131 topics in the Corpus of Czech Verse and its coherence score is 0.454.

Figure 2.11: An example of a topic generated by Top2Vec

```
['lod' 'plavec' 'kormidlo' 'stozar' 'paluba' 'korab' 'clun' 'stezen'
 'prid' 'plachta' 'lodnik' 'pristav' 'vlajka' 'vrak' 'plout' 'plavba'
 'veslo' 'morsky' 'lodka' 'majak' 'lano' 'kapitan' 'kotva' 'vlna' 'uskali'
 'more' 'racek' 'pristat' 'namornik' 'ostrov' 'priboj' 'lodni' 'breh'
 'delfin' 'ocean' 'clunek' 'lodicka' 'pobrezi' 'brazdit' 'lodice' 'bradlo'
 'plavit' 'pena' 'utonout' 'zivel' 'skalisko' 'zpeneny' 'vyplout' 'barka'
 'vichr']
```

2.7 Evaluation

For evaluation of trained models, the topic coherence measure was used, namely the C_V variant [6]. The topic coherence measures the semantic similarity between the high-scoring words in the topic; the C_V variant is based on a sliding window and uses normalized pointwise mutual information and cosine similarity. According to the work of Röder et al. [6], the C_V measure has shown the highest correlation with human ratings.

To perform the topic coherence measure, the `CoherenceModel` from the `gensim` library was used. In case of other `gensim` models such as `LsiModel` or `LdaMulticore`, the measure can be performed directly with passing the model into the `CoherenceModel` constructor; in other cases, a dictionary must be created and used along with the generated topics for the computation.

Chapter 3

Results and discussion

An overview of all implemented methods sorted in descending order by their coherence scores:

Table 3.1: Methods and coherence scores

method	coherence score
Top2Vec	**0.454**
LDA	0.432
BERTopic with Word2Vec and Leaf clustering	0.376
BERTopic with Word2Vec and Excess of Mass clustering	0.375
BERT	0.371
LSA	0.362
BERTopic with SentenceTransformer and Leaf clustering	0.359
BERTopic with SentenceTransformer and Excess of Mass clustering	0.335

The greatest coherence score was achieved using the Top2Vec model; in my personal opinion, its results also seem to be most meaningful and easily interpretable. Another model which generated topics that I find very understandable from the human perspective is BERT.

The `BERTopic` models which used Word2Vec embeddings achieved greater coherence score than those which were trained using the SentenceTransformers; it seems that the word-level embeddings are a better choice for short documents such as poems. However, I find the topics generated with the use of SentenceTransformer generally more understandable. Also, Leaf clustering seems to provide slightly more coherent results than Excess of Mass clus-

tering. The topics created by different BERT models also seem to be quite variable. When applying BERT models, the majority of poems usually falls into outliers.

Some topics show certain clarity and appear multiple times across different models: these are for example *nature*, *God/religion*, *Czech nation*, *poetry/literature*, *love* or *sea/boat*. These topics may be considered important for Czech poetry. Many topics are not easily understandable and occur only seldom.

For an example of how a particular poem will be classified by different models, the poem *Jaroslavu Vrchlickému* written by Eduard Albert was chosen; it is the first poem of the whole dataset. The full-text version of this poem reads as follows:

"Tvá lod' jde po vysokém moři,
v ně brázdu jako stříbro reje,
svou přídu v modré vlny noří
a bok svůj pěnné do peřeje.

Tvá lana sviští, plachty duní
a třepe vlajka. V noční chvíli
zříš magický svit mořských tůní,
a ve snu, Albatros jak pílí.

Já samotním, jsem na ostrově,
ohýnek topím, rybku lově
zasedám na břeh za večera.

Dým v kotoučích se modrých krade,
kdes písklo ptáče, ještě mladé,
tma na mne hrozí z pološera." [33]

The classification of this text by different methods is captured in Table 3.2. The topics determined by Top2Vec and LDA seem to be very accurate from the human point of view while the topics generated by LSA and one of the `BERTopic` models seem to be assigned almost randomly. Other BERT models mark this poem as an outlier.

