Master Thesis

Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Department of Control Engineering

NLI Models for Assessing Facticity in Summarization Methods

Bc. Jan Dusil

Supervisor: Ing. Jan Drchal, Ph.D.
August 2022

MASTER‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name:	**Dusil Jan**	Personal ID number:	**457007**
Faculty / Institute:	**Faculty of Electrical Engineering**		
Department / Institute:	**Department of Control Engineering**		
Study program:	**Cybernetics and Robotics**		
Branch of study:	**Cybernetics and Robotics**		

II. Master’s thesis details

Master’s thesis title in English:

NLI Models for Assessing Facticity in Summarization Methods

Master’s thesis title in Czech:

Použití NLI modelů pro ověřování fakticity sumarizací

Guidelines:

The task is to:
1) Research state-of-the-art NLP approaches for text summarization, focusing on evaluation methods for assessment of generated summaries. Explore mainly model-based approaches.
2) Discuss available options for the Czech language.
3) Experiment with the NLI models developed at AIC as a module of the full fact-checking pipeline. Finetune or train new models, if needed. Work with CsFEVER and CTKFacts or other datasets supplied by the supervisor.
4) Assess the correlation of the NLI models with other approaches like ROUGE

Bibliography / sources:

[1] Drchal, Jan et al. "CsFEVER and CTKFacts: Czech Datasets for Fact Verification" arXiv preprint arXiv:2201.11115 (2022).
[2] Puspitaningrum, Diyah. "A Survey of Recent Abstract Summarization Techniques." Proceedings of Sixth International Congress on Information and Communication Technology. Springer, Singapore, 2022.
[3] Fabbri, Alexander R., et al. "Summeval: Re-evaluating summarization evaluation." Transactions of the Association for Computational Linguistics 9 (2021): 391-409.
[4] Cao, Meng, et al. "Factual error correction for abstractive summarization models." arXiv preprint arXiv:2010.08712 (2020).
[5] Kryściński, Wojciech, et al. "Evaluating the factual consistency of abstractive text summarization." arXiv preprint arXiv:1910.12840(2019).

Name and workplace of master’s thesis supervisor:

Ing. Jan Drchal, Ph.D. Artificial Intelligence Center FEE

Name and workplace of second master’s thesis supervisor or consultant:

Date of master’s thesis assignment: **09.02.2022** Deadline for master's thesis submission: **15.08.2022**

Assignment valid until:
by the end of summer semester 2022/2023

Ing. Jan Drchal, Ph.D. Supervisor’s signature	prof. Ing. Michael Šebek, DrSc. Head of department’s signature	prof. Mgr. Petr Páta, Ph.D. Dean’s signature

III. Assignment receipt

The student acknowledges that the master’s thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master’s thesis, the author must state the names of consultants and include a list of references.

.

Date of assignment receipt

Student’s signature

Acknowledgements

First and foremost, I would like to express my high gratitude to my supervisor Ing. Jan Drchal, Ph.D. for the great support and provided expertise throughout my thesis and his kindness during the times I was struggling.

Secondly, I would like to acknowledge my parents, Marek and Doris Dusilovi. My friends Jan Buchtík, David Kopecký, Stanislav Zdrůbek, Martin Kubala, Adam Born, Marek Mikeš a Jakub Sommer for help with annotating datasets as well as moral support throughout.

Last but not least, I acknowledge the support of the OP VVV-funded project CZ.02.1.01/0.0/0.0/16_019/0000765 "Research Center for Informatics".

This thesis is dedicated to David Kopecký and my grandmother Dana, without whom I would not be able to push myself and finish it.

Declaration

I declare that I have worked on my thesis separately and that I have listed all the information sources used in accordance with a Methodical Guideline on Ethical Principles in Preparation college final thesis.

In Prague, 15. August 2022

Abstract

In recent years, neural networks, namely the Transformers architecture, have been dominating the field of Natural Language Processing. This approach is showing state-of-the-art results, and the field is progressively developing. One of these fields is the **abstractive text summarization**. However, feeding the models based on Transformers calls for the need for large datasets. Moreover, the field is mainly advancing in the most-used languages like English, Spanish or Chinese.
This master thesis presents an overview of state-of-the-art NLP approaches, with a focus on text summarization. We discuss the challenges and motivation for the task in the environment of the Czech language. In the practical part, we have created a custom annotated dataset and developed an NLI-fact-checking pipeline to test and evaluate the performance of selected NLI models to assess the facticity of generated summaries.
As the result of this thesis, we have presented a compact summary of the state-of-art in text summarization. In addition, the results of the NLI-fact-pipeline discover that with a suitable dataset that the NLI models have great potential of being an automatic model-based evaluation medium.

Keywords: NLP, NLI, BERT, Summarization, Abstractive summarization, Facticity verification, SumeCzech

Supervisor: Ing. Jan Drchal, Ph.D.
Centrum umělé inteligence FEL,
Karlovo náměstí 13,
Praha 2

Abstrakt

V posledních letech neuronové sítě, konkrétně Transformers architektura dominují pole Natural Language Processing. Tento způsob modelování jazyka vykazuje state-of-art výsledky a posouvá celý obor k rychlejšímu vývoji. Vyjímkou není ani **abstraktivní sumarizace** textu. Transformers architektura a modely založené na ni ovšem také přináší určitá úskalí a výzvy. Obor v současné chvíli nejvíce postupuje pro nejvíce užívané jazyky jako je angličtina, španělština a čínština. Zjeména kvůli dostupnosti datasetů skoro výhradně pro tyto jazyky.
Tato práce ukazuje přehled state-of-art přístupů v oblasti NLP se soustředěním na sumarizaci textu. Dále jsou diskutovány výzvy a překážky v prostředí sumarizace pro český jazyk. V praktické části je vytvořen vlastní anotovaný dataset and vytvořen program pro automatickou evaluaci NLI modelů na vytvořených sumarizacích.
Výsledkém práce je kompatní shrnutí state-of-art v oblasti automatické sumarizace textu. Dále, jsou prezentovány výsledky evaluace použití NLI modulů se zjištěním, že v případě použití vhodných a datasetů NLI modely ukazují velký potenciál stát se vhodnou metrikou pro ověřování generovaných sumarizací.

Klíčová slova: NLP, NLI, BERT, BART, Sumarizace, Abstraktivní sumarizace, Ověřování fakticity, SumeCzech

Překlad názvu: Použití NLI modelů pro ověřování fakticity sumarizací

Contents

Figures

Tables

Chapter 1

Introduction

Finance, healthcare, e-commerce, and military, to name a few; in the recent two decades, artificial intelligence (AI) has rapidly entered and influenced almost every area one may imagine. The motivation for the subsequent adoption of AI is obvious - generally reduced cost and increased efficiency whenever it is deployed. Even though the prices of computational power increased during the recent pandemic of COVID-19, the adoption of AI has skyrocketed [27]. There are examples like virtual assistants or fashion where AI has already brought drastic changes in terms of automation [36]. With many applications of AI rising, there are already numerous disciplines hidden behind the term. One of these disciplines is natural language processing or NLP. In this thesis, I studied one of the NLP fields - automatic text

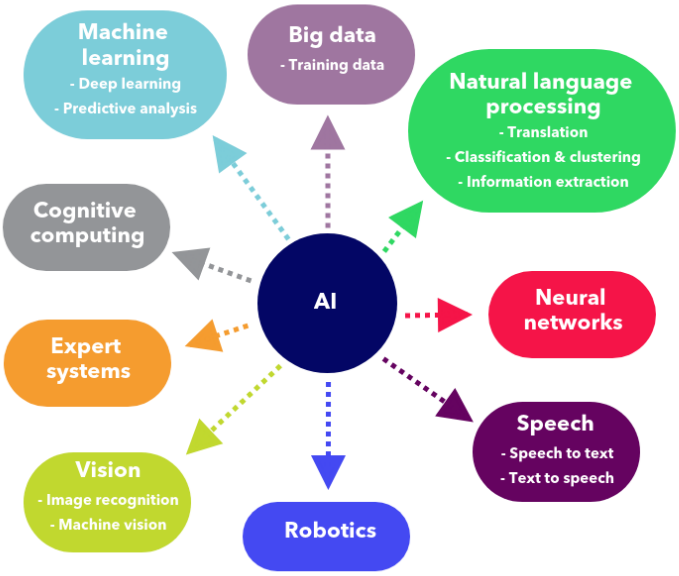


Figure 1.1: Illustrative picture - core disciplines of AI.

summarization. Automatic text summarization is one of the problems in AI, how to generate a condensed version of an input text that includes the essential parts. The motivation behind automatic text summarization and NLP generally comes from the fact that we live in the era of massive amounts of digital data being generated every second. Therefore, there is a rising need for such tools to process and evaluate this data [3]. Common approaches to automatic summarization are extractive, abstractive, and hybrid [37].

Despite such a demand for automatic text summarization and "significant efforts made by the research community, there are still many challenges limiting progress in summarization" [18]. One of the challenges is the lack of sufficient evaluation protocols. In addition, the current evaluation protocols mainly focus on fact-checking generated summaries more than evaluating the factual consistency of the summary. This thesis mainly focuses on abstractive text summarization and the utilization of NLI models for their assessment. In the theoretical 2 we break down the NLP field to general subtask and introduce them briefly. Then the focus is shifted towards summarization and NLI. This part highlights and summarizes state-of-the-art methods and abstractive summarization techniques and provides background for the current state of evaluation of automatic summarizations. The task of NLI is briefly defined and introduced.

In the practical part 4, we go over the workspace provided by the Artificial Intelligence Center (AIC) with the collaboration of Czech Technical University in Prague, and we list used software in the process of implementation. Moreover, we present a custom annotated dataset and an NLI-fact-checking pipeline for the assessment of automatically generated datasets.

In the last part 5, we present the results and evaluations of five picked NLI models that were used to assess generated headlines and summaries on two Czech datasets and discuss the outcomes.

Chapter 2

Background

This chapter provides a theoretical background for the practical part of this thesis. Broad research on NLP is presented; in addition, we list a summary of state-of-the-art summarization methods, models, datasets and evaluation methods.

