CZECH TECHNICAL
UNIVERSITY
IN PRAGUE

F3

Faculty of Electrical Engineering
Department of Computer Science

Bachelor's Thesis

Assessing Facticity in Abstractive Summarization Methods

Czech ROUGE & Model-based metric

Šimon Zvára
Software Engineering and Technology

May 2022
Supervisor: Ing. Jan Drchal Ph.D.

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Zvára** Jméno: **Šimon** Osobní číslo: **491871**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra počítačů**

Studijní program: **Softwarové inženýrství a technologie**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Ověřování fakticity výstupů metod abstraktivní sumarizace textů

Název bakalářské práce anglicky:

Assessing Facticity in Abstractive Summarization Methods

Pokyny pro vypracování:

The task is to:
1) Research state-of-the-art NLP approaches for text summarization, focusing on evaluation methods for assessment of generated summaries including ROUGE and model-based approaches.
2) Discuss options for Czech datasets and models.
3) Select most promising evaluation methods and implement them (if needed). Propose extensions improving their support for Czech language.
4) Compare the methods using datasets and models supplied by the supervisor.

Seznam doporučené literatury:

[1] Nan, Feng, et al. "Entity-level Factual Consistency of Abstractive Text Summarization." arXiv preprint arXiv:2102.09130 (2021).
[2] Cao, Meng, et al. "Factual error correction for abstractive summarization models." arXiv preprint arXiv:2010.08712 (2020).
[3] Puspitaningrum, Diyah. "A Survey of Recent Abstract Summarization Techniques." Proceedings of Sixth International Congress on Information and Communication Technology. Springer, Singapore, 2022.
[4] Fabbri, Alexander R., et al. "Summeval: Re-evaluating summarization evaluation." Transactions of the Association for Computational Linguistics 9 (2021): 391-409.
[5] Kryściński, Wojciech, et al. "Evaluating the factual consistency of abstractive text summarization." arXiv preprint arXiv:1910.12840(2019).

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Jan Drchal, Ph.D. centrum umělé inteligence FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **02.02.2022** Termín odevzdání bakalářské práce: **20.05.2022**

Platnost zadání bakalářské práce: **30.09.2023**

Ing. Jan Drchal, Ph.D. podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	prof. Mgr. Petr Páta, Ph.D. podpis děkana(ky)

Acknowledgement / Declaration

I would like to thank my family, my friend Martin Hubal and my supervisor, Jan Drchal, for their support and encouragements throughout my work on this thesis.

I hereby declare that I have completed this thesis independently and that I have listed all the literature and publications used. I have no objection to usage of this work in compliance with the act §60 Zákon č. 121/2000Sb. (copyright law), and with the rights connected with the copyright act including the changes in the act.

In Prague, 11. May 2022

.....

Abstrakt / Abstract

Naše práce prozkoumává existující metody pro evaluaci generativních modelů používaných v úlohách sumarizace textů a navrhuje dvě metody pro evaluaci textů v Českém jazyce. Nejprve předkládá ROUGE-CS, upravenou verzi metriky ROUGE, rozšířenou o využití slovníků českých synonym, antonym, lemmat, výplňových slov a o porovnávání n-gramů na základě podobnosti vektorových reprezentací slov. Poté práce navrhuje Memes-CS (Metric for Evaluating Model Effectiveness in Summarization), metriku založenou na naučeném transformer modelu RoBERTa, a na závěr práce navrhuje metodu pro automatické generování datasetu vhodného k porovnávání kvalit sumarizací za pomoci transformací prováděných nad již existujícím českým sumarizačním datasetem SumeCzech. Účinnost obou metrik je porovnávána s původní verzí metriky ROUGE na ručně anotované množině párů sumarizací za pomoci výpočtu korelace s hodnoceními, která udělil člověk.

Klíčová slova: sumarizace textu, evaluace modelů, metrika ROUGE, transformer, model-based metrika, bakalářská práce.

Překlad titulu: Ověřování fakticity výstupů metod abstraktivní sumarizace textů (Český ROUGE a Model-based metrika)

Our work examines existing methods for the evaluation of generative models used in text summarization tasks and proposes two methods for evaluating texts written in the Czech language. It first introduces ROUGE-CS a modified version of the ROUGE metric, augmented by the use of dictionaries of Czech synonyms, antonyms, lemmas, and filler words, and by comparing n-grams based on the similarity of word embeddings. Secondly, we introduce Memes-CS (Metric for Evaluating Model Effectiveness in Summarization), a metric based on a pre-trained transformer model RoBERTa, and thirdly we introduce a method for automatic generation of a dataset suitable for comparing the quality of summarization using various types of transformations performed over the existing Czech summarization dataset SumeCzech. The performance of both metrics is compared with the original version of the ROUGE metric on a manually annotated set of summarizations by computing the correlation with human judgment.

Keywords: text summarization, model evaluation, ROUGE metrics, transformer, model-based metric, bachelor's thesis.

Contents /

Tables / Figures

Chapter 1
Introduction

In recent years, there has been a rapid expansion of machine learning in a variety of disciplines, mostly focusing on computer vision tasks such as image processing, recognition, classification or generation, patient diagnosis, but also on human language understanding. This involves both the processing of sound, such as speech recognition [Nassif, 2019] and the processing of written text converted into computer form [Sarker, 2021]. These tasks include language modeling, text generation, machine translation [Zhang, 2015], abstract dialogue, question answering [Andreas, 2016], and, last but not least, text summarization [Puspitaningrum, 2022].

In this thesis, we focus on text summarization. Summarization is a problem where only a longer text is available, from which we try to use machine learning methods to create a shorter snippet, suitable for publication in a commercial environment, on social networks, etc. The importance of summarization lies primarily in saving time that people would otherwise be forced to spend reading long publications such as internet articles, newspaper reports, or even entire books or magazines.

Making summarized versions of articles will not only increase the amount of information a reader can absorb, but will also increase the reader's interest in reading a wider variety of publications and finding the one that interests them and for which they would be willing to spend time reading. This can be crucial for services such as news portals, whose success depends on the number of users and the time they spend on their service.

Currently, there are already a significant number of neural models and datasets available for summarization, mostly in English. Examples of current state-of-the-art models are Google's Pegasus [Zhang, 2020] and T5 [Raffel, 2019] models, both using the so-called transformer architecture, consisting of an encoder/decoder used to convert the input into vectors, and several so-called attention layers, which are important for identifying how strongly parts of the input are related to each other [Vaswani, 2017].

Datasets suitable for machine summarization are now relatively easily accessible on the Internet. Most news portals have articles divided into three parts: a headline, a short abstract, and then the full text, therefore a large amount of data can be automatically extracted from these services. Various summarization models can be learned on this collected data such as text-to-abstract and text-to-headline, but also other generative models such as abstract-to-text[1].

Datasets can also be created manually by human annotation of the texts, which consists of reading them and then writing a summary by an annotator. However, this procedure is very expensive, especially if the source texts are longer, such as articles on Wikipedia.

To successfully solve the machine summarization problem, it is necessary to find a way to compare the individual learned neural models with each other to determine the one that gives the best results. In machine learning disciplines with discrete output, such as text or image classification, measuring the quality of a model is usually easy by

[1] `https://deepai.org/machine-learning-model/text-generator`

calculating the ratio of accurate classifications to the total number of input data. These models can then be simply ranked according to their accuracy and the best one can be selected. However, the method of measuring the accuracy may have its shortcomings, for example, if misclassification of some data carries more weight than misclassification of another.

In the case of machine summarization and text generation in general, measuring model quality is more complicated. There is no simple concept of accuracy here because the output of these models is usually a text of a certain length. In the case of summarization, we have three objects available: the original long text, the reference summarization (also known as the *gold* summarization; in the case of news portals, it's the abstract of the article; in the case of manual dataset annotation, this is the summarization written by a human for the article), and then the summarization generated by the neural model, the so-called *system* summarization. There is currently no uniform consensus on how to compare two pieces of text based on their similarity. This problem is addressed by various metrics and algorithms for the evaluation of machine text generation models. The most prominent ones are described in section 1.1.

1.1 Current evaluation methods

Nowadays, there are several methods for evaluating generative models based on the similarity of the output text to the input text or parts of it. For summarization models, similarity can be measured either to a reference summarization or to the original document.

The most notable evaluation methods optimized for the English language are listed in the following sections:

1.1.1 ROUGE

Rouge [Lin, 2004] is one of the older but still widely used metrics. It consists in calculating the overlap of token (token is most frequently a single word) tuples, so-called *n-grams*, between the system and reference summarization. There are different versions of the metric depending on the length of the tuples measured, the most common being `Rouge-1`, which measures the overlap individually token by token, and then `Rouge-2`, which measures the overlap of pairs of tokens adjacent to each other. There is also an extended version of `Rouge-L` that measures the longest sequence of tokens that occurs in both summaries simultaneously.

The output of the metric is a trio of decimal values ranging from 0 to 1. These are *precision*, *recall*, and *f-measure*.

Precision indicates the amount of information from the summary that is present in the reference summary. Calculated as the ratio between the number of matching n-grams and the total number of n-grams in the system summarization.

$$P = \frac{N_{matching\ ngrams}}{N_{system\ ngrams}}$$

Using this value, we try to answer the question: *"Doesn't the summaries contain made-up or irrelevant extra information?"*. Low precision may also mean that the words in the summarization are repeated too often.