3.1 Future work

A next step can be hyperparameter tuning of BERT models. Different embedding models can be tried; `sentence-transformers` and `gensim` offer many pretrained models. Another models which can be applied are provided by the

Table 3.2: The main topic of *Jaroslavu Vrchlickému* according to different models

method	**top ten words of the topic**
Top2Vec	[loď, plavec, kormidlo, stožár, paluba, koráb, člun, stěžeň, příď, plachta]
LDA	[moře, vlna, břeh, voda, loď, plout, řeka, proud, bouře, plachta]
BERTopic with Word2Vec and Leaf clustering	outlier
BERTopic with Word2Vec and Excess of Mass clustering	outlier
BERT	outlier
LSA	[srdce, bůh, oko, ruka, duše, hlava, láska, země, svět, tvář]
BERTopic with SentenceTransformer and Leaf clustering	outlier
BERTopic with SentenceTransformer and Excess of Mass clustering	[srdce, duše, láska, oko, svět, bůh, země, ruka, hlava, zas]

`flair` library [14].

The hyperparameters of UMAP and HDBSCAN can be also tuned. While reducing dimensionality with UMAP, the size of the local neighborhood, dimensionality and minimal distance between samples can be the subject of an experiment. In HDBSCAN clustering, the minimal cluster size can be tuned.

Another option is implementation of the rest of methods described in the theoretical section, which are Probabilistic Latent Semantic Analysis and Non-Negative Matrix Factorization.

[14] https://github.com/flairNLP/flair

Conclusion

The aim of this thesis was to survey methods of topic modeling, apply them on the Corpus of Czech Verse and evaluate and compare their results. All of these tasks have been accomplished, although there is certainly room for improvement and further experimentation.

Although some of the trained models did not show very understandable results when applied on the Corpus of Czech Verse, some topics that are present in the Czech poetry from the 19th and the beginning of 20th century were captured. Topics that are generally considered important for Czech poetry such as *love*, *nature*, *religion* or *Czech nation* were extracted from the corpus.

All results were evaluated using the topic coherence measure. The most coherent topics were generated by the Top2Vec model; according to my personal opinion, they also showed the best understandability. In some cases, the topic coherence score did not fully correlate with my subjective view of the quality of the extracted topics.

Bibliography

[1] Vayansky, I.; Kumar, S. A. P. A review of topic modeling methods. *Information Systems*, June 2020, [cit. 2022-01-30]. Available from: https://www.sciencedirect.com/science/article/pii/S0306437920300703

[2] Angelov, D. Top2Vec: Distributed Representations of Topics. *arXiv preprint arXiv:2008.09470*, August 2020, [cit. 2022-02-04]. Available from: https://arxiv.org/abs/2008.09470

[3] Kherwa, P.; Bansal, P. Topic Modeling: A Comprehensive Review. *EAI Endorsed Transactions on Scalable Information Systems*, July 2019, [cit. 2022-02-04]. Available from: http://eprints.eudl.eu/id/eprint/682/

[4] Kherwa, P.; Bansal, P. Latent Semantic Analysis: An Approach to Understand Semantic of Text. In *2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC)*, September 2017, pp. 870–874, [cit. 2022-02-04]. Available from: https://ieeexplore.ieee.org/abstract/document/8455018

[5] Mohammed, S. H.; Al-augby, S. LSA & LDA topic modeling classification: comparison study on e-books. *Indonesian Journal of Electrical Engineering and Computer Science*, July 2019, [cit. 2022-02-17]. Available from: https://www.researchgate.net/profile/Salam-Al-Augby/publication/338817648_LSA_LDA_Topic_Modeling_Classification_Comparison_study_on_E-books/links/5e2bcdb04585150ee780a203/LSA-LDA-Topic-Modeling-Classification-Comparison-study-on-E-books.pdf

[6] Röder, M.; Both, A.; et al. Exploring the Space of Topic Coherence Measures. In *Proceedings of the eighth ACM international conference on Web search and data mining*, 2015, pp. 399–408, [cit. 2022-03-12]. Available from: https://doi.org/10.1145/2684822.2685324

[7] Young, M.; Johnson, D. A Comparison of LDA and NMF for Topic Modeling on Literary Themes. [online], April 2018, [cit. 2022-02-20]. Available from: https://wiki.ubc.ca/Course:CPSC522/A_Comparison_of_LDA_and_NMF_for_Topic_Modeling_on_Literary_Themes

[8] Barde, B. V.; Bainwad, A. M. An overview of topic modeling methods and tools. In *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*, 2017, pp. 745–750, doi:10.1109/ICCONS.2017.8250563.