2.1 Natural language processing

Nowadays, the NLP is one of the core disciplines of AI. It concerns computers' ability to understand texts and spoken words in much the same way human beings can. The motivation behind NLP is simple - automatize the analysis of textual or voice data. At an unmeasurably rapid pace, new textual and voice data are being generated every second. The volume of these data is so high that it is practically impossible for people to analyze them, categorize them or extract relevant information effectively. Moreover, these data are often entirely unstructured, which adds to the complexity of the task.

The study of NLP began around the 1950s as an intersection of AI and linguistics. At the beginning of NLP, the central area was text translation with simplistic word-to-word approaches. "The earliest NLP applications were hand-coded, rules-based systems that could perform certain NLP tasks, but could not easily scale to accommodate a seemingly endless stream of exceptions or the increasing volumes of text and voice data" [17] [23]. The failure of these methods highlighted some of the obstacles that NLP research faces even today. "Homonyms, homophones, sarcasm, idioms, metaphors, grammar and usage exceptions, and variations in sentence structure—these are just a few of the irregularities of human language that take humans years to learn but that programmers must teach natural language-driven applications to recognize and understand accurately from the start, if those applications are going to be useful" [17].

These approaches were highly limited in terms of obtaining the text semantics, which is a key factor for efficient automatic text evaluation and understanding. It was the merge with machine learning (ML) probabilistic methods that gave birth to statistical NLP during the 1980s. The statistical approaches generally have better results and are more robust; however, the need for large annotated datasets and more computational power rose from emerging in this direction. In recent years the research has moved rapidly, overcoming these obstacles. The increased availability of large-scale datasets and the increased availability of computational power (memory and speed) has shifted the field toward utilizing neural network architectures, which have been proven as the state-of-art approach for the whole NLP spectrum of tasks [43].

Nowadays, NLP is much more than just machine text translation. It has split into many subbranches and is considered one of the most advanced fields of AI [30].

2.1.1 Tasks

The field of NLP has many branches, and it is rather complicated to find a unified summary of those branches. As a part of the state-of-art research for this thesis we list the tasks in this subsection to provide a better understanding and a logical structure of the field. The presented summary is based on a wide variety of sources several sources. An illustrative overview is presented in the figure 2.1. The section is logically divided into text and audio tasks.

The main NLP tasks or use cases of NLP subjected to this thesis are **natural language inference (NLI)** and **abstractive text summarization (ATS)**, which are later discussed in more detail in separate sections 2.2 and 2.3.

Text tasks

- **Fill-Mask** - or masked language modelling is the task of predicting which words should replace artificially created masks in a sentence. "These models are useful when we want a statistical understanding of the language in which the model is trained in" [8].
  In 2.2 an example of the performance of the fill-mask model distilroberta-base is presented [38]. The output shows different possible values with a specific score. The higher the score, the more suitable value according to the model.

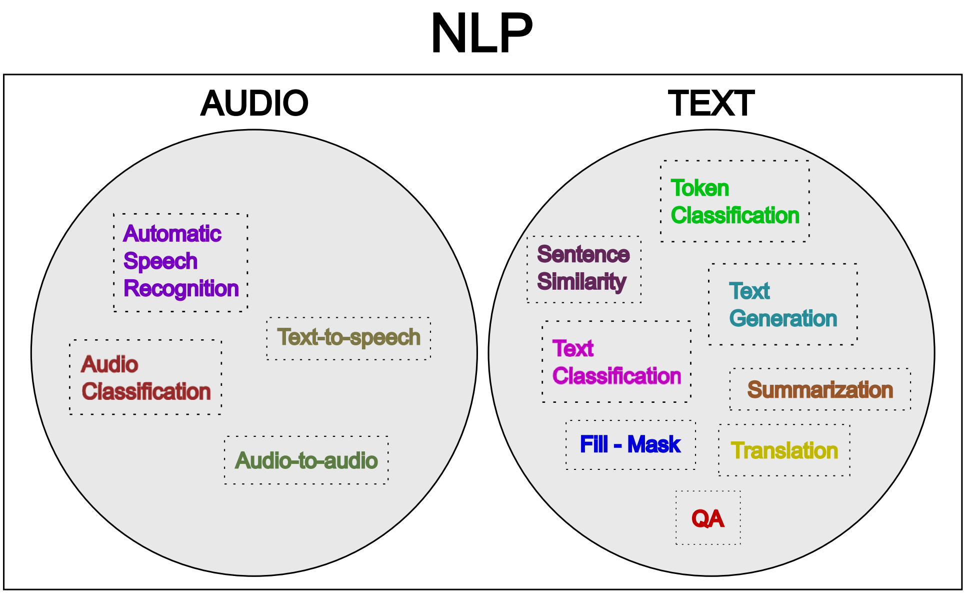
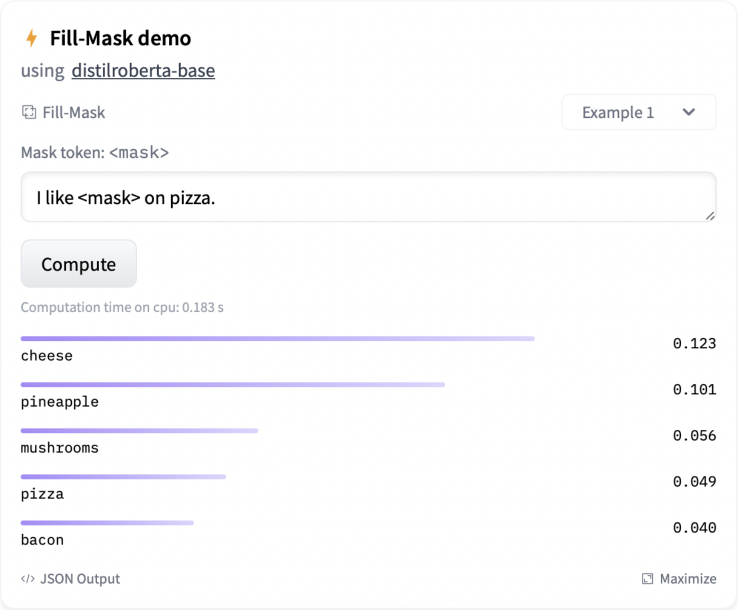


Figure 2.1: Illustrative picture - disciplines of NLP.

Use cases:



using distilroberta-base

Mask token: <mask>

I like <mask> on pizza.

Computation time on cpu: 0.183 s

```
cheese       0.123
pineapple    0.101
mushrooms    0.056
pizza        0.049
bacon        0.040
```

Figure 2.2: Fill-mask model performance example.

- Domain adaptation - masked language models do not require labelled data, which can be used as an advantage in finetuning models for domain-specific problems. Masked language models are trained simply by masking words into sentences, and the model is expected

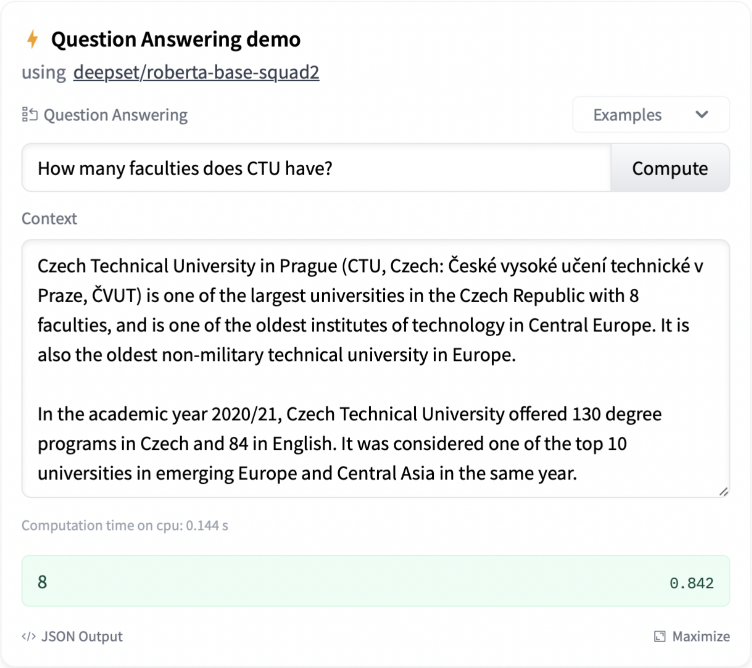
to guess the masked word. The most significant advantage is that such a model, trained on domain-specific data such as automotive research papers, has a statistical understanding of the language used in this domain.
That is a convenient trait which can be later used to finetune models solving different tasks like **Text Classification** or **Question Answering** for a selected domain [8].

- **Question Answering** - question answering (QA) is a task of correctly answering an input question. Notably, unlike search engines, QA models "present only the requested information instead of searching full documents like a search engine. As information in day-to-day life is increasing, so to retrieve the exact fragment of information even for a simple query requires large and expensive resources" [21].
  The QA task can be divided into two categories - closed-domain and open-domain. The closed-domain QA task only requires domain-specific knowledge and is hence easier to solve in terms of collecting the datasets. It is also more accurate than the open-domain QA task [21].
  Input-output variants can also differentiate the QA tasks. There are three variants: extractive QA, open generative QA and closed generative QA. In extractive QA, the model extracts the answer directly from a provided context (text, table, HTML). For open generative QA, the answer is **generated** based on the context. Furthermore, for closed generative QA, the answer is generated without provided context [10].
  In 2.3 an example of such a model is presented. The output shows the answer according to the selected model with a probability score [11].

  Use cases:

  - Frequently asked questions - QA models can be used to automate the response to frequently asked questions by using a knowledge base of domain-specific documents as context [10].
  - Chatbots and virtual agents - a significant part of the conversation between humans and chatbots and virtual agents are questions. The QA models are used to obtain the correct answer from entries that are textual or originally voice entries [28].