Recall indicates the amount of information from the original text that is contained in the system summary. Calculated again as the number of identical n-grams but now in proportion to the total number of n-grams in the reference summarization. It attempts to answer the question: *"Does the summarization contain all the information it should?"*.

$$R = \frac{N_{matching\ ngrams}}{N_{gold\ ngrams}}$$

F-Measure is calculated by combining both recall and precision into one formula, giving an overall assessment of the quality of the summarization.

$$F = 2 * \frac{(Precision * Recall)}{(Precision + Recall)}$$

The ROUGE-2 algorithm can be demonstrated using the following pair of sentences:

Sweden's foreign minister signs official NATO membership application.

The foreign minister of Sweden signs an application for official NATO membership.

Precision: 0.273 | Recall: 0.429 | F-Measure: 0.333

More detailed examples are listed in the section Problems with ROUGE.

Instead of regular n-grams, the metric can also use so-called *skip n-grams*, which allow taking tuples of tokens that do not follow directly after each other. In this case, the maximum token distance (gap size) in a skip n-gram is usually given because of the sharp increase in the number of n-grams in the text. Examples of skip-grams are depicted in Figure 1.1.

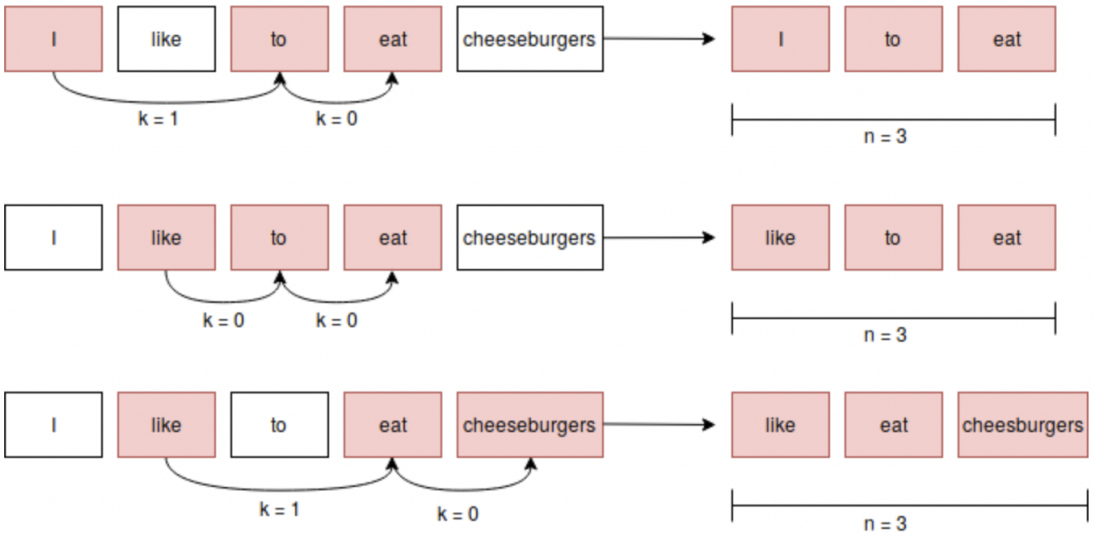

Figure 1.1. Example of 1-skip 3-grams, reprinted from [Struwig, 2019].

1.1.2 ROUGE-WE

An extension of the ROUGE metric using cosine similarity calculations of Word2vec embeddings [Mikolov, 2013] for individual tokens in n-grams. Tested on an English dataset, ROUGE-WE correlates better with human judgments than the original metric. [Ng, 2015]

1.1.3 S^3

A metric based on a neural model that attempts to assign an overall score to a system summary. The model takes as input the results of three metrics ROUGE, ROUGE-WE, and JS-divergence. [Peyrard, 2017]

1.1.4 BERTScore

A metric comparing generated and reference summarizations at the token level using the BERT model [Devlin, 2018]. The metric calculates a similarity score for each token in the system summarization with each token in the reference summarization, however, BERTScore does not compare direct matches such as ROUGE. Instead, it compares similarities of contextual embeddings. BERTScore is more robust when compared to ROUGE. [Zhang, 2019]

1.1.5 MoverScore

Metric based on the distance between generated and reference summarization using Earth Mover's Distance. The metric shows a higher correlation with human judgment on several text generation tasks including summarization, machine translation, image captioning, and data-to-text generation. MoverScore combines contextualized representations with distance measurements. [Zhao, 2019]

1.1.6 Sentence Mover's Similarity (SMS)

MoverScore extension using sentence embeddings in addition to word embeddings. Suitable for automatic evaluation of multi-sentence texts. [Clark, 2019]

1.1.7 SummaQA

A method applying the BERT model learned on question answering tasks. Certain tokens are masked in the original document and the model attempts to fill them in using the system summarization. The advantage is the unencessity of a reference summarization. [Scialom, 2019]

1.1.8 BLANC

A metric that measures the performance gain of a language-understanding model that operates on the original document text when it has access to the system summarization. The advantage is the unnecessity of a reference summarization. [Vasilyev, 2020]

1.1.9 SUPERT

A method that compares the generated summarization with a pseudo-reference that is created by an extractive selection of the most salient sentences from the original document. The advantage is the unnecessity of a reference summarization. [Gao, 2020]

1.1.10 BLEU

An old metric based on comparing identical n-grams in parts of the reference summarization and the system summarization, penalizing overly concise summaries. Unlike ROUGE, it is precision-oriented. It is the primary metric used in machine translation. [Papineni, 2002]

1.1.11 CHRF

A metric extending the original ROUGE to comparing the number of matching n-grams at the level of individual characters instead of whole tokens. Shows a higher correlation with human judgments. [Popovic, 2015]

1.1.12 METEOR

An automatic metric for machine translation evaluation that has been demonstrated to have high levels of correlation with human judgments of translation quality, significantly outperforming the more commonly used BLEU metric. METEOR utilizes mapping individual words from the system summarization to their corresponding words from the reference. It also takes into account synonyms, word roots, and paraphrases. [Lavie, 2007]

1.1.13 CIDEr

A consensus-based evaluation protocol for image description evaluation by detecting the number of shorter n-grams (1-4 words) that occur in both system and reference summarization. It uses the cosine similarity of individual n-grams in the calculations and reduces the weight of overly common n-grams. [Vedantam, 2015]

A more comprehensive comparison of the above methods can be found in the SummEval paper [Fabbri, 2021].

1.2 Motivation

The main motivation for this thesis is that common metrics such as ROUGE do not correlate very well with human assessment and thus do not reliably guarantee the facticity of machine summarizations. The problem further magnifies when these metrics are applied in languages with more complex grammar, such as Czech. We consider facticity to be a critical property of summarization, as the possible introduction of false information can have profound negative effects in some cases, and summarization can easily become misinformation. With the frequent occurrence of factual errors, the credibility of the source, such as a news portal, also decreases substantially from the public's point of view. Therefore, it is very important for us to be able to detect and filter out summaries whose content does not correspond to reality.

The original version of ROUGE has several problems that reduce its effectiveness when comparing pieces of text in non-English languages. In the first part of this thesis, we deal with its transformation and adaptation to be suitable for use with the Czech language.

We try to solve these problems by implementing ROUGE-CS, a Czech version of ROUGE, using heuristics, dictionaries of Czech synonyms, and Czech word embeddings using the Word2Vec [Mikolov, 2013] vector representation. In Section 2.1, the possible shortcomings of the ROUGE metric are discussed in more detail.

Since ROUGE is a very simple deterministic algorithm in implementation, it does not handle more complex differences between two texts very well. Therefore, in the second part of the thesis, we propose Memes-CS, a metric based on a learned XLM-RoBERTa [Conneau, 2019] model based on the transformer [Vaswani, 2017] architecture. For this particular task, we modified the dataset used in natural language understanding (NLI) tasks and also created a custom dataset using heuristic algorithms. We will discuss this approach in a later section regarding model-based metric.

1.3 Evaluation of proposed metrics

Since the aim of this work is to introduce two completely different metrics to evaluate system summaries to replace the widely used ROUGE, it is necessary to have a *meta-metric*, a way to compare these metrics against each other to determine which is the most accurate to evaluate the performance of generative models.

We found comparing the degree of correlation of the output metrics with human-annotated data to be the most appropriate solution. For this purpose, a set of pairs of reference and system summaries were generated using the MBART summarization model [Liu, 2020]. One hundred selected pairs were then manually annotated.

Although the output of all compared metrics is a decimal number, the pairs were annotated only with values of 1.0 (the system summary semantically matches the reference) and 0.0 (the system summary provides largely false or irrelevant information)

These pairs were then scored during the testing phase using the implemented metrics and a measure of correlation was calculated between the resulting vectors.[2]

Example of metric evaluation using correlation:

Human annotated labels:
[1.0, 1.0, 1.0, 0.0, 0.0, 0.0]

Labels assigned by the tested metric:
[0.46, 0.34, 0.65, 0.23, 0.05, 0.40]

Correlation = 0.687

The manually annotated test dataset is available in the file `annotated.json`.