[9] Blei, D. M.; Ng, A. Y.; et al. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 2003, [cit. 2022-03-08]. Available from: https://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf?TB_iframe=true&width=370.8&height=658.8

[10] Plechac, P.; Haider, T. Mapping Topic Evolution Across Poetic Traditions. *arXiv preprint arXiv:2006.15732*, August 2020, [cit. 2022-02-28]. Available from: https://arxiv.org/abs/2006.15732

[11] Navarro-Colorado, B. On Poetic Topic Modeling: Extractiong Themes and Motifs From a Corpus of Spanish Poetry. *Frontiers in Digital Humanities*, June 2018, [cit. 2022-03-01]. Available from: https://www.frontiersin.org/articles/10.3389/fdigh.2018.00015/full

[12] Haider, T. Diachronic Topics in New High German Poetry. *arXiv preprint arXiv:1909.11189*, September 2019, [cit. 2022-03-02]. Available from: https://arxiv.org/abs/1909.11189

[13] Rhody, L. M. Topic Modeling and Figurative Language. [online], 2012, [cit. 2022-03-30]. Available from: https://academicworks.cuny.edu/gc_pubs/452/

[14] Devlin, J.; Chang, M.-W.; et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*, May 2019, [cit. 2022-02-04]. Available from: https://arxiv.org/abs/1810.04805

[15] Hoyle, A.; Goel, P.; et al. Improving Neural Topic Models Using Knowledge Distillation. *arXiv preprint arXiv:2010.02377*, October 2020, [cit. 2022-03-04]. Available from: https://arxiv.org/abs/2010.02377

[16] Kannan, S.; Gurusamy, V. Preprocessing Techniques for Text Mining. *International Journal of Computer Science & Communication Networks*, 2014, [cit. 2022-03-08]. Available from: https://www.researchgate.net/profile/Vairaprakash-Gurusamy/publication/273127322_Preprocessing_Techniques_for_Text_Mining/links/54f8319e0cf210398e949292/Preprocessing-Techniques-for-Text-Mining.pdf

[17] Khyani, D.; Siddhartha, B. S.; et al. An Interpretation of Lemmatization and Stemming in Natural Language Processing. *Shanghai Ligong Daxue Xuebao/Journal of University of Shanghai for Science and Technologys*, 2020, [cit. 2022-03-09]. Available from: https://www.researchgate.net/profile/Siddhartha-B-S/publication/348306833_An_Interpretation_of_Lemmatization_and_Stemming_in_Natural_Language_Processing/links/6048467f299bf1e078696a3a/An-Interpretation-of-Lemmatization-and-Stemming-in-Natural-Language-Processing.pdf

[18] Wikipedia contributors. Czech language — Wikipedia, The Free Encyclopedia. [online], 2019, [cit. 2022-03-28]. Available from: https://en.wikipedia.org/w/index.php?title=Czech_language&oldid=890788875

[19] Wikipedie. Čeština — Wikipedie: Otevřená encyklopedie. [online], 2022, [cit. 2022-03-28]. Available from: https://cs.wikipedia.org/w/index.php?title={%}C4{%}8Ce{%}C5{%}A1tina&oldid=21074242

[20] contributors, W. Czech declension — Wikipedia, The Free Encyclopedia. [online], 2022, [cit. 2022-03-28]. Available from: https://en.wikipedia.org/w/index.php?title=Czech_declension&oldid=1069923050

[21] Martins, C. A.; Monard, M. C.; et al. Reducing the dimensionality of bag-of-words text representation used by learning algorithms. In *Proc of 3rd IASTED International Conference on Artificial Intelligence and Applications*, 2003, pp. 228–233.