- **Sentence similarity** - "sentence similarity is the task of determining how similar two texts are" [12]. Even though the task is challenging due to the ambiguity and variability of linguistic expressions, modelling of sentence similarity has gained marginal attention in recent years because it lies at the core of many NLP applications.
  In recent years, with the success of word embeddings, the modelling switched towards sentence embeddings which convert input sentences into vectors while targeting to preserve the syntactic and semantic information [35]. The sentence similarity task is beneficial for information retrieval and clustering/grouping [6]. There are two variants of sentence



Question Answering demo

using deepset/roberta-base-squad2

How many faculties does CTU have?

Context

Czech Technical University in Prague (CTU, Czech: České vysoké učení technické v Praze, ČVUT) is one of the largest universities in the Czech Republic with 8 faculties, and is one of the oldest institutes of technology in Central Europe. It is also the oldest non-military technical university in Europe.

In the academic year 2020/21, Czech Technical University offered 130 degree programs in Czech and 84 in English. It was considered one of the top 10 universities in emerging Europe and Central Asia in the same year.

Computation time on cpu: 0.144 s

8 0.842

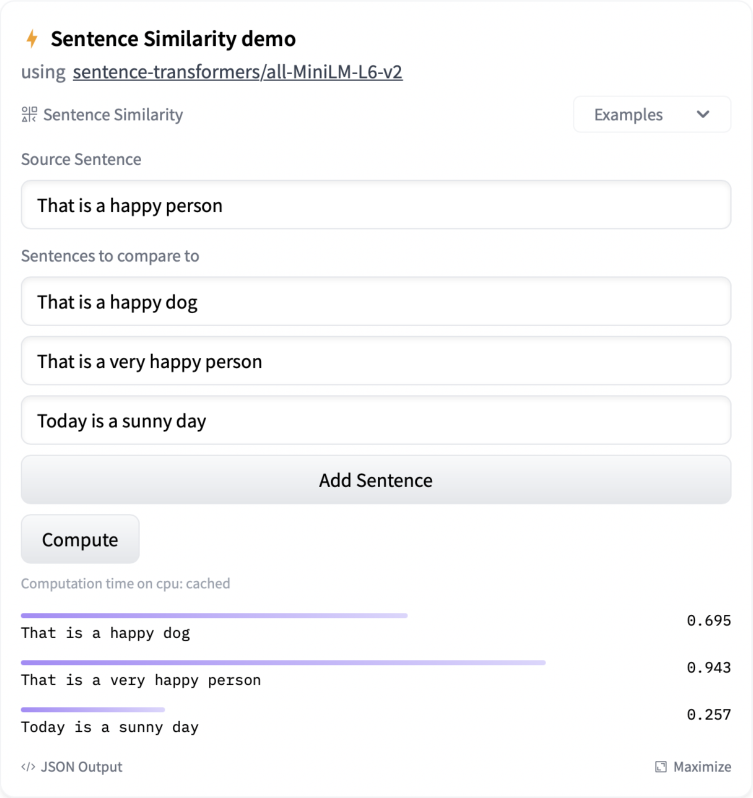
Figure 2.3: QA model performance example.

similarity task: passage ranking and semantic textual similarity.
2.4 shows example output of selected model - all-MiniLM-L6-v2 [6]. Each example sentence of the output carries information about the similarity with the source sentence. This value is a cosine similarity, a simple similarity measure between two vectors.

Use cases:

  - Information Retrieval - You can extract information from documents using Sentence Similarity models. The first step is to rank documents using Passage Ranking models. You can then get to the top-ranked document and search it with Sentence Similarity models by selecting the sentence that has the most similarity to the input query.

- **Summarization** - the task of summarization aims to compress a document to a short, fluent and human-readable form while preserving its important information. Some models can extract text from the original input, while other models can generate entirely new text [34, 13]. There are three main approaches for automatic text summarization: **abstractive**, extractive and hybrid. In extractive methods, the models directly copy salient parts of the input document; in abstractive methods, the models aim to paraphrase the most important parts, and the hybrid methods partially combine the functionality of extractive, and abstractive methods [18].



Sentence Similarity demo

using sentence-transformers/all-MiniLM-L6-v2

Sentence Similarity

Source Sentence: That is a happy person

Sentences to compare to:

That is a happy dog

That is a very happy person

Today is a sunny day

Computation time on cpu: cached

```
That is a happy dog                0.695
That is a very happy person        0.943
Today is a sunny day               0.257
```

Figure 2.4: Sentence similarity model performance example.

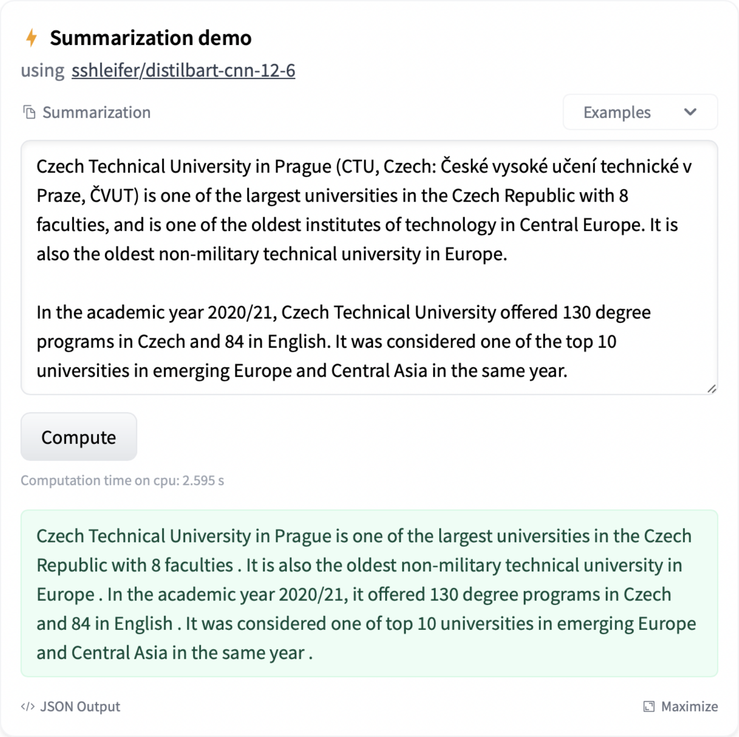
In 2.5 an example of extractive model[1] output is presented. The summarization task is the main focus of this thesis, and it is later discussed in more detail in section 2.2.

Use cases:

- Research Paper Summarization - Research papers can be summarized to allow researchers to spend less time selecting which articles to read. There are several approaches you can take for a task like this: Use an existing extractive summarization model on the Hub to do inference. Pick an existing language model trained for academic papers. This model can then be trained in a process called finetuning so it can solve the summarization task. Use a sequence-to-sequence model like T5 for abstractive text summarization.
- News Summarization - in the informational era, the necessity to quickly asses incoming news is quickly gaining greater and bigger importance. One of the use cases of text summarization is news summarizing task. As per other fields, the flow of information is becoming unbearable for human beings to process it real-time. Automatic summarization of incoming news helps asses and quickly

[1] `https://huggingface.co/sshleifer/distilbart-cnn-12-6`

distinguish the never-ending income of news data.



Summarization demo

using sshleifer/distilbart-cnn-12-6

Czech Technical University in Prague (CTU, Czech: České vysoké učení technické v Praze, ČVUT) is one of the largest universities in the Czech Republic with 8 faculties, and is one of the oldest institutes of technology in Central Europe. It is also the oldest non-military technical university in Europe.

In the academic year 2020/21, Czech Technical University offered 130 degree programs in Czech and 84 in English. It was considered one of the top 10 universities in emerging Europe and Central Asia in the same year.

Computation time on cpu: 2.595 s

Czech Technical University in Prague is one of the largest universities in the Czech Republic with 8 faculties . It is also the oldest non-military technical university in Europe . In the academic year 2020/21, it offered 130 degree programs in Czech and 84 in English . It was considered one of top 10 universities in emerging Europe and Central Asia in the same year .

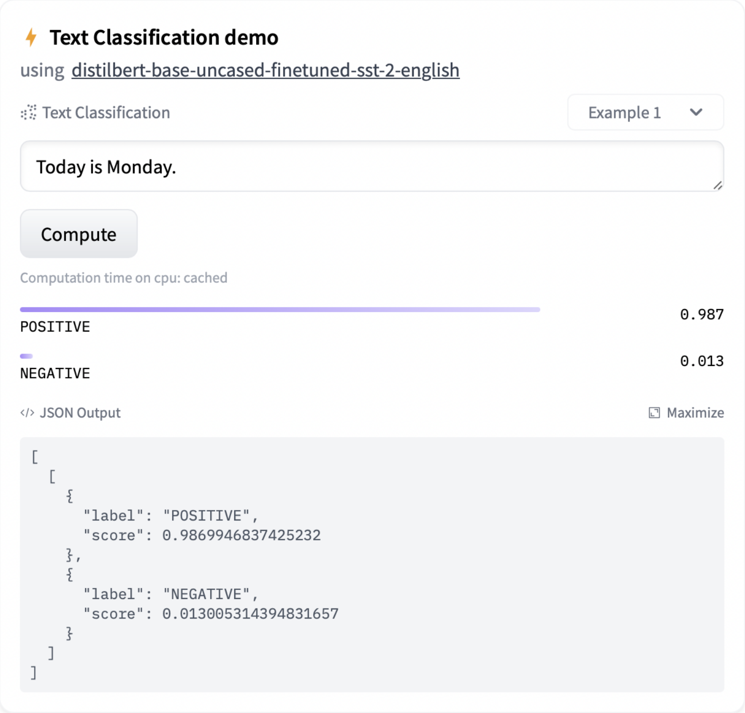
Figure 2.5: Summarization model performance example.

- **Text classification** - “text classification is the task of assigning a label or class to a given text” [14]. It is estimated that over 80% of the world’s information is unstructured, giving the task of text classification huge importance [29]. NLP offers scalability, real-time analysis and consistent criteria, to name a few that would not be possible with the volume of increasing data. There are many text classification applications such as topic labelling, NLI, language detection, grammatical correctness etc. Some of them are listed and introduced below. Moreover, in 2.6, an example of text classification model output is presented. The model analyses input text and labels it either positive or negative with some score [7].