[2] NumPy function `corrcoef` was used

Chapter 2
Czech ROUGE

2.1 Problems with ROUGE

As hinted at in the motivating text, the ROUGE metric has several fundamental flaws that reduce its effectiveness when used to evaluate pairs of summaries. The problems are discussed in the following sections:

2.1.1 Hypersensitivity

The most serious problem we encountered when testing the ROUGE metric was its hypersensitivity to even the smallest changes in individual words or sentences in the system summarization compared to the reference one. The ROUGE metric was originally implemented and tested on texts written in the English language, which uses fewer word forms than the Czech language. There are no cases or word genders in English and the individual persons (he/she/it) are only minimally distinguished. Since ROUGE compares the equivalence of strings representing individual words, any possible change due to declension or use of a different person is taken as a significant deviation between the reference and the system summarization and reduces the final text score on the output not insignificantly.

We can take the following pair of sentences as an example:

Marilyn Monroe, narozená v Los Angeles v Kalifornii, byla americká filmová herečka a zpěvačka.

Marilyn Monroová, s narozením v Kalifornském Los Angeles, bývala americkou filmovou herečkou a zpěvačkou.

Rouge-1: 0.35, Rouge-2: 0.07

As we can see in the example above, two semantically identical sentences with only slight changes in word shapes due to different declensions dropped the ROUGE metric from 1.0 (identical strings of text) to 0.35. This is because only 5 of the original 14 words were retained after editing.

The following example shows why this decrease is considered a negative feature of the metric:

Marilyn Monroe, narozená v Los Angeles v Kalifornii, byla americká filmová herečka a zpěvačka.

Marilyn Manson, zpěvák žijící v Kalifornii, se nedávno přestěhoval do Los Angeles.

Rouge-1: 0.38, Rouge-2: 0.17

Here, on the other hand, we can see two sentences that are very different in meaning, but for which the ROUGE metric gives more favorable results than in the first case, thanks to the preservation of the shapes of some words. The frequent occurrence of similar cases can lead to a situation where ROUGE favors models that extract some identical strings from the original text, regardless of their use in a semantically correct context.

When implementing the Czech version of the metric, we solved this problem using so-called lemmatization, i.e. finding a common root of words that is invariant concerning declension or conjugation.

2.1.2 Paraphrasing

In any language, we often encounter the problem where a single sentence can be written in two or more completely different ways using various synonyms, vernacular names, abbreviations, etc. The original ROUGE metric cannot cope with this problem very well, due to not taking into account the possible semantic similarity of individual words or phrases and treating any changes as an error.

We can take the following pair of sentences as an example:

Donald John Trump je americký republikánský politik, v letech 2017–2021 prezident Spojených států, podnikatel a bývalá televizní osobnost.

Donald J. Trump byl mezi lety 2017-21 v čele USA za stranu republikánů. Dnes je již pouze úspěšným businessmanem, též známým pro své vystupování v televizi.

Rouge-1: 0.18, Rouge-2: 0

This case shows that very similar sentences, just shaped differently, are treated as fundamentally different by the ROUGE metric. If the second sentence were a system summarization, it would most like be discarded due to its low score, despite being semantically similar to the reference summary.

Inspired by the ROUGE-WE [Ng, 2015], we solved the paraphrasing problem in the implementation of the Czech version of the metric by using synonym dictionaries and by calculating the distance of vector embeddings of individual n-gram tokens.

2.1.3 Negation

Most texts can be modified using negations or antonyms to give exactly the opposite meaning or at least to some extent to suppress the original meaning. Since antonyms differ from their counterparts in their shape, the ROUGE metric treats them as an error and their occurrence thus lowering the overall system summarization score. Although this behavior is considered desirable, the problem is that even a slight negation can fundamentally affect the meaning of an entire paragraph of text, in our case the entire summarization. However, ROUGE treats the replacement of a small number of words by their antonyms (for example, by adding the prefix “ne”) as only a slight deviation concerning the part of the words that remained unchanged. Again, this shows up as a lack of understanding of the semantics of the language, since the metric only focuses on the proportion of syntactically distinct words.

We can take the following pair of sentences as an example:

Ilana Bergerová je izraelská novinářka a bývalá profesionální tenistka. Ve své kariéře na okruhu WTA Tour nevyhrála žádný turnaj. V rámci ITF získala sedm titulů ve dvouhře a patnáct ve čtyřhře.

Ilana Bergerová je izraelská novinářka a současná profesionální tenistka. Ve své kariéře na okruhu WTA Tour neprohrála žádný turnaj. V rámci ITF získala sedm titulů ve dvouhře a patnáct ve čtyřhře.

Rouge-1: 0.93, Rouge-2: 0.87

In this case, it can be observed that by simply substituting two words with their respective antonyms, the meaning has been fundamentally changed, but the ROUGE metric shows only a small deviation from a perfect score. Such modified summaries could be considered semantically correct, which would in some cases lead to very negative misinformation effects.

We tried to deal with this problem again using synonym dictionaries and simple heuristics based on prefixes of words denoting negative meaning.

2.1.4 Pronoun, entity, and number swap

A common problem or generative model that extracts information from a context (in the case of summaries it's the original document) is the swapping of certain types of words, most often pronouns identifying people in the text, names of people, countries, organizations, days, months of the year, or numerals. Just like antonyms, these types of words can fundamentally alter the meaning of a text, but the ROUGE metric does not identify these problems to the necessary extent. These swaps occur most often when the model inserts a word from the original document into the system summarization, but it does not fit the context. However, it is also not uncommon for the model to insert random words from the corpus, which is also called *hallucinating*.

We can take the following pair of sentences as an example:

Ilana Bergerová je izraelská novinářka a bývalá profesionální tenistka. Ve své kariéře na okruhu WTA Tour nevyhrála žádný turnaj. V rámci ITF získala sedm titulů ve dvouhře a patnáct ve čtyřhře.

Martina Bergerová je palestinská novinářka a bývalá profesionální tenistka. Ve své kariéře na okruhu ITF Tour nevyhrála žádný turnaj. V rámci WTA získala patnáct titulů ve dvouhře a sedm ve čtyřhře.

Rouge-1: 0.93, Rouge-2: 0.63

The example shows that a semantically fundamental change elicits only a minimal response in the form of a reduced ROUGE rating.

We addressed the problem of word swapping by using heuristic methods to identify numerals and proper nouns in the text and introducing a higher penalty when the model confuses them for another word.

2.1.5 Unimportant words excessively affect the outcome

Whether some words represent important entities or just serve as fillers, the ROUGE metric assigns equivalent weight to deviation in any of them, impacting the final score equally. Filler words (so-called stop words), such as "vlastně" or "ještě", are common in the Czech language and do not contribute to the overall meaning of the text. The problem arises when the generative model omits or adds these filler words, thus negatively affecting the results.

We can take the following pair of sentences as example:

Martinu zajímalo, kolik by ty pěkné šaty tak mohly nakonec vlastně stát.

Martinu zajímalo, kolik by ty pěkné šaty mohly stát.

Rouge-1: 0.85, Rouge-2: 0.63

In the example, it can be observed that although the removal of a few filler words did not change the meaning of the sentence, the ROUGE score has been significantly reduced.

We solved this problem in our implementation by using dictionaries of Czech filler words to identify semantically unimportant parts of the system and reference summarization.

2.1.6 Noise injection

The final issue that we consider to be significant, and which we have focused on in this thesis, is the identification of noise introduced into the system summarization. By noise, one can picture random repetitions of words or, conversely, random omission. Unnecessary repetitions tend to be a common problem of generative models. ROUGE deals with this problem relatively well, which is why it is mentioned last.

We can take the following pair of sentences as example:

Marilyn Monroe byla americká filmová herečka a zpěvačka.

Marilyn Monroe byla americká filmová herečka, herečka, zpěvačka a zpěvačka.

Rouge-1: 0.89, Rouge-2: 0.75

The example shows that although the noise reduces the readability and clarity of the sentence, it does not significantly affect its semantic meaning.

When implementing the Czech metric, we attempted to solve this problem by penalizing already seen n-grams.

2.2 ROUGE-CS

In the first part of this thesis, we introduce ROUGE-CS, a modified version of the deterministic metric ROUGE to provide more relevant results when comparing system and reference summarization in the Czech language. We aimed to maximize the correlation of the new metric with human-annotated summarization pairs.

Although our metric differs from the original ROUGE metric in numerous aspects, we aim to maintain the same interface and means of interpreting the results. Thus, like the original metric, our metric takes as input a pair of texts (in our case, a pair of reference and system summarization) and outputs a triplet of precision, recall, and f-measure values. Our metric also supports comparing the two texts based on n-grams of arbitrary length, which can be set using an input argument. However, for complexity reasons, our metric does not allow comparing texts based on the longest common token subsequence.

The process of comparison is performed by first tokenizing both texts, then the Czech stop words are removed from both texts and then words of similar meanings are identified, this part is referred to as synonymization. Then, all combinations of n-grams are generated according to the chosen length, which are compared to each

other based on vector similarity. During the comparison, the total overlap of similar n-grams is computed at each step in the manner of the original metric. Overlap is then transformed into the resulting precision, recall, and f-measure in the last step using the formulas of the original metric.

The individual steps of the algorithm are described in more detail in the following sections.

2.2.1 Tokenization

ROUGE-CS uses a very simple tokenization heuristic where the text is split using a space character and then the punctuation marks ".", "!" and "?" are removed from each part.