[22] Svoboda, L.; Brychcín, T. New word analogy corpus for exploring embeddings of Czech words. [online], 2016, doi:10.48550/ARXIV.1608.00789. Available from: https://arxiv.org/abs/1608.00789

[23] Landauer, T. K.; Foltz, P. W.; et al. An introduction to latent semantic analysis. *Discourse processes*, 1998, [cit. 2022-03-30]. Available from: https://www.tandfonline.com/doi/abs/10.1080/01638539809545028

[24] Evangelopoulos, N. E. Latent semantic analysis. *Wiley Interdisciplinary Reviews: Cognitive Science*, 2013, [cit. 2022-03-30]. Available from: https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/wcs.1254?casa_token=Ny9FJjxKeaIAAAAA{%}3A83t3534HguAKyX6TJGXqpD_WrQiQNW0Mc4XAAavm6E81BSKUm5XT6jil6H8qFi2dJqGO7u2GLBftIw

[25] Hofmann, T. Probabilistic Latent Semantic Analysis. *CoRR*, 2013, [cit. 2022-04-02]. Available from: http://arxiv.org/abs/1301.6705

[26] Brants, T.; Chen, F.; et al. Topic-Based Document Segmentation with Probabilistic Latent Semantic Analysis. In *Proceedings of the eleventh international conference on Information and knowledge management*, New York, NY, USA: Association for Computing Machinery, 2002, ISBN 1581134924, pp. 211–218, doi:10.1145/584792.584829, [cit. 2022-04-11]. Available from: `https://doi.org/10.1145/584792.584829`

[27] Jelodar, H.; Wang, Y.; et al. Latent Dirichlet allocation (LDA) and topic modeling: models, applications, a survey. *Multimedia Tools and Applications*, 2018, [cit. 2022-04-13]. Available from: `https://link.springer.com/content/pdf/10.1007/s11042-018-6894-4.pdf`

[28] Shi, T.; Kang, K.; et al. Short-Text Topic Modeling via Non-Negative Matrix Factorization Enriched with Local Word-Context Correlations. In *Proceedings of the 2018 World Wide Web Conference*, Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2018, ISBN 9781450356398, p. 1105â€“1114, doi:10.1145/3178876.3186009, [cit. 2022-04-13]. Available from: `https://doi.org/10.1145/3178876.3186009`

[29] Atagün, E.; Hartoka, B.; et al. Topic Modeling Using LDA and BERT Techniques: Teknofest Example. In *2021 6th International Conference on Computer Science and Engineering (UBMK)*, 2021, pp. 660–664, doi:10.1109/UBMK52708.2021.9558988, [cit. 2022-04-15].

[30] McInnes, L.; Healy, J.; et al. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. [online], 2018, doi:10.48550/ARXIV.1802.03426, [cit. 2022-05-08]. Available from: `https://arxiv.org/abs/1802.03426`

[31] Asyaky, M. S.; Mandala, R. Improving the Performance of HDBSCAN on Short Text Clustering by Using Word Embedding and UMAP. In *2021 8th International Conference on Advanced Informatics: Concepts, Theory and Applications (ICAICTA)*, 2021, pp. 1–6, doi:10.1109/ICAICTA53211.2021.9640285, [cit. 2022-05-08].

[32] Campello, R. J. G. B.; Moulavi, D.; et al. Density-Based Clustering Based on Hierarchical Density Estimates. In *Advances in Knowledge Discovery and Data Mining*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, ISBN 978-3-642-37456-2, pp. 160–172.

[33] Albert, E. *Na zemi a na nebi*. Praha: František Šimáček, 1900.

[1] [14] [2] [6] [3] [4] [5] [7] [10] [11] [12] [15] [16] [17] [8] [9] [22] [21] [18] [19] [20] [13] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32] [33]

Appendix A

Acronyms

NLP Natural Language Processing

LSA Latent Semantic Analysis

PLSA Probabilistic Latent Semantic Analysis

LDA Latent Dirichlet Allocation

NMF Non-Negative Matrix Factorization

BERT Bidirectional Encoder Representations from Transformers

UMAP Uniform Manifold Approximation and Projection

HDBSCAN Hierarchical Density-Based Spatial Clustering of Applications with Noise

Appendix B

Content of enclosed CD

- `readme.txt` ..... brief description of the content of CD
- `src`
  - `thesis` ..... source code of the thesis in LaTeX format
  - `impl` ..... Jupyter notebooks with implementation of methods
- `text` ..... text of the thesis
  - `thesis.pdf` ..... text of the thesis in PDF format
  - `thesis.ps` ..... text of the thesis in PS format