  Use cases:

  - **NLI** - is the task of determining a relationship between two input texts. Specifically, one of the inputs is considered as a premise and the second as a hypothesis. The model outputs one of the three classes: entailment, contradiction and neutral. “The benchmark dataset for this task is GLUE (General Language Understanding Evaluation)” [14]. The usage of NLI is one of the main topics of this thesis, and it is later discussed in more detail in section 2.3.

- Sentiment Analysis - in the sentiment analysis task, the classes can be very diverse. The target of sentiment analysis can be subjected to polarities like positive, negative, neutral, or sentiments such as happiness or anger.
- Topic labelling - a task of understanding and labelling the content of given input text. A specific example is labelling customer service reports into predefined categories to simplify the process.



Text Classification demo

using distilbert-base-uncased-finetuned-sst-2-english

Text Classification

Example 1

Today is Monday.

Computation time on cpu: cached

POSITIVE 0.987

NEGATIVE 0.013

```
[
  [
    {
      "label": "POSITIVE",
      "score": 0.9869946837425232
    },
    {
      "label": "NEGATIVE",
      "score": 0.013005314394831657
    }
  ]
]
```

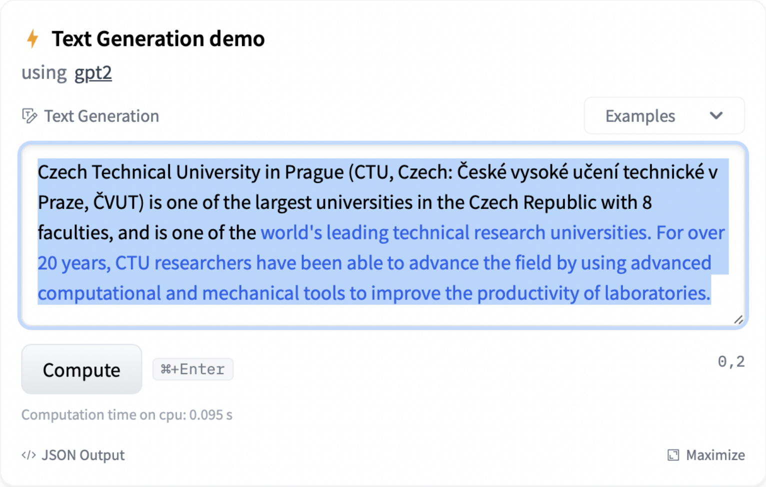
Figure 2.6: Text classification model performance example.

- **Text generation** - is the task of producing new text. This task is included in other previously mentioned tasks, for example, in ATS, where the model paraphrases the origin input. Two main variants of this task are completion generation and text-to-text generation. An example of completion generation is sentence completion. The text-to-text variant of text generation is used, for instance, in the previously mentioned ATS. More details about text-to-text models like BART provided in section 2.2. In 2.6 an example of text generation model, concretely completion generation model, gpt2 is presented [9, 15].

  Use cases:

  - Code Generation - is using so-called causal language models trained on particular code to help the programmers in their repetitive coding tasks.

- Stories Generation - a story generation model creates a story-like sequence based on the input.



Czech Technical University in Prague (CTU, Czech: České vysoké učení technické v Praze, ČVUT) is one of the largest universities in the Czech Republic with 8 faculties, and is one of the world's leading technical research universities. For over 20 years, CTU researchers have been able to advance the field by using advanced computational and mechanical tools to improve the productivity of laboratories.

Computation time on cpu: 0.095 s

Figure 2.7: Text generation model performance example.

- **Token classification** - the goal of this task is to detect and label various entities from an input text. These entities might be person, location, organization, dates, places, or even parts of speech [16].
  In 2.8 a example of token classification model[2] output is presented, concretely named entity recognition (NER).

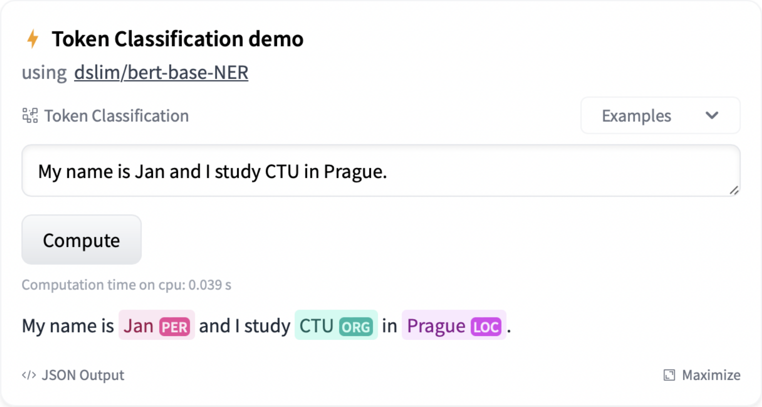
  Use cases:

  - Part-of-speech (POS) tagging - is a variant of token classification where the model aims to label each word of an input with part of speech [28, 16].
  - NER - another variant of token classification where the goal is to tag each word with a predefined named entity.

- **Translation** - Translation is the task of converting text from one language to another.
  Use cases:

  - Multilingual conversational agents - Translation models, can be used to build conversational agents across different languages. This can be done in two ways. Translate the dataset to a new language. You can translate a dataset of intents (inputs) and responses to the target language. You can then train a new intent classification model with this new dataset. This allows you to proofread responses in the target language and have better control of the chatbot's outputs. Translate the input and output of the agent. You can use

[2] https://huggingface.co/dslim/bert-base-NER



Token Classification demo

using dslim/bert-base-NER

Token Classification

Examples

My name is Jan and I study CTU in Prague.

Compute

Computation time on cpu: 0.039 s

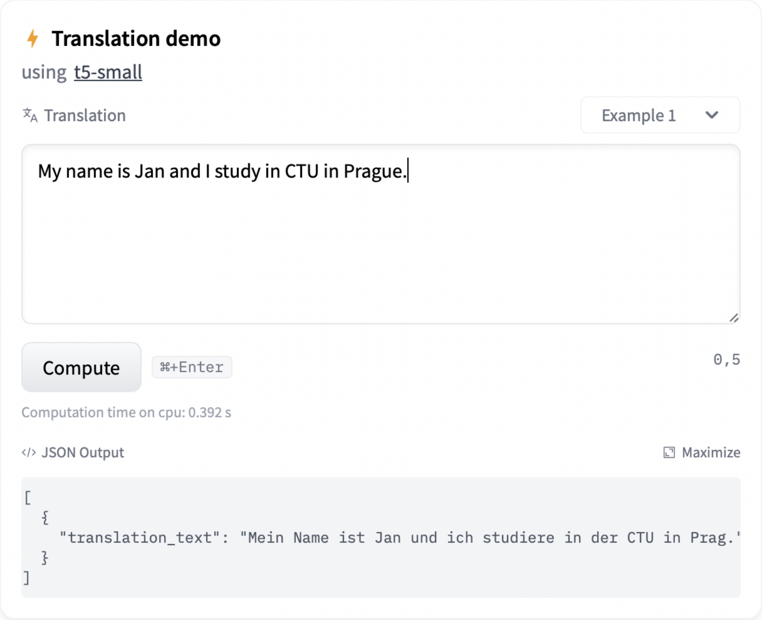
My name is Jan PER and I study CTU ORG in Prague LOC .

</> JSON Output

Maximize

Figure 2.8: Token classification model performance example.

a Translation model in user inputs so that the chatbot can process them. You can then translate the output of the chatbot into the language of the user. This approach might be less reliable as the chatbot will generate responses that were not defined before.



Translation demo

using t5-small

Translation

Example 1

My name is Jan and I study in CTU in Prague.

Compute

⌘+Enter

0,5

Computation time on cpu: 0.392 s

</> JSON Output

Maximize

```
[
  {
    "translation_text": "Mein Name ist Jan und ich studiere in der CTU in Prag.'
  }
]
```

Figure 2.9: Translation model performance example.

Audio tasks

- Text-to-speech - also called speech-to-text, is the task of reliably converting voice data into text data. Speech recognition is required for any application that follows voice commands or answers spoken questions.

What makes speech recognition especially challenging is the way people talk—quickly, slurring words together, with varying emphasis and intonation, in different accents, and often using incorrect grammar.

- **Automated speech recognition (ASR)** - NLP techniques are actually designed for text but can also be applied to spoken input. ASR transcribes oral data into a stream of words. Neural networks and hidden Markov models are used to reduce speech recognition's error rate, however, it's still far from perfect. The main challenge is the lack of segmentation in oral documents. And while human listeners can easily segment spoken input, the automatic speech recognizer provides unannotated output.
- **Dialogue systems** - perhaps the omnipresent application of the future, in the systems envisioned by large providers of end-user applications. Dialogue systems, which usually focus on a narrowly defined application (e.g. your refrigerator or home sound system), currently utilize the phonetic and lexical levels of language. It is believed that utilization of all the levels of language processing explained above offer the potential for truly habitable dialogue systems.

2.1.2 Preprocessing

- **Tokenization** - is an essential task in natural language processing used to break up a string of words into semantically useful units called tokens. Sentence tokenization splits sentences within a text, and word tokenization splits words within a sentence. Generally, word tokens are separated by blank spaces and sentence tokens by stops. However, you can perform high-level tokenization for more complex structures, like words that often go together, otherwise known as collocations (e.g., New York).
- **Part of speech tagging** - also called grammatical tagging, is the process of determining the part of speech of a particular word or piece of text based on its use and context.
- **Lemmatization and Stemming** - When we speak or write, we tend to use inflected forms of a word (words in their different grammatical forms). To make these words easier for computers to understand, NLP uses lemmatization and stemming to transform them back to their root form. The word as it appears in the dictionary – its root form – is called a lemma. When we refer to stemming, the root form of a word is called a stem. Stemming "trims" words, so word stems may not always be semantically correct. While lemmatization is dictionary-based and chooses the appropriate lemma based on context, stemming operates on single words without considering the context. Even though stemmers can lead to less-accurate results, they are easier to build and perform

faster than lemmatizers. But lemmatizers are recommended if you're seeking more precise linguistic rules.