During the experimentation, both variants converting to lower case and preserving the original case were tested. Results can be observed in Table 2.1. Although keeping the original case yielded a slight improvement of the results in the Rouge-1 case due to its greater sensitivity to changes in proper names starting with a capital letter, we eventually used the lowercase variant for easier string operations and proper names are being identified using a part-of-speech tagger.

2.2.2 Removing stop-words

The easiest method of removing stop words was sequentially iterating over all tokens and checking whether the token is present in a previously prepared dictionary of Czech stop words. The dictionary was maintained as a hash set and therefore each search has constant complexity.

ROUGE-CS uses freely available dictionaries[1][2][3], which were manually checked and from which words considered too semantically significant were removed. The resulting dictionary contains 183 words and can be found in the file `stopwords.txt`.

Removing stop words significantly improved the metric results for Rouge-1 and Rouge3 as can be observed in Table 2.1.

2.2.3 Synonymization

The purpose of this step is to identify synonyms between the two texts and to ensure that ROUGE-CS does not overly reduce the final similarity score when the overall meaning of the sentence remains unchanged.

During synonymization, the algorithm iterates over all pairs of words between the reference and system summarization and computes similarity using the distance of their vector representations. For the representation, a freely available dataset of Word2vec embeddings for the Czech language is used [Mikolov, 2013][4]. Getting the similarity of two vectors works on the principle of calculating the cosine similarity[5].

Since the result of the word embedding similarity calculation is a float in the range of 0.0–1.0, we determined a threshold of 0.7 in our metric that had to be exceeded for the two words to be considered as semantically indifferent. However, it is necessary to additionally test whether the words in question are direct antonyms because words of opposite meanings tend to have word embeddings close to each other. For this purpose, due to the lack of available antonym dictionaries, a simple heuristic based on identifying whether a given word is a number or finding the negative prefix "ne" was used.

[1] `https://github.com/Alir3z4/stop-words/blob/master/czech.txt`
[2] `https://github.com/stopwords-iso/stopwords-cs`
[3] `https://countwordsfree.com/stopwords/czech`
[4] `http://vectors.nlpl.eu/repository/`
[5] `https://www.sciencedirect.com/topics/computer-science/cosine-similarity`

If the threshold of 0.7 was not exceeded, the words were tested for similarity of meaning using a synonym dictionary search. Our metric uses the Dictionary of Spell Check, Word Splitting, and Synonyms for Czech, available as an add-on for LibreOffice[6]. The content of the dictionary is first converted into a hash map, where each word was associated with a list of possible synonyms. Then the determination of whether two words are synonyms is performed with a constant complexity. Due to the limited size of the dictionary, it is necessary to lemmatize the words first. The lemmatization is described in more detail in section 2.2.4.

For semantically equal word pairs, it was necessary to ensure that any syntactical difference between them was ignored. This is achieved by replacing the word in system summarization with its synonym occurring in the reference summarization. After this step, the two words are identical and the metric no longer captures any deviation.

During the experiments, in addition to Word2Vec representations, fastText [Bojanowski, 2017][7] representations were also tested, but the operations over them were slower due to the larger size of the set and we also achieved significantly worse results when fastText was used. We suspect that this might be due to fastText capturing too many different shapes of individual words, which causes unwanted noise when searching for nearest neighbors.

Synonymization greatly improved the results of all variants of the metric because of the increased emphasis on the meaning of the sentence and less focus on the spelling. Results can be observed in Table 2.1.

After synonymization, the step of stop-words removal was repeated due to possible newly occurring filler words in the text.

Pseudocode of the synonymization step is provided in Figure 2.1.

2.2.4 Lemmatization

The open-source parts-of-speech tagger MorphoDiTa [Straka, 2014][8] is used for lemmatization and converting words to their base variant (for example, the word “plánuje” is converted to “plánovat”).

During experimentation, alternative lemmatizer UDPipe [Straka, 2016][9] was also tested, but with less promising results, as can be observed in Table 2.1.

2.2.5 Generating n-grams

To generate n-grams of a given length from a list of tokens, an algorithm was used to iterate sequentially over the entire list, where each token was recursively associated with a one shorter n-gram starting at the succeeding token.

A variant with so-called *skip-n-grams* was also tested, where any number of skipped tokens could occur between tokens forming one n-gram. In this case, however, the number of valid n-grams increased exponentially and the efficiency of the whole metric decreased, therefore skip-n-grams were eventually not used.

2.2.6 Comparing n-grams

To compare n-grams in our implementation, we use an algorithm to traverse all n-gram pairs for both input texts. The aim is to compute the total overlap of n-gram similarities,

[6] https://extensions.libreoffice.org/en/extensions/show/czech-dictionaries
[7] https://fasttext.cc/docs/en/crawl-vectors.html
[8] https://ufal.mff.cuni.cz/morphodita
[9] https://ufal.mff.cuni.cz/udpipe

```
FOR gold_word IN gold_summary_words:
    FOR system_word IN system_summary_words:

        similarity = Word2vec.similarity(gold_word, system_word)

        words_are_antonyms =
            both words are numbers but not equal
               OR
            gold_word == "ne" + system_word
                OR
            system_word == "ne" + gold_word

        words_are_similar =
            similarity > 0.7
                AND
            words_are_antonyms == false

        words_are_synonyms =
            synonym_dictionary[system_word] CONTAINS gold_word
                OR
            synonym_dictionary[gold_word] CONTAINS system_word

        if words_are_synonyms OR words_are_similar:
            REPLACE system_word WITH gold_word
```

Figure 2.1. Pseudocode of the synonymization step used in ROUGE-CS.

which is calculated as the cumulative sum of the similarities of each n-gram in the system summarization with the most similar n-gram in the reference summarization, where the similarity of each pair is given by the sum of the similarities of the words on the same n-gram indices. The value for each n-gram pair is weighted by the chosen length of the n-gram and lies in the range between 0 and 1.

System n-grams for which a similar reference n-gram exists (with a similarity of at least 0.7) are counted positively in the total, while n-grams for which no major match was found (maximum similarity below 0.3) are counted negatively.

To calculate the similarity of each pair of words, a part-of-speech tagger is used to compare the word class. If the class is not equal or the words are antonyms, ROUGE-CS assigns the pair a similarity of 0. If the words are equivalent, a similarity of 1 is awarded. Otherwise, the similarity is computed as the cosine similarity of the word embeddings.

System n-grams that the metric has already seen before are counted negatively. This step serves to penalize overly repetitive expressions in the system summarization.

2.2.7 Calculating precision, recall, and f-measure

The output values of precision, recall, and f-measure are calculated for ROUGE-CS using the similar formulas used by the original ROUGE. The recall is equal to the proportion of accumulated similarity and the total number of reference n-grams. Precision is equal to the proportion of accumulated similarity and the total amount of system n-grams.

The resulting f-measure is calculated as follows:

$$F = 2 \cdot \frac{accumulated\ similarity}{N_{system\ ngrams} + N_{gold\ ngrams}}$$

2.3 Results

Finally, the resulting implementation of ROUGE-CS was compared with the original ROUGE, namely the Rouge-RAW [Straka, 2018][10] implementation. A list of manually annotated summarization pairs was used as the test dataset, which is described in more detail in section 1.3.

A comparison of the results is shown in Table 2.1.

Metric type	N-gram length	Correlation - dev	Correlation - test
Rouge-RAW	1-gram	0.187	0.169
	2-gram	0.166	0.170
	3-gram	0.139	0.133
	4-gram	0.141	0.140
ROUGE-CS	1-gram	0.374	0.359
Final	2-gram	0.278	0.256
	3-gram	0.225	0.220
	4-gram	0.197	0.184
ROUGE-CS	1-gram	**0.400**	**0.371**
without	2-gram	0.253	0.251
lowercase tokenization	3-gram	**0.240**	**0.226**
	4-gram	0.198	0.187
ROUGE-CS	1-gram	0.340	0.325
without	2-gram	**0.285**	**0.280**
stop-word removal	3-gram	0.183	0.185
	4-gram	0.192	**0.188**
ROUGE-CS	1-gram	0.246	0.225
without	2-gram	0.200	0.180
synonymization	3-gram	0.162	0.159
	4-gram	0.161	0.160
ROUGE-CS	1-gram	0.364	0.325
without	2-gram	0.284	0.267
MorphoDiTa lemmatizer	3-gram	0.226	0.212
(replaced by UDPipe)	4-gram	**0.204**	0.186

Table 2.1. Ablation study of ROUGE-CS and its comparison to Rouge-RAW.

2.4 Discussion

The table shows that ROUGE-CS correlates better with human-annotated pairs of summaries for all selected n-gram lengths (Rouge-1, Rouge-2, Rouge-3, Rouge-4), but the correlation is still at a relatively low level, below our initial expectations.

[10] `https://ufal.mff.cuni.cz/sumeczech`

The higher correlation values than the original metric are interpreted as that by comparing the similarity of the vector representations of individual words, ROUGE-CS can more correctly estimate the semantic meaning of both summarizations and does not focus only on their syntactic form.

The Rouge-L method is not available in our implementation due to the complexity of the individual steps of ROUGE-CS, which cannot be simply incorporated into the original algorithm that utilizes the dynamic programming[11] approach to optimize its performance.

A possible future improvement is the use of more sophisticated methods to determine the meaning of individual words in a given context, such as Named Entity Recognition [Lample, 2016], which can identify entities in text such as proper names, organization names, timestamps, and similar.

Further work may also involve an attempt to implement a method for identifying the longest common Rouge-L sequence that can utilize the techniques described above used by ROUGE-CS. However, there is the problem of finding an algorithm efficient enough to compare a large number of longer summaries.

[11] `https://brilliant.org/wiki/problem-solving-dynamic-programming/`

Chapter 3
Model-based metric (Memes-CS)

In the second part of this thesis, we propose Memes-CS (Metric for Evaluating Model Effectiveness in Summarization), a metric suitable for evaluating the quality of summarization based on a neural transformer model learned for logistic regression downstream task on a specifically designed dataset.

Unlike ROUGE-CS, our Czech implementation of the ROUGE algorithm, Memes-CS utilizes a complex neural architecture for its underlying operations and therefore its exact results may be considered non-deterministic.

3.1 Models

The powerful HuggingFace Transformers[1] and PyTorch[2] libraries are used to experiment with neural models and to construct our metric.

We tested and compared the following models:

3.1.1 XLM-RoBERTa

XLM-RoBERTa [Conneau, 2019] is a transformer model pretrained on a large English corpus using self-supervised Masked language modeling (MLM) task. Taking a sentence, the model randomly masks 15% of the words in the input then run the entire masked sentence through the model and has to predict the masked words, which allows the model to learn a bidirectional representation of the sentence. This way, the model learns an inner representation of the language that can then be used to extract features useful for downstream tasks.

The full Transformer architecture is depicted in Figure 3.1.

The model was designed to address several shortcomings that arose during the training of the original BERT model on which RoBERTa is based.

In this thesis, we specifically used a version pre-trained on the NLI-related SQuAD2 dataset [Rajpurkar, 2016] provided by HuggingFace[3].

3.1.2 CZERT

Czert [Sido, 2021] is the first Czech monolingual language representation model based on BERT [Devlin, 2018] and ALBERT [Lan, 2019] architectures, pretrained on more than 340K of sentences. During testing, Czert outperformed other multilingual models and reached state-of-the-art results on nine Czech datasets. In this thesis, we used the version provided by HuggingFace[4].

[1] https://huggingface.co/docs/transformers/index
[2] https://pytorch.org/
[3] https://huggingface.co/deepset/xlm-roberta-base-squad2
[4] https://huggingface.co/UWB-AIR/Czert-B-base-cased

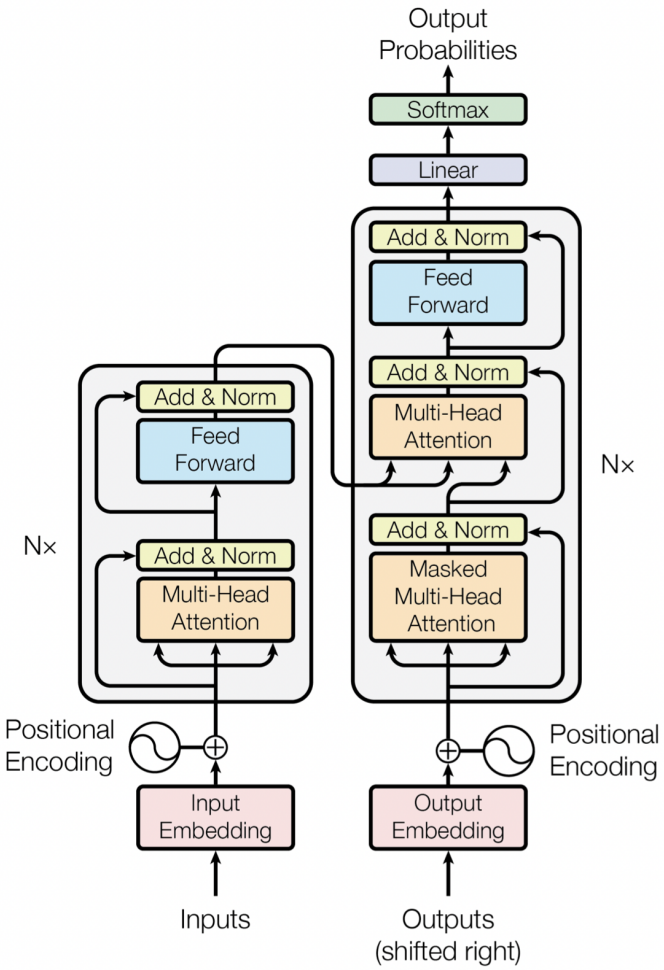

Figure 3.1. Transformer model architecture, reprinted from [Vaswani, 2017].

3.1.3 RobeCzech

RobeCzech [Straka, 2021] is a monolingual RoBERTa language representation model trained on Czech data, which achieved significantly higher success rates on several different NLP tasks than the compared Czech models. In this thesis, we used the version provided by HuggingFace[5].

3.2 Dataset

We trained our models on a combination of two datasets: the Czech NLI dataset CSFever and our custom, algorithmically generated dataset that we named the Grad Cortex (Gradual Corruption of Text).

Both parts of the final training dataset are described in more detail in following sections 3.2.1 and 3.2.2.

For evaluation, a smaller dataset consisting of several dozen manually annotated pairs of summarizations was used, which is described in more detail in section 1.3.

[5] `https://huggingface.co/ufal/robeczech-base`

3.2.1 CSFever

CSFever [Drchal, 2022] is a Czech NLI dataset designed for automatic verification of claims based on context, which is considered trusted ground truth. It was produced by translating the original English version of the dataset called FEVER [Thorne, 2018] using machine translation. Versions translated by Google Translate[6] and DeepL[7] were available, the former being used for the generation of our dataset. Individual records contain a context + query pair followed by a label having one of three possible values `SUPPORTS`, `REFUTES` or `NOT ENOUGH INFO`. A sample of CSFever data can be observed in Figure 3.2

id (string)	label (class label)	evidence (string)	claim (string)
170685	2 (SUPPORTS)	Lisabon. Jeho městská oblast přesahuje správní hranice města a žije v ní…	V Lisabonu žije více než 1 mld.
111602	0 (REFUTES)	Willie Nelson. Po návratu Nelson dva roky navštěvoval Baylorovu univerzitu, ale…	Willie Nelson zanechal studia po třech letech.
119264	2 (SUPPORTS)	AC/DC je australská rocková skupina, kterou v roce 1973 založili bratři…	Malcolm Young byl spoluzakladatelem australské hardrockové skupiny AC/DC.
106718	1 (NOT ENOUGH INFO)	"Nice & Slow" je singl z Usherova druhého alba My Way z roku 1998. Píseň napsali…	Nice & Slow je jazzový singl.
77712	2 (SUPPORTS)	Newfoundland a Labrador. Provincie je jazykově nejhomogennější v Kanadě, 97,6 …	Newfoundland a Labrador je jazykově nejhomogennější z celé Kanady.
184132	0 (REFUTES)	Furia je francouzské romantické drama z roku 1999, které natočil režisér…	Furia je adaptací povídky Anny Politkovské.

Figure 3.2. A sample of CSFever dataset, reprinted from HuggingFace[8].

Because enumeration is used to label the data for the NLI task, its direct use for logistic regression was not possible and a minor adaptation was necessary.

The original labels were mapped to decimal numbers, with the `SUPPORTS` label assigned a value of 1.0 and the other two labels assigned a value of 0.0.

We justify this modification by stating that in the case of `SUPPORTS` the statement is true, and hence any system summarization would in this case contain a subset of the information found in the reference summarization. In the other two cases, the system summarization would contain either outright false or irrelevant information. In both cases, we consider these summarizations incorrect. We also experimented with the possibility that the label `NOT ENOUGH INFO` would be converted to a value of 0.5, because the potential system information could still contain correct information from the original document.

The results of the metric model learned using the two methods mentioned above are listed in Table 3.2.

It can be observed from the table that mapping only to the extreme values of 0.0 and 1.0 results in slightly more promising results, and is therefore used for the final version of the dataset.

3.2.2 Grad Cortex

The second part of the final dataset is an algorithmically created partition called Grad Cortex (Gradual Corruption of Text). It is based on the dataset of Czech summaries

[6] https://translate.google.com/
[7] https://www.deepl.com/

SumeCzech [Straka, 2018], a large dataset containing over 1 million documents from Czech news portals, with each document divided into the headline, abstract, and text. The headline is the title of the article most often appearing in the uppermost part of the document, being distinctly separated from the rest of the text. The abstract is represented by the first paragraph, which is usually located below the headline and is differentiated by color from the rest of the document.

We use only the abstract of each record, which we consider as a reference summary for the text of the document.

Our Grad Cortex dataset is generated by gradually performing random transformations over each reference summarization in an attempt to distort its quality. Subsequently, the transformed summarization is assigned a label ranging from 0 to 1 depending on how much semantic change occurred during the transformations. A value of 1 represents a summarization equal to the reference, while a value of 0 represents a summarization containing major semantic differences.