- **Stopword Removal** - removing stop words is an essential step in NLP text processing. It involves filtering out high-frequency words that add little or no semantic value to a sentence, for example, which, to, at, for, is, etc. You can even customize lists of stopwords to include words that you want to ignore. Let's say you want to classify customer service tickets based on their topics. In this example: "Hello, I'm having trouble logging in with my new password", it may be useful to remove stop words like "hello", "I", "am", "with", "my", so you're left with the words that help you understand the topic of the ticket: "trouble", "logging in", "new", "password".

2.2 Summarization

As mentioned in 2.1.1, the task of summarization aims to compress input text while preserving the most salient information. There are three variants of summarization techniques. First is the *extractive* method, where the summary is created by directly picking the parts of the original text. Second technique is the **abstractive** method. This technique is essentially the task of text generation. The model paraphrases parts of the input text to create the summary. The third variant is *hybrid* which is a combination of the first two variants.

As Gupta et al. state, *extractive summarizazion* has been a very extensively researched topic a has reached its maturity stage mainly due to its simplicity. Moreover, the abstractive approach helps to solve the dangling anaphora problem[3]. Therefore, generally creates more readable, concise and cohesive summaries and avoids information redundancy [5]. Thus, the research focus has shifted mainly towards **abstractive summarization**. ATS is the summarization technique this thesis concerns.

From a perspective of the number of documents considered as an input to the summarization process, Gupta et al. also propose differentiating *single-document* and *multi-document* summarization.

Moreover, there are different variants on the basis of how much information is to be summarized. The *generic* summarization, where the summary of the whole text is obtained, and the *query-focused summarization*, where only the summary according to the context which is specified by user is obtained [5].

Final distinguishing considers *indicative summarization*, where focus is on

[3] `https://en.wikipedia.org/wiki/Anaphora_(linguistics)`

telling what the text is about and *informative summarization*, “where the main content of the text is extracted by analyzing the original text” [5].

In this section, we present an overview of the state-of-art of summarization task from various perspectives. Firstly, we present the common tasks of summarization considering the dataset structures. Secondly, the overview of the approaches to ATS is presented, and the field-dominant Transformers architecture is briefly explained. Thirdly, we list some popular datasets both for English and, most importantly, the Czech language. Lastly, the most notable models and evaluation methods are discussed with a focus on model-based methods.

2.2.1 Tasks

Typically the summarization datasets have three common things in their structures. Each dataset consists of a full text, its headline and a summary. Based on this premise, Straka et al. provide a well-arranged list of different summarization tasks to differentiate [41]:

- **Full text to abstract (T2A)** - generate multi-sentence abstract from the original text.
- **Full text to headline (T2H)** - generate a one-sentence headline from the original text.
- Abstract to headline (A2H) - generate a one-sentence headline from the abstract.

In the practical part, we evaluate our NLI models over the highlighted tasks T2A and T2H.

2.2.2 State-of-art techniques

Gupta et al. present an extensive summary of present techniques for abstractive summarization [5]. “Structure-based approaches are those where the vital information of the text is populated into the predefined structure to create the abstractive summaries. Structure-based approaches are divided into tree-based, template- based, ontology-based, lead-and-body phrase, graph-based and rule-based methods according to the structure used for creating summaries.

Whereas Semantic-based approaches are those which take the text document as input, create the semantic representation of text and then feed this representation to the Natural Language Generation system to create the final abstractive summary. They are divided into information-item-based, predicate-argument based, semantic-graph based and multimodal" [5]. Figure

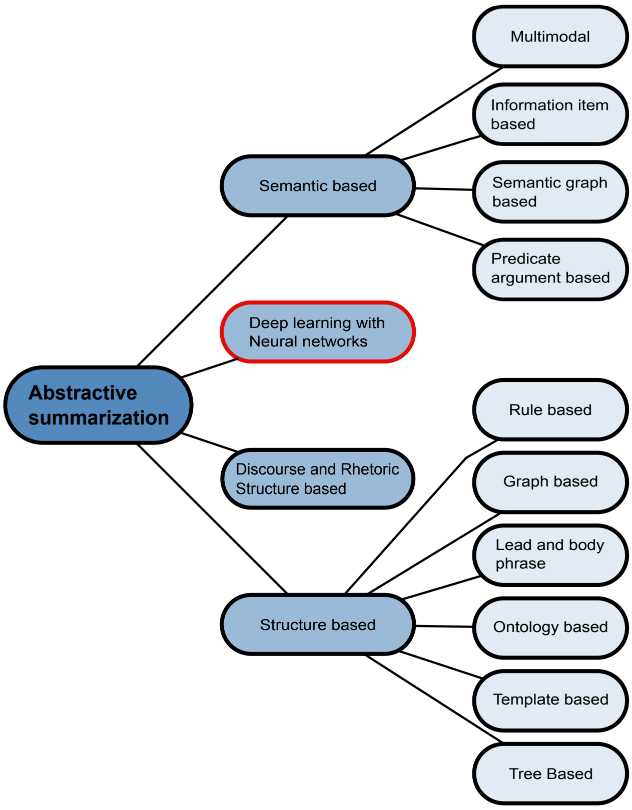


Figure 2.10: Technique overview of abstractive summarization methods.

2.10 presents an overview of current ATS techniques based on broad research of Gupta et al. [5]. The highlighted *deep learning with neural networks* is considered the state-of-art approach. Concretely, Recurrent Neural Network (RNN) has achieved eminent performance for NLP tasks [5, 42].

Transformer architecture

In 2017 Vaswani et al. introduced an innovative **Transformer architecture** which is nowadays considered as the state-of-art technique [42, 43]. Their approach combines two already utilized methods - recurrent neural networks and attention mechanisms. "Recurrent models typically factor computation along the symbol positions of the input and output sequences" [42]. This inherently sequential nature prevents the possibility of parallelization within training examples, a critical issue considering computational efficiency. In contrast, the attention mechanism allows to model dependencies without regard to their distance in the input or output sequences [39]. Vaswani et al.

utilized combining RNN and attention mechanism to create a new approach which replaced the procedurally time-consuming recurring part.

The model architecture is depicted in figure 2.10. The figure consists of two essential parts of the transformer architecture - the **encoder** and the **decoder**.

Encoder: "The encoder is composed of a stack of N = 6 identical layers. Each layer has two sub-layers. The first is a multi-head self-attention mechanism, and the second is a simple, position-wise, fully connected feed-forward network" [42].

Decoder: "The decoder is also composed of a stack of N = 6 identical layers. In addition to the two sub-layers in each encoder layer, the decoder inserts a third sub-layer, which performs multi-head attention over the output of the encoder stack" [42].

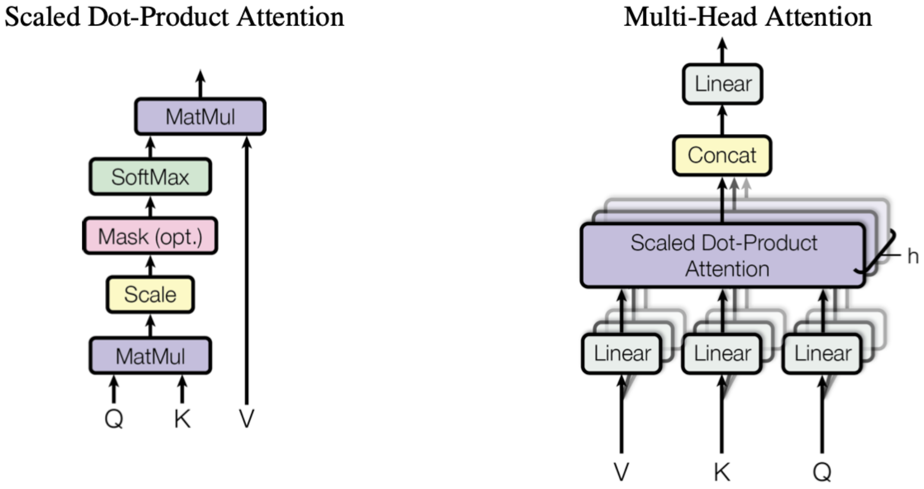


Figure 2.11: (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel. Copied from [42].

The last essential part of the architecture is the **attention** mechanism. The attention can be described as a mapping between a query and a set of key-value pairs to an output [42]. All of the information is represented as a vector. "The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key" [42].

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{2.1}$$

Equation 2.1 depicts how the scaled dot-product attention is calculated. "The input consists of queries and keys of dimension d_k , and values of dimension

d_v" [42]. The dot products of the query with all the keys is calculated. Then, we divide each by dk, and a softmax function is applied to obtain the weights on the values. In practice, the attention function is computed on a set of queries, values and keys simultaneously, chained together into a matrices Q, V and K.

Equation 2.2, depicts the way the MultiHead attention is calculated. "Multi-head attention allows the model to jointly attend to information from different representation subspaces at different positions. With a single attention head, averaging inhibits this" [42].

$$MultiHead(Q, K, V) = Concat(head1, ..., headh)W^O \tag{2.2}$$

Where the projections are parameter matrices $W_i^Q \in R^{d_{model}\times d_k}$,$W_i^K \in R^{d_{model}\times d_k}$, $W_i^V \in R^{d_{model}\times d_v}$ and $W^O \in R^{hd_{model}\times d_v}$. In total, eight parallel attention layers or heads are deployed. For each of these $dk = dv = d_{model}/h = 64$ is used. "Due to the reduced dimension of each head, the total computational cost is similar to that of single-head attention with full dimensionality" [42].