Each type of transformation is heuristically assigned a different weight depending on how semantically significant the change is. For example, transformations that only reduce the readability of the text but did not in any way reduce its veracity were given a low weight. Even when such changes were repeated several times, the value of the assigned label did not fall below 0.5. On the other hand, transformations that introduced false information into the text resulted in a decrease of the label value even below 0.5, and if several such changes were made at the same time, the value of the label could drop to zero.

In addition to the severity of the change, the number of transformations performed or the ratio of the transformed part to the total length of the summarization contributed to the final label value.

The calibration of the weights of each type of transformation was done manually based on our judgment, guided by the question: *"What label would we assign to the summarization corrupted in such a manner?"*

3.3 Grad Cortex transformations

The specific types of transformations are listed in the following sections:

3.3.1 Gold summary

In this case, the transformed summary is just a copy of the reference one with no changes being made to it. The resulting pair of two equivalent summaries is always assigned a label value of 1.0, indicating a perfect score.

The reason for this type of transformation is to teach the model to recognize perfect summaries or summaries very similar to the reference one and to give these outputs the highest possible score.

3.3.2 Number swap

This transformation changes the numbers found in the summation to random numbers in a certain range. First, all integers are extracted from the summation using the `\d+` regex. For each is then generated a random number in the range from 0 to twice the original value and the original numbers are replaced.

An example of the transformation can be demonstrated in the following two texts:

Britskému princi Charlesovi (70) se na stará kolena zapalují lýtka. Krásné ženy po něm touží!

Britskému princi Charlesovi (53) se na stará kolena zapalují lýtka. Krásné ženy po něm touží!

The weight of the transformation depends on the number of changed numerals and the magnitude of each change according to the following formulas:

$$w_i = \frac{|n_i - n_i'|}{n_i}$$

$$w_T = \sum_{i=1}^{N} w_i$$

Where w_i is the weight of a single transformation, n_i is the original number, n_i' is the chosen replacement, N is the total number of numerals in the summary and w_T is the total weight of the transformations.

The label value for the resulting pair of summarizations is calculated as follows:

$$l = 1 - \frac{w_T}{N}$$

The reason for this type of transformation is to teach the model to respond to false numerical values in the system summarization.

3.3.3 Part of speech swap

This transformation replaces the words in the summarization with random words from the original document that have the same word class.

First, all word classes are extracted from the original document using a part-of-speech tagger. Then a key-value dictionary is constructed, where the key is represented by the word class, and the value is a list of all words of that word class occurring in the original document. Numeric values and words beginning with a capital letter are omitted since in these cases the words have most likely more important meaning.

Similarly, all pairs of words and their word types are extracted from the summarization. The individual words are then successively replaced by a random word of the same word type and ending with the same letter from the original document if such a word is available.

For tag extraction, we used MorphoDiTa tagger, an open-source tool for morphological analysis of natural language texts [Straka, 2014].

An example of the transformation can be demonstrated in the following two texts:

Britskému princi Charlesovi (70) se na stará kolena zapalují lýtka. Krásné ženy po něm touží!

Britskému kanci Charlesovi (70) se na stará polena zapalují lýtka. Krásné žáby po něm krouží!

The transformation weight w_T is equal to the number of swaps performed.

The label value for the resulting pair of summarizations is calculated as follows:

$$l = 1 - \min(1, \frac{2 \cdot w_T}{n})$$

Where n is the total amount of tagged words in the summary.

The reason for this type of transformation is to teach the model to recognize substitutions of less meaningful parts of the text for others from the original document.

3.3.4 Named Entity swap

This transformation replaces named entities from the summarization with random named entities from the selected text. In our metric, we used the original document for a given summarization and then a random document occurring in the corpus.

Similar to the part-of-speech swap section, all named entities along with their type are extracted from the selected text and summarization. Again, they are arranged in a dictionary and the named entities from the summarization are successively replaced by randomly selected named entities of the same type from the document. Numeric values are omitted.

For named entity extraction we used the web service NameTag, an open-source tool for named entity recognition (NER) [Straková, 2014][9].

An example of the transformation can be demonstrated in the following two texts:

Britskému princi Charlesovi (70) se na stará kolena zapalují lýtka. Krásné ženy po něm touží!

Maďarskému princi Olafovi (70) se na stará kolena zapalují lýtka. Krásné ženy po něm touží!

The transformation weight w_T is equal to the number of swaps performed.

The label value for the resulting pair of summarizations is calculated as follows:

$$l = 0.5 - \frac{w_T}{2 \cdot n}$$

Where n is the total amount of named entities in the summary.

The reason for this type of transformation is an attempt to teach the model to identify substitution of more meaningful parts of the text such as proper names, names of organizations, months of the year, etc.

3.3.5 Sentence swap

This transformation replaces the sentences from the summarization with random sentences from the selected text. The reference and then a random extract from the original document are used as the summarization. As the selected text, the original document and then a random document occurring in the corpus are used.

First, the summarization and the selected text are divided into a list of individual sentences. To do this, the regex `;|\.|\?|!` is used, splitting the text according to the most common sentence separators. Subsequently, the individual sentences in the summarization are replaced with random sentences from the selected text.

An example of the transformation can be demonstrated in the following two texts:

Britskému princi Charlesovi (70) se na stará kolena zapalují lýtka. Krásné ženy po něm touží!

Britskému princi Charlesovi (70) se na stará kolena zapalují lýtka. Ještě aby ne, vypadá tak mladě!

The transformation weight is equal to the ratio of the sum of the lengths of the original sentences that were transformed to the total original length of the summarization:

[9] `http://lindat.mff.cuni.cz/services/nametag/`

$$w_T = \frac{\sum_{s \in S} length(s)}{\text{original summary length}}$$

Where S is the list of swapped sentences.

If a reference summarization and the original document are used, the label value for the resulting pair of summarizations is calculated as follows:

$$l = 1 - \frac{w_T}{2}$$

If a random extract is used as the summarization and a random document from the corpus is used as the selected text, the label value for the resulting pair of summarizations is calculated as follows:

$$l = 0.5 - \frac{w_T}{2}$$

The formulas show that a random extract from the original document always has a label value equal to 0.5 and a random extract from a random document always has a value of 0.

The reason for this type of transformation is to teach the model to recognize summaries that were either created as a random extract from the original document, or that contain nothing semantically related to the original document.

3.3.6 Named Entity removal

This transformation removes named entities from the summarization.

First, all named entities are extracted from the summarization along with their type and then sequentially replaced with an empty string. Numeric values are omitted.

For named entity extraction we used the web service NameTag, an open-source tool for named entity recognition (NER) [Straková, 2014][10].

An example of the transformation can be demonstrated in the following two texts:

Britskému princi Charlesovi (70) se na stará kolena zapalují lýtka. Krásné ženy po něm touží!

** princi * (70) se na stará kolena zapalují lýtka. Krásné ženy po něm touží!*

The transformation weight w_T is equal to the number of named entities removed.

The label value for the resulting pair of summarizations is calculated as follows:

$$l = \frac{1}{2 \cdot w_T + 0.5} + 0.5$$

The reason for this type of transformation is to try to teach the model that removing a named entity should have less effect on the meaning of the summation than swapping it for another.

[10] `http://lindat.mff.cuni.cz/services/nametag/`

3.3.7 Sentence removal

This transformation removes sentences from the summarization.

First, the summarization is divided into a list of individual sentences. To do this, the regex `;|\.|\?|!` is used, splitting the text according to the most common sentence separators. Then the individual sentences in the summarization are sequentially replaced by an empty string. At least one sentence is always left in the summarization.

An example of the transformation can be demonstrated in the following two texts:

Britskému princi Charlesovi (70) se na stará kolena zapalují lýtka. Krásné ženy po něm touží!

Britskému princi Charlesovi (70) se na stará kolena zapalují lýtka. *

The transformation weight is equal to the ratio of the sum of the lengths of the sentences that were removed to the total original length of the summarization:

$$w_T = \frac{\sum_{s \in S} length(s)}{\text{original summary length}}$$

Where S is the list of swapped sentences.

The label value for the resulting pair of summarizations is calculated as follows:

$$l = 1 - \frac{w_T}{2}$$

The reason for this type of transformation is to try to teach the model that removing a sentence should not affect the meaning of the summation as much as swapping it for another.

3.3.8 Synonym replace

This transformation replaces the words in the summarization with their synonyms.

First, words with their corresponding lemma are extracted from the summarization using a lemmatizer. Then, for each lemma, a random synonym is selected from the dictionary to replace the original word in the summarization.

For lemma extraction, we used the MorphoDiTa tagger, an open-source tool for morphological analysis of natural language texts [Straka, 2014].

As a synonym dictionary was used the Dictionary of spell checking, word splitting, and synonyms for Czech, which is available as an add-on for LibreOffice[11]. Unfortunately, the dictionary contains only the basic forms of words, so it is not possible to replace words with synonyms in the correct form. You can read more about its use in section 2.2.3.

An example of the transformation can be demonstrated in the following two texts:

Britskému princi Charlesovi (70) se na stará kolena zapalují lýtka. Krásné ženy po něm touží!

Britskému princi Charlesovi (70) se na starobylý kolena zapalují lýtka. Sexy ženy po něm prahnout!

The transformation weight w_T is equal to the number of word changes made.

The label value for the resulting pair of summarizations is calculated as follows:

[11] `https://extensions.libreoffice.org/en/extensions/show/czech-dictionaries`

$$l = 1 - \frac{w_T}{2 \cdot n}$$

Where n is the total number of words in the summarization.