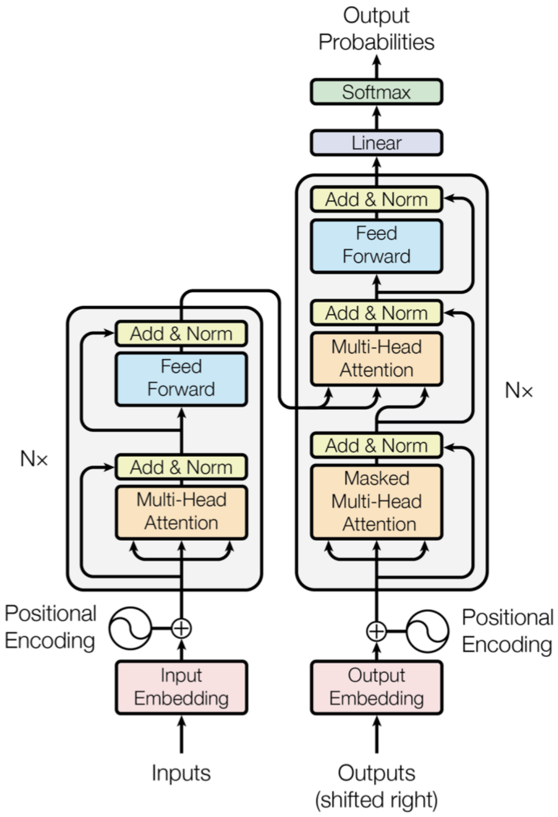


Figure 2.12: The Transformer - model architecture. Copied from [42]

2.2.3 Datasets

The necessity of large-scale supervised datasets has risen by shifting towards the Transformers approach that utilizes the RNNs. The common approach for solving the summarization task is to finetune pre-trained transformer-based models on a dataset [18, 1].

In this section, two of the most-used summarization datasets are presented for informative purposes. More importantly, an overview of present datasets for **Czech language** is covered in more detail. Namely, the benchmark SumeCzech dataset and the closed Czech News Center (CNC) dataset are covered, including some general statistics of the datasets and example entries. These two datasets were later used in this thesis which is discussed in chapter 4.

CNN/Daily mail

The CNN/Daily[4] mail dataset is perceived as the standard dataset for the English text summarization [4]. Originally a dataset for QA was firstly used for summarization by Nallapati et al. [31]. The dataset contains over three hundred thousand articles with short highlights. The dataset split is presented in 2.1 and an example entry from the dataset is presented in 2.1.

Listing 2.1: Example entry from the CNN/Daily mail dataset.

```
{
  "id": "0054d6d30dbcad772e20b22771153a2a9cbeaf62",
  "article": "(CNN) -- An American woman died aboard
      a cruise ship that docked at Rio de Janeiro on
      Tuesday, the same ship on which 86 passengers
     previously fell ill, according to the state-run
      Brazilian news agency, Agencia Brasil. The
     American tourist died aboard the MS Veendam,
     owned by cruise operator Holland America.
     Federal Police told Agencia Brasil that
     forensic doctors were investigating her death.
     The ship's doctors told police that the woman
     was elderly and suffered from diabetes and
     hypertension, according the agency. The other
     passengers came down with diarrhea prior to her
      death during an earlier part of the trip, the
     ship's doctors said. The Veendam left New York
     36 days ago for a South America tour.",
```

[4] https://huggingface.co/datasets/cnn_dailymail

```
  "highlights": "The elderly woman suffered from
     diabetes and hypertension, ship's doctors say
     .\nPreviously, 86 passengers had fallen ill on
     the ship, Agencia Brasil says."
}
```

Split	Number of entries
Train	287 113
Validation	13 368
Test	11 490

Table 2.1: Structure of the CNN/Daily mail dataset.

WikiLingua

In 2020, Ladhak et al. introduced a new benchmark dataset named WikiLingua with articles/summary pair extracted from open-source WikiHow[5]. The added value of this dataset is that it fills the gap of multilingual datasets for ATS [19] which is key to multilingual and cross-lingual summarization. The dataset contains over seven hundred thousand entries from eighteen different languages including seven thousand entries of **Czech language** and it is the largest known multilingual dataset. Each of the article/summary pair is reviewed by average by twenty people to ensure the quality [19].

SumeCzech

In 2018, Straka et al. created a benchmark open-source Czech dataset for text summarization [41]. The dataset consists of over one million articles collected from five Czech websites.

In 2.2 and example entry picked from SumeCzech dataset is presented.The included information at each entry of the dataset follows a standardized structure for text summarization datasets - each entry consisting of a headline, a several sentence long abstract and a full text.

Listing 2.2: Example entry from the SumeCzech dataset.

```
{
   "abstract":"Kdo hledá do interiéru tak trochu
      jinou podlahu, určitě dříve či později objeví
      přírodní marmoleum, nové dekory měkčeného PVC
      nebo sametový vinyl s názvem Flotex, který
```

[5] https://www.wikihow.com/Main-Page

```
      nejvíce připomíná nakrátko střižený koberec.
      Jsou to všechno materiály nejen krásné, ale
      hlavně praktické a cenově přijatelné.",
   "dataset":"test",
   "filename":"crawl-data/CC-MAIN-2017-51/segments
      /1512948567042.50/warc/CC-MAIN
      -20171215060102-20171215080102-00442.warc.gz",
   "headline":"Trendy podlahy vyzývají ke kreativitě
       i návratu k přírodě",
   "length":17015,
   "md5":"bb6acc62f3901611d20de3bc04997f2d",
   "offset":798320668,
   "published":"2011-03-01T06:45:00+0100",
   "section":"bydleni",
   "subdomain":"novinky.cz",
   "text":"Volba podlahy do interiéru je zásadní.
      Podlaha zaujímá velkou plochu a ovlivňuje
      vzhled a styl místnosti. Je mnohem důležitější
       než odstín sedačky nebo kuchyňské linky.
      Teprve podle podlahy vybíráme a ladíme ostatní
       kusy nábytku, materiály, odstíny ...",
   "url":"https://www.novinky.cz/bydleni/tipy-a-
      trendy/225436-trendy-podlahy-vyzyvaji-ke-
      kreativite-i-navratu-k-prirode.html"
}
```

The dataset was collected using Common crawler API[6] from five news websites. The breakdown of each part of the dataset is presented in figure 2.2 and the dataset split structure is presented in figure 2.3. Notably, the authors of the SumeCzech dataset provide even an out-of-domain test split. This could be convenient for emulating the standard challenge of training and testing models on different datasets. However, for our case, we have taken advantage of having access to the private CNC dataset, which is described in the next section.

Website	Number of entries
ceskenoviny.cz	4854
denik.cz	157 581
idnes.cz	463 192
lidovky.cz	136 899
novinky.cz	239 067
Total	1 001 593

Table 2.2: Breakdown of the sources for SumeCzech dataset. Recreated from [41].

[6] http://commoncrawl.org

Split	Number of entries
Train	867 596
Dev	44 567
Test	44 454
Out-of-domain test	44976

Table 2.3: Structure of the SumeCzech mail dataset. Recreated from [41].

CNC

The private CNC dataset was supplied by thesis supervisor Jan Drchal. It contains approximately eight hundred thousand entries of Czech articles. Each entry is a standard JSON Lines[7] format and contains the original article text, a few-sentence abstract and a headline.

CNC-Sum

The CNC-Sum dataset is a recently created dataset by my colleague Martin Krotil from the CTU university[26]. This private dataset was created by concatenating the SumeCzech and the CNC dataset. It is worth mentioning because it is arguably the largest Czech dataset created to this day containing over 1.7 million Czech news articles and corresponding abstracts and headlines.

2.2.4 Models

In this section, we briefly introduce some of the benchmark models in the field of text summarization. The biggest emphasis here is put on **mBart** which is later used as the model for abstractive text summarization part of the NLI-fact-check pipeline discussed in section 4.3.

BERT

The Bidirectional Encoder Representations from Transformers (BERT) is one of the most used known and used models for text summarization [2]. Similarly, like CNN/Daily mail for datasets and ROUGE for metrics, BERT is considered as a benchmark in its context [34]. BERT was introduced in 2019 by Devlin et al. and was one of the first encoder-based pre-trained models derived from the Transformer architecture [2]. Moreover, it is very

[7] https://jsonlines.org

much worth mentioning since most of the current models are somehow a derivation of BERT[34].

BART

BART is another benchmark model. It is a “a denoising autoencoder for pretraining sequence-to-sequence models. BART is trained by (1) corrupting text with an arbitrary noising function and (2) learning a model to reconstruct the original text. It uses a standard Transformer-based neural machine translation architecture which, despite its simplicity, can be seen as generalizing BERT (due to the bidirectional encoder)” [22].

mBART

In 2020, the Facebook research team proposed a first multilingual sequence-to-sequence denoising autoencoder. “mBART is trained by applying the BART [22] to large-scale monolingual corpora across many languages. The input texts are noised by masking phrases and permuting sentences, and a single Transformer [42] model is learned to recover the texts.

Different from other pre-training approaches for machine translation, mBART pre-trains a complete auto-regressive Seq2Seq model. mBART is trained once for all languages, providing a set of parameters that can be finetuned for any of the language pairs in both supervised and unsupervised settings, without any task-specific or language-specific modifications or initialization schemes” [25].

This model is used in our work in the NLI-fact-checking pipeline for the summarization part. It was recommended by the thesis supervisor as the best-performing model.

2.2.5 Evaluation methods

In this section, we distinguish between three methods of text summarization evaluation. They are, *human annotations*, *similarity based* metrics and **model based metrics**.

Human annotations

Human annotation is still the predominant metric for evaluating generated summaries which is actually one of the major drawbacks in the field. Despite significant progress in the task of text summarizazion over the last decade, the insufficient number of automatic evaluation protocols is one of the aspects that is significantly slowing down the improvements in the quality of summarizations [18].

Similarity based

- **ROUGE** - or Recall-Oriented Understudy for Gisting Evaluation is the most used metric for assessing quality of a summarization [18, 24]. It was firstly introduced by Chin-Yew Lin in 2004 [24]. Even though being almost twenty years old, the protocol it is still dominating the field and is considered a default automatic evaluation metric [4]. In the time when this paper was introduced, the only evaluation were human. The main idea was to create first automatic evaluation which would save significant time.
  The rouge metric is usually calculated compactly in three different scores:
  1. Recall - is a measure that can be defined as follows:
     $$r = \frac{N_{overlaps}}{N_{words_{golden}}} \tag{2.3}$$
     where $N_{overlaps}$ is the number of overlapping words between generated text, usually noted as *system* and original text commonly noted as *gold/golden*.
  2. Precision - calculates the similarity between the generated text and the overlapping parts of the two texts:
     $$p = \frac{N_{overlaps}}{N_{words_{system}}} \tag{2.4}$$
  3. F1-score - or the f-score is a harmonic mean of the two previous measures:
     $$F1 = \frac{2rp}{r+p} \tag{2.5}$$

  To put it into words, *recall* demonstrates how is the generated text similar to the original word-by-word. Precision shows how much extra information on top of the original text is in the generated one. And the f-score combines those two measures.
- **ROUGE-WE** - “extends ROUGE by using soft lexical matching based on the cosine similarity of Word2Vec embeddings” [18, 32].

- **METEOR** - “computes an alignment between candidate and reference sentences by mapping unigrams in the generated summary to 0 or 1 unigrams in the reference, based on stemming, synonyms, and paraphrastic matches. Precision and recall are computed and reported as a harmonic mean” [4, 20].

- **ROUGE-CS** - is a custom-created variant of ROUGE created by a fellow colleague from my university Šimon Zvára for his bachelor thesis [40]. The metric aims to better capture the similarities since the original metric ROUGE is trained mainly on English datasets it does not perform well on varied languages such as Czech. The score is very sensitive to paraphrasing, noise injection and other transformations that do not have to necessarily change the meaning and similarity with the original claim [40].

Model based

- S^3 - In 2017, Peyrard et al. introduced one of the first model-based approaches. Their approach was to learn a new metric that was trying to maximize the correlation over standard similarity metrics like ROUGE and hefty humanly-annotated datasets [33].

- **FactCC** - Another model-based approach out of a few available is FactCC. An approach proposed by Kryścinński et al. Their approach involved training a BERT-like model on a custom-created dataset after applying rule-based transformations to an original text. These transformations included entity swaps, noise injections, sentence negations and others [18].

- **Memes-CS** - As a part of his thesis, Šimon Zvára also introduced arguably one of the first model-based approach for evaluating Czech summarizations. His approach was adopted from the work of Kryścinński et al. on FactCC. He trained the model on a similarly created dataset utilizing several text transformations [40].

2.3 Natural language inference

The Natural Language Interface (NLI) is one of the most important disciplines from the set of NLP tasks. The task of NLI, also known as Recognizing Textual Entailment, is to determine the inference relation between two short ordered texts or statements (premise and hypothesis). This relation is then determined into three categories:

- the hypothesis is an entailment of the premise
- the hypothesis is a contradiction of the premise
- the hypothesis is neutral to the premise

This discipline is widely used in many applications where the input obtained from the human/end-user need to follow certain hypothesis (banking, retail, finance etc.).

Chapter 3

Challenges for Czech summarization

In this chapter, we briefly go through the motivation and challenges behind the Czech summarization.

3.1 Focusing on the Czech language

In general, the recent survey has shown that at least 30% of the generated summaries are unusable [18].

In the environment of the Czech language, which is a much less spread language than, for example, English, many more challenges arise when it comes to automatic summarization tasks. Therefore in the field where pre-trained models dominate, we lack large corpus datasets that would greatly help development.

The second biggest challenge is the great variety of the Czech language. Compared to English, the Czech language is much more variate and rich in expressions. Its structure in its definition is much more diverse, and to capture and model it is a great challenge that again depends on suitable datasets.

Lastly, there are challenges in terms of evaluation metrics. As Kryściński et al. [18] state, the field of ATS still lacks robust and suitable automatic metrics that assess the factual consistency and overall quality of the generated summarization. There are some generalized metrics like ROUGE, but these metrics are showing to be almost useless when dealing with different languages. What Zvára has correctly pointed out and picked is that ROUGE in particular is very sensitive towards small changes in the sentence structure that do not necessarily changed the meaning [40]. That presents a big challenge when dealing with a highly variate language like the Czech. Therefore, one of the goals of this thesis is to examine the behaviour of the NLI model-based metric for assessing the facticity of generated summaries.

Chapter 4

Practical part

In this chapter, we go through the practical part of this thesis. Firstly, the fundamental information about the implementation is presented such as the used hardware and software. In the second part of this chapter, we go through the process of creating custom annotated dataset. Finally, in the last part of this chapter the implementation of the NLI fact-checking pipeline is discussed and described.

4.1 Fundamentals

This section covers the fundamentals used for the implementation of the presented work - software and hardware. For the hardware part we discuss the utilized infrastructure of super computers need for training and evaluating of the used models. In the software section all the utilized software during the implementation is listed and briefly discussed.

4.1.1 Hardware

All presented work was done utilizing an infrastructure of Research Center for Informatics (RCI)[2]. The RCI hosts a cluster for high performance computing that is available to researchers at the Czech technical university in Prague. There are two sets of nodes for computation - Intel and AMD nodes. Tables 4.1 and 4.2 show detailed specifications of these nodes. The whole interconnection diagram of the cluster is presented in 4.1.

[1] Copied from `http://www.rci.cvut.cz`

[2] `http://www.rci.cvut.cz`

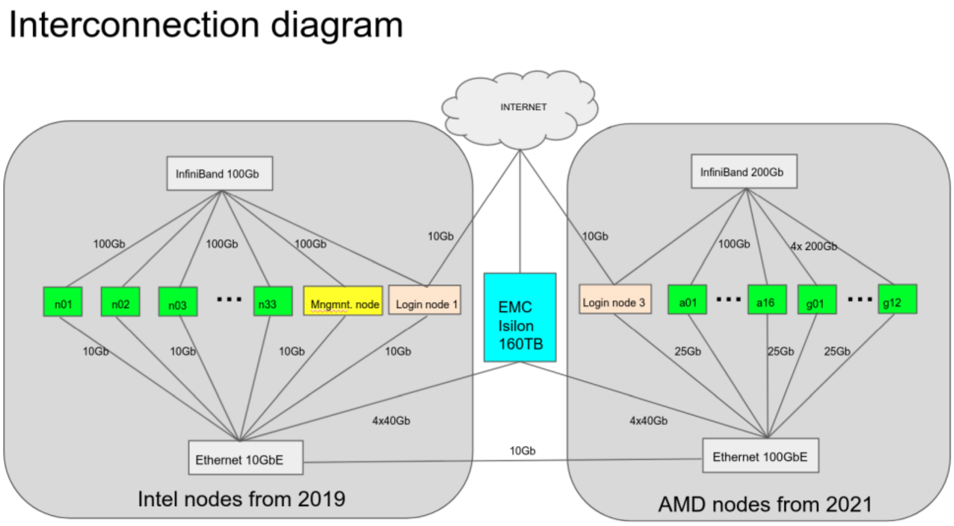


Figure 4.1: Interconnection diagram of RCI cluster[1].

Id	Specification
n_{01-20} CPU nodes	24 cores/48 threads 3.2GHz (2 x Intel Xeon Scalable Gold 6146), 384GB RAM
n_{21}–n_{32} GPU nodes	36 cores/72 threads 2.7GHz (2 x Intel Xeon Scalable Gold 6150), 384GB RAM, 4 x Tesla V100 with NVLink
n_{33} multi-CPU node	192 cores/ 384 threads 2.1GHz, 1536GB RAM

Table 4.1: Description of Intel nodes of the RCI cluster.

4.1.2 Software

This section briefly covers the used software during the implementation process. The RCI cluster comes up with preinstalled utils to simplify workspace setup and auxiliary modules. Namely, Lmod[3] is available as a main util for creating custom workspaces.

Python

For all the work presented we used open source machine learning framework pytorch[4], specifically *1.7.1-fosscuda-2019b-Python-3.7.4*.

[3] `https://lmod.readthedocs.io/en/latest/`

[4] `https://pytorch.org`

Id	Specification
a_{01-16} CPU nodes	64 cores/128 threads 3.1GHz (2 x AMD EPYC 7543), 1TB RAM
g_{01-g10} 4GPU nodes	64 cores/128 threads 3.1GHz (2 x AMD EPYC 7543), 1TB RAM, 4 x Tesla A100 40GB with NVLink
g_{11-12} 8GPU nodes	128 cores/256 threads 3.1GHz (2 x AMD EPYC 7763), 1TB RAM, 8 x Tesla A100 40GB with NVLink

Table 4.2: Description of AMD nodes of the RCI cluster.

Transformers library

During the phase of the summarization model training and evaluating we also used the open-source library **Transformers** provided by HuggingFace[5]. This library provides broad functionalities to help to manipulate with datasets and models.

Slurm

The RCI cluster uses an open-source software Slurm[6] as a job scheduler. Most of the scripts were run as a Slurm job (sbatch files) on the cluster of Intel nodes.

4.2 Custom dataset generation

One of the critical part of the implementation was to develop a human annotated abstracts and headlines. As mentioned in, 2.2.3 human annotation is still of great value among evaluation metrics. For comparing purposes of we have gather in total 202 annotated entries from CNC and SumeCzech test splits combined.

At first the generated summaries and headlines were copied into private

[5] `https://huggingface.co`
[6] `https://slurm.schedmd.com`

Google sheets file[7]. Each line in the document corresponded to a single article. The annotators were given the original text and a generated abstract and headline and were asked to mark it either **"1" - GOOD**, or **"0" - BAD**. The team of annotators contained in total eight people, myself included. All of them are rightfully acknowledged at the beginning of this thesis.

After the annotation process, a python script to mine the data back was created. For each dataset-task (e.g. sumeczech-headlines) pair a JSON lines file was generated. Example of such file is in listing 4.1.

Listing 4.1: Example entry from mined annotated data.

```
{
   "id":8,
   "gold":"Daleko od bran semifinále zůstal i druhý
      český zástupce prsař Jiří Jedlička.",
   "system":"Krasobruslař Tomáš Verner překonal česk
      ý rekord v počtu vyhraných zápasů na okruhu
      ATP. V dnešním čtvrtfinále na světovém šampion
      átu v čínském Chang-čou porazil o více než
      deset bodů krajana Tomáše Magnuska.",
   "evaluation":"0"
}
```

The distribution of annotated articles is shown in table 4.3.