The reason for this type of transformation is to teach the model that replacing words with their synonyms should have less effect on the meaning of the summarization than replacing them with words of a different meaning.

3.3.9 Antonym replace

This transformation replaces the words in the summarization with their antonyms.

Similar to synonym substitution, words with their corresponding lemmas are extracted from the summarization using a lemmatizer. Then, for each lemma, a synonym is selected from the dictionary, to which the addition of one of the prefixes "ne" or "proti" produces a grammatically correct, existing word. The original word is then replaced by this word in the summarization, assuming that the two words have opposite meanings to each other.

An example of the transformation can be demonstrated in the following two texts:

Britskému princi Charlesovi (70) se na stará kolena zapalují lýtka. Krásné ženy po něm touží!

Britskému princi Charlesovi (70) se na stará kolena nezapalují lýtka. Krásné ženy po něm nedychtit!

The transformation weight w_T is equal to the number of word changes made.

The label value for the resulting pair of summarizations is calculated as follows:

$$l = 1 - \min(1, \frac{4 \cdot w_T}{n})$$

Where n is the total number of words in the summarization.

The reason for this type of transformation is to try to teach the model that replacing words with their antonyms should have a greater effect on the meaning of the summarization than replacing them with random words or synonyms.

3.4 Comparing the dataset partitions

Figure 3.3 shows the training process of the XLM-RoBERTa-Large model pre-trained on the SQUAD 2 dataset. Specifically, it is learning on three variants of our dataset: the red curve represents learning only on the CSFever dataset, and the gray curve represents learning only on the algorithmically generated Grad Cortex dataset. The green curve represents learning on the final dataset, generated by composing the two mentioned partitions together.

Table 3.2 shows that the model trained on the final, combined dataset shows the most promising results during evaluation. The model trained solely on the CSFever dataset shows a sharper increase in correlation at first, but after a few thousand steps, overfitting is already evident and the success rate drops rapidly. The model trained solely on the Grad Cortex dataset shows the lowest success rate. We speculate that the reason for this is the imperfection of the deterministic algorithm for generating

Figure 3.3. The learning phase of Memes-CS model on different parts of the final dataset.

transformations, which cannot always produce grammatically correct and meaningful text due to its limitations.

Table 3.1 shows the distribution of the final dataset used to train the Memes-CS metric. The ratio between the CSFever and Grad Cortex dataset parts is roughly 1:1. The manually annotated dataset mentioned in Section 1.3, which is used for evaluation and testing purposes, is also shown in the table.

Dataset	Number of pairs
Final (CSFever + Grad Cortex)	408,346
CSFever	208,346
Grad Cortex	200,000
Correlation (Evaluation)	78
Correlation (Test)	26

Table 3.1. Distribution and sizes of datasets used in learning phase of Memes-CS metric.

3.5 Training

We found the transformer model with logistic regression head on top to be the most appropriate learning method because for each input text (in our case, a pair of reference and system summaries), the model output is a single decimal number, in our case representing the level of similarity of the two summaries.

Due to the large size of the dataset and to reduce the possibility of overfitting, the batch size was fixed at 32 and the maximum gradient size was limited to 2.0. A weight decay of 0.01 was introduced when learning the model.

For each model and each dataset version, an output directory path is generated where the highest-ranking checkpoints of the learned models can be found:

```
[dataset name]/[model name]\_bs[batch size]\_lr[learning rate]\_s[seed]
```

An existing tokenizer predefined for each pre-trained model is always used to tokenize the dataset. Furthermore, a data collator is used to ensure correct padding of the input

texts, ensuring compatibility with the model architecture. Although no maximum input text length is set, the trained models based on the BERT architecture have an implicit maximum input length of 512 tokens. In our case, the input elements are only pairs of short summaries separated by a special token, therefore their length does not exceed this limit, avoiding summarization pruning that could result in a loss of semantic information and thus negatively affect the models' performance.

AdamW [Kingma, 2014] is used as an optimizer and 10

Although the number of epochs is statically set to 20, overfitting is observed in all models, and therefore learning typically occurs in only 2-3 epochs.

After every 300 steps, the model is evaluated and saved. As the evaluation dataset, a subset of the manually annotated pairs of summaries from the Evaluating proposed metrics chapter is selected, comprising 100 text pairs. In each round of evaluation, the model with a higher degree of correlation with the human annotations is selected.

The portal Weights & Biases was used to log the training phase[12].

The course of the training phase of each model is presented in Figure 3.4.

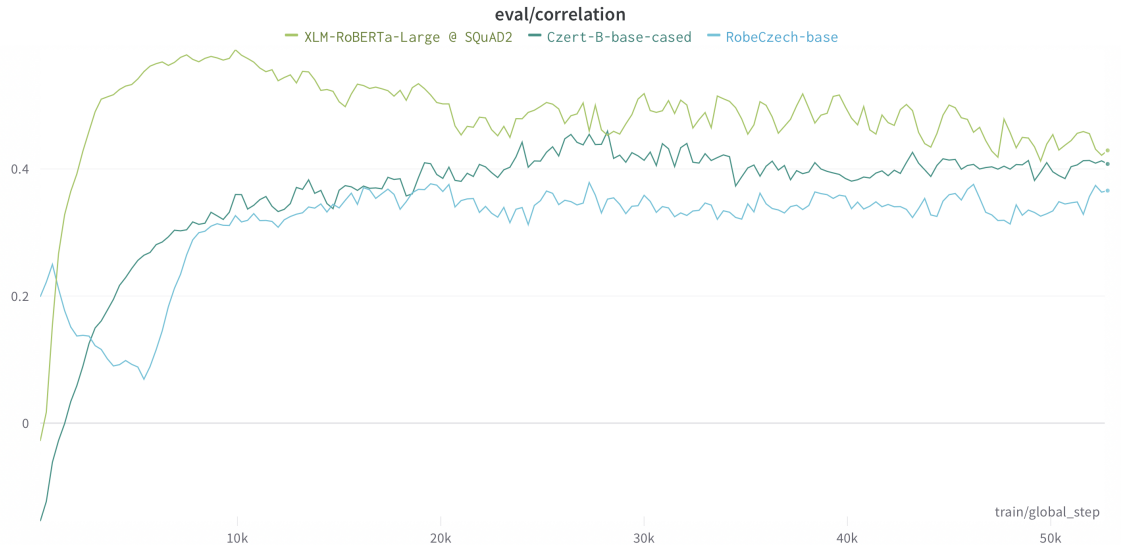

Figure 3.4. The learning phase of Memes-CS models on the final dataset.

It can be observed from the plot that the majority of the models achieved the most favorable results after approximately ten thousand steps, with the highest correlation level of 0.588 being achieved by the XLM-RoBERTa-Large model pre-trained on the SQuAD2 dataset after 9900 steps.

3.6 Results

Table 3.2 presents the correlation values of the tested models with a randomly preselected subset of the human-annotated text pairs introduced in section 1.3. Models did not encounter this particular data during the evaluation phase.

Histograms 3.5 and 3.6 show the distribution of system labels for data annotated as `GOOD` where higher label values are expected and data annotated as `BAD` where lower label values are expected.

[12] `https://wandb.ai/`

Metric type	Correlation - dev	Correlation - test
Memes-CS Final	**0.588**	**0.521**
Memes-CS with NEI label converted to 0.5	0.542	0.501
Memes-CS learned only on CSFever dataset	0.533	0.489
Memes-CS learned only on Grad Cortex dataset	0.407	0.395
Memes-CS based on Czert model	0.459	0.437
Memes-CS based on RobeCzech model	0.400	0.381
ROUGE-CS (1-gram) Final	0.374	0.359
Rouge-RAW (1-gram) Final	0.187	0.169

Table 3.2. Ablation study of Memes-CS metric and its comparison to ROUGE.

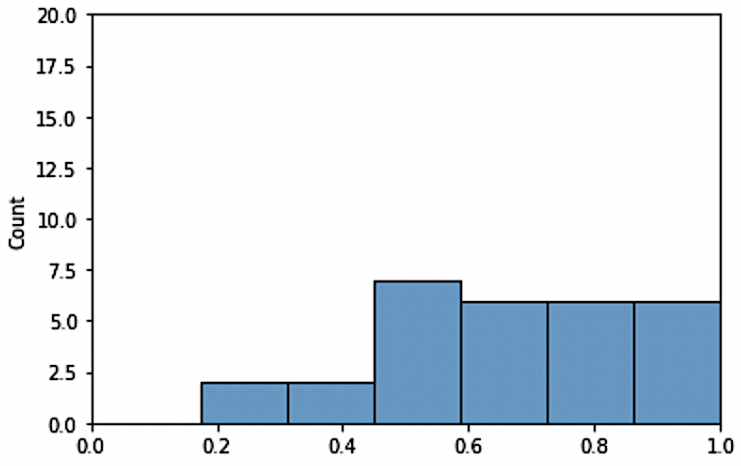

Figure 3.5. Histogram of labels for pairs annotated as `GOOD`.

From the histograms, we can see that the distribution of system labels roughly matches human expectations. Summaries annotated as `GOOD` were never given a score close to zero by our model, however, some summaries annotated as `BAD` were mistakenly scored as near perfect by our model.

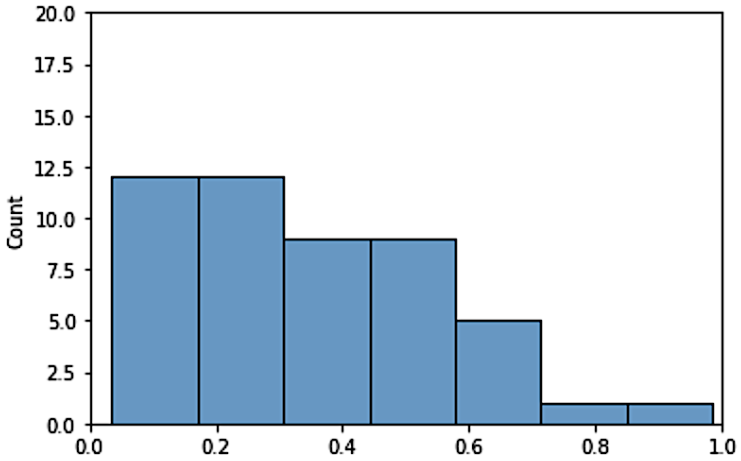

Figure 3.6. Histogram of labels for pairs annotated as `BAD`.

3.7 Discussion

Table 3.2 shows that RoBERTa scored the best of all models by a significant margin, and therefore we decided to use it in the first version of Memes-CS. The Czert model ranked second in the evaluation with a significantly lower score, and surprisingly, the lowest score was achieved by the RobeCzech model, which we originally expected to outperform Czert, judging by the results presented in its paper.

Given that Memes-CS is based on machine learning, which is rapidly advancing, there is no guarantee that a model will not be discovered in the near future that achieves a higher score on our dataset than the XLM-RoBERTa model we tested. Also, improvements to the metric could occur in the future simply by adjusting the hyper-parameters or optimizing the underlying algorithms. For this reason, new versions of our metric may emerge in the future, which will need to be distinguished from each other by choosing a unique label. Thus, the final version of our metric will be labeled `Memes-CS_1.0` to indicate that it is the first public version. The learned model is available on HuggingFace[13].

[13] `https://huggingface.co/SimonZvara/Memes-CS_1.0`

Chapter 4
Conclusion

In this thesis, we proposed two new metrics for evaluating the performance of summarization models in terms of facticity. In the first part, we introduced ROUGE-CS, an implementation of a deterministic evaluation algorithm based on the ROUGE metric. Our metric merges several different methods of text transformations and n-gram matching at the semantic level. It operates with Word2Vec word embeddings, several dictionaries of Czech terms, synonyms, antonyms, and stop words. It also utilizes algorithms for tokenization, lemmatization, and identification of word classes in text. With these advanced methods, ROUGE-CS can better understand the overall semantics of summaries and thus better detect factual errors and other deficiencies than the original metric.

In the second part of this thesis, we focused on creating a custom dataset and subsequent implementation of Memes-CS, a non-deterministic metric utilizing a neural model based on the BERT architecture. Our dataset combines existing Czech datasets used in training related NLI models and our own algorithmically produced data. The generation is conducted through successive transformations of the SumeCzech dataset by chaining several different transformation techniques, including named entity swapping, sentence swapping, number swapping, parts-of-speech swapping, lemmatization, removal of sentences, removal of named entities, synonym and antonym replacements. Our model-based metric is more successful at emphasizing the semantic meaning of summarizations and does not focus solely on syntactic deviations, as is the case with the original ROUGE metric. Our model-based metric correlates more strongly with human-annotated data than the competing ROUGE, and we believe it marks a major step forward in solving the problem of finding an appropriate method for evaluating the performance of summarization models in terms of facticity.

The implementation of the metrics was preceded by the study of scientific literature on advances in the areas of abstract and extractive summarization, appropriate model architectures, evaluation of summaries, internet fact-checking, and the verification of claims in a ground truth context.

References

Ali Bou Nassif, Ismail Shahin, Imtinan Attili, Mohammad Azzeh, and Khaled Shaalan. Speech recognition using deep neural networks: A systematic review. *IEEE access*. 2019, 7 19143–19165.

Iqbal H Sarker. Machine learning: Algorithms, real-world applications and research directions. *SN Computer Science*. 2021, 2 (3), 1–21.

Jiajun Zhang, Chengqing Zong, and others. Deep Neural Networks in Machine Translation: An Overview.. *IEEE Intell. Syst.*. 2015, 30 (5), 16–25.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Learning to compose neural networks for question answering. *arXiv preprint arXiv:1601.01705*. 2016,

Diyah Puspitaningrum. *A Survey of Recent Abstract Summarization Techniques.* In: *Proceedings of Sixth International Congress on Information and Communication Technology*. 2022. 783–801.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. *Pegasus: Pre-training with extracted gap-sentences for abstractive summarization.* In: *International Conference on Machine Learning*. 2020. 11328–11339.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*. 2019,

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*. 2017, 30

Chin-Yew Lin. *Rouge: A package for automatic evaluation of summaries.* In: *Text summarization branches out*. 2004. 74–81.

Michael Struwig. *What is a skipgram?* 2019. `https://www.notsobigdatablog.com/2019/01/02/what-is-a-skipgram/`.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*. 2013, 26

Jun-Ping Ng, and Viktoria Abrecht. Better summarization evaluation with word embeddings for rouge. *arXiv preprint arXiv:1508.06034*. 2015,

Maxime Peyrard, Teresa Botschen, and Iryna Gurevych. *Learning to score system summaries for better content selection evaluation..* In: *Proceedings of the Workshop on New Frontiers in Summarization*. 2017. 74–84.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*. 2018,

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*. 2019,

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M Meyer, and Steffen Eger. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. *arXiv preprint arXiv:1909.02622*. 2019,

Elizabeth Clark, Asli Celikyilmaz, and Noah A Smith. *Sentence mover's similarity: Automatic evaluation for multi-sentence texts.* In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2019. 2748–2760.

Thomas Scialom, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. Answers unite! unsupervised metrics for reinforced summarization models. *arXiv preprint arXiv:1909.01610*. 2019,

Oleg Vasilyev, Vedant Dharnidharka, and John Bohannon. Fill in the BLANC: Human-free quality estimation of document summaries. *arXiv preprint arXiv:2002.09836*. 2020,

Yang Gao, Wei Zhao, and Steffen Eger. SUPERT: Towards new frontiers in unsupervised evaluation metrics for multi-document summarization. *arXiv preprint arXiv:2005.03724*. 2020,

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. *Bleu: a method for automatic evaluation of machine translation.* In: *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 2002. 311–318.

Maja Popovic. *chrF: character n-gram F-score for automatic MT evaluation.* In: *Proceedings of the Tenth Workshop on Statistical Machine Translation*. 2015. 392–395.

Alon Lavie, and Abhaya Agarwal. *METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments.* In: *Proceedings of the second workshop on statistical machine translation*. 2007. 228–231.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. *Cider: Consensus-based image description evaluation.* In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015. 4566–4575.

Alexander R Fabbri, Wojciech Kryscinski, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*. 2021, 9 391–409.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*. 2019,

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*. 2020, 8 726–742.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*. 2013,

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*. 2017, 5 135–146.

Milan Straka, and Jana Straková. MorphoDiTa: Morphological dictionary and tagger. 2014,

Milan Straka, Jan Hajic, and Jana Straková. *UDPipe: trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, pos tagging and parsing.* In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. 2016. 4290–4297.

Milan Straka, Nikita Mediankin, Tom Kocmi, Zdeněk Žabokrtský, Vojtěch Hudeček, and Jan Hajic. *Sumeczech: Large czech news-based summarization dataset.* In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. 2018.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*. 2016,

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*. 2016,

Jakub Sido, Ondřej Pražák, Pavel Přibáň, Jan Pašek, Michal Seják, and Miloslav Konopík. Czert–Czech BERT-like Model for Language Representation. *arXiv preprint arXiv:2103.13031*. 2021,

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*. 2019,

Milan Straka, Jakub Náplava, Jana Straková, and David Samuel. *RobeCzech: Czech RoBERTa, a monolingual contextualized language representation model.* In: *International Conference on Text, Speech, and Dialogue*. 2021. 197–209.

Jan Drchal, Herbert Ullrich, Martin Rýpar, Hana Vincourová, and Václav Moravec. CsFEVER and CTKFacts: Czech Datasets for Fact Verification. *arXiv preprint arXiv:2201.11115*. 2022,

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*. 2018,

Jana Straková, Milan Straka, and Jan Hajic. *Open-source tools for morphology, lemmatization, POS tagging and named entity recognition.* In: *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. 2014. 13–18.

Diederik P Kingma, and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. 2014,

Appendix A
Acronyms

List of acronyms appearing in this thesis:

ALBERT	A Lite BERT
BERT	Bidirectional Encoder Representations from Transformers
FEVER	Fact Extraction and Verification – series of Shared tasks focused on fact-checking
Grad Cortex	Gradual Corruption of Text
MEMES	Metric for Evaluating Model Effectiveness in Summarization
NEI	Not Enough Information
NER	Named Entity Recognition
NLI	Natural Language Inference
ROUGE	Recall-Oriented Understudy for Gisting Evaluation