Dataset	Number of annotations	Split
SumeCzech	102	Test
CNC	100	Test

Table 4.3: Distribution of annotated articles.

4.3 NLI fact-checking pipeline

In this section we describe the implemented fact-checking pipeline for comparing the functionality as evaluators of summarization tasks T2A and T2H mentioned in subsection 2.2.1.

4.3.1 Steps

- **Load summarization model** - the summarization model was provided by our supervisor. For the summarization task we used the *mBart*. The

[7] https://www.google.com/sheets/about/

model was trained on cnc dataset with learning rate set to $2e^{-05}$ and batch size 8.

- **Generate examples** - this pre-trained model was then ran on CNC and SumeCzech datasets. That provided us two sets of results for testing the performance of selected NLI models.
- **NLI models** In total five different pre-trained NLI models were selected for comparison in assessing factual consistency of summaries and headlines.

NLI Model	Train dataset	Validation dataset	Score
bert-base-multilingual			
xlm-roberta-large			
xlm-roberta-large-xnli			
FERNET-C5			
xlm-roberta-fever			

Table 4.4: Description of the used NLI models.

- **Rules for assessing facticity with NLI** As a part of experiments we have introduced two evaluating rules:
  (a) **Mild rule** - Typically the scores of NLI outputs are: *0 - "SUPPORTS", 1 - "REFUTES", 2 - "NOT ENOUGH INFO"*. For our purpose, we have switched the scoring from the output of the model to either *1 - "SUPPORTS", 0.5 - "NOT ENOUGH INFO", 0 - "REFUTES"*. For **mild rule** the headline or abstract is given score as mean of the scores of the sentences (claims) in it. For mild rules there can be claims with "NOT ENOUGH INFO". The abstract/-headline is given *0 - "REFUTES"* if and only if there is at least one claim with this score.
  (b) **Strict rule** - every abstract or headline which has at least one claim evaluated as *0 - "REFUTES"* **or** *0.5 - "NOT ENOUGH INFO"* is automatically evaluated to *0 - "REFUTES"* as a whole.
- **Correlation statistics** - we have selected in total **eight** metrics for which we calculate correlation with the NLI score. We have selected **ROUGE metric**, **ROUGE-CS** (for both metrics we picked unigrams, 2-grams and L versions), the human annotations and finally output of the model-based MEMES-CS. The corresponding results are presented in the next chapter.

Chapter 5

Experiments and results

In this chapter the results of the evaluation fact-checking pipeline are presented. As stated in the previous section, we have created summarizations both on CNC and SumCzech dataset. The chosen model was *mbart25-large* provided by the supervisor as the best competing pre-trained model available. Both tasks were assessed using the **mild rule**.

The correlations in the tables are abbreviated as follows:

- **ann/nliy** - correlation between annotated and nli classes
- **rge-1/nliy/nliy** - correlation between ROUGE-1 and nli classes
- **rge-2/nliy/nliy** - correlation between ROUGE-2 and nli classes
- **rge-l/nliy/nliy** - correlation between ROUGE-L and nli classes
- **rge-1-cs/nliy** - correlation between ROUGE-CS-1 and nli classes
- **rge-1-cs/nliy** - correlation between ROUGE-CS-2 and nli classes
- **rge-1-cs/nliy** - correlation between ROUGE-CS-L and nli classes
- **mms-cs/nliy** - correlation between MEMES-CSand nli classes

5.1 Headlines

Here we presents the results obtained by evaluating the generated headlines from the test splits of **CNC** and **SumeCzech** datasets. Listings 5.1 and

5.2 show one entry of the output of the evaluation pipeline and the collected correlations respectively.

Listing 5.1: Example output of the evaluation pipeline - one entry.

```
{
   "model":"bert_base_model",
   "gold":"Trendy podlahy vyzývají ke kreativitě i n
      ávratu k přírodě",
   "system":"Podlaha, která se hodí do každého
      interiéru. Vybíráme ji pečlivě!",
   "ycls":1,
   "rouge-1:":0.6885245901639344,
   "rouge-2:":0.23333333333333334,
   "rouge-l:":0.3442622950819672,
   "rouge_cs_1":-0.5490909090909091,
   "rouge_cs_2":-0.0018487394957983194,
   "rouge_cs_l":0.409622317454854,
   "annotated":1
}
```

Listing 5.2: Example output of the evaluation pipeline - resulted correlations.

```
{
   "model":"FERNET-C5",
   "ann/nliy":0.28039985489311253,
   "rouge-1/nliy":-0.14974145442445447,
   "rouge-2/nliy":-0.14974145442445447,
   "rouge-l/nliy":-0.17399099567750204,
   "rouge-1-cs/nliy:":0.1331707093279698,
   "rouge-2-cs/nliy:":0.10385153748418738,
   "rouge-l-cs/nliy":0.10385153748418738,
   "memes-cs/nliy ":0.21801313444333113
}
```

5.1.1 SumeCzech

Firstly, we have ran through the pipeline SumeCzech headlines created by pre-trained *mbart*. Tables 5.1 and 5.2 show the resulted calculated correlations. We can see that namely **FERNET-C5** and **xlm-roberta-xnli** performed the best out of the presented.

	Correlations			
Models	ann/nli_y	rouge-1/nli_y	rouge-2/nli_y	rouge-l/nli_y
xlm-roberta-large	0.025	0.272	0.272	0.240
bert-base-mode	-0.061	**0.272**	0.272	0.240
xlm-roberta	0.0857	0.2718	0.2718	0.240
FERNET-C5	0.150	0.272	0.2718	0.240
xlm-roberta-xnli	-0.132	0.272	0.272	0.240

Table 5.1: Correlations between modified output of the NLI model and selected metrics - headlines/sumeczech.

	Correlations			
Models	rge-1-cs/nli_y	rge-2-cs/nli_y	rge-l/nli_y	mms-cs/nli_y
xlm-roberta	-0.180	0.070	0.070	0.127
bert-base-mode	-0.180	0.070	0.070	0.090
xlm-roberta	-0.180	0.070	0.070	**0.232**
FERNET-C5	-0.180	0.070	0.070	0.217
xlm-roberta-xnli	0.180	0.070	0.070	0.082

Table 5.2: Correlations between modified output of the NLI model and selected metrics - headlines/sumeczech.

5.1.2 CNC

Secondly, we have ran through the pipeline CNC headlines created by pretrained *mbart*. Tables 5.3 and 5.4 show the resulted calculated correlations. We can see that namely **FERNET-C5** performed the best out of the presented. WIth highest correlation with the annotated dataset and above-average correlation with other metrics like MEMES-CS.

	Correlations			
Models	ann/nli_y	rouge-1/nli_y	rouge-2/nli_y	rouge-l/nli_y
xlm-roberta	0.235	**0.244**	**0.244**	0.210
bert-base-model	0.187	-0.192	-0.192	-0.223
xlm-roberta	0.238	-0.289	-0.289	-0.341
FERNET-C5	**0.280**	-0.149	-0.149	-0.173
xlm-roberta-xnli	**0.258**	-0.114	-0.114	-0.204

Table 5.3: Correlations between modified output of the NLI model and selected metrics - headlines/cnc.

	Correlations			
Models	rge-1-cs/nli_y	rge-2-cs/nli_y	rge-l/nli_y	mms-cs/nli_y
xlm-roberta	0.180	0.241	0.120	**0.124**
bert-base-model	-0.046	-0.095	-0.223	0.233
xlm-roberta-fever	-0.044	-0.181	-0.341	0.053
FERNET-C5	0.133	0.103	-0.173	0.335
xlm-roberta-xnli	0.069	-0.114	0.034	0.123

Table 5.4: Correlations between modified output of the NLI model and selected metrics - headlines/cnc.

5.2 Abstracts

In this section we present the resulted evaluations of the pipeline over generated abstracts following the **mild ruling**.

5.2.1 SumeCzech

Tables 5.5 and 5.6 show the resulted calculated correlations. We can observe that the correlations between standard ROUGE are quite poor. However, the results between NLI and ROUGE-CS show promising signs. The ROUGE-CS should be more Czech-language-prone and it seems the NLI model trained on this dataset correlates with it quite well.

	Correlations			
Models	ann/nli_y	rouge-1/nli_y	rouge-2/nli_y	rouge-l/nli_y
xlm-roberta	-0.164	-0.086	-0.086	-0.188
bert-base-model	0.061	-0.086	-0.086	-0.188
xlm-roberta-fever	-0.019	-0.086	-0.086	-0.188
FERNET-C5	0.052	-0.086	-0.086	-0.188
xlm-roberta-xnli	-0.015	-0.086	-0.086	-0.188

Table 5.5: Correlations between modified output of the NLI model and selected metrics - abstracts/sumeczech.

5.2.2 CNC

Tables 5.7 and 5.8 show the resulted calculated correlations. Once again, we can observe the same trend where standard ROUGE seems to be performing badly in correlation with the NLI. However, *FERNET-C5* shows high correlation with annotated dataset and with MEMES-CS which shows, that here

Models	Correlations			
	rge-1-cs/nli_y	rge-2-cs/nli_y	rge-l/nli_y	mms-cs/nli_y
xlm-roberta	0.1285	0.1303	0.1303	0.213
bert-base-model	0.1285	0.1303	0.1303	0.112
xlm-roberta-fever	0.1285	0.1303	0.1303	0.099
FERNET-C5	0.1285	0.1303	0.1303	0.133
xlm-roberta-xnli	0.1285	0.1303	0.1303	0.205

Table 5.6: Correlations between modified output of the NLI model and selected metrics - abstracts/sumeczech.

the NLI is outperforming standard ROUGE.

Models	Correlations			
	ann/nli_y	rouge-1/nli_y	rouge-2/nli_y	rouge-l/nli_y
xlm-rober	0.0	-0.036	-0.036	-0.123
bert-base-model	0.057	0.013	0.013	-0.212
xlm-roberta-fever	0.026	0.174	0.174	0.152
FERNET-C5	**0.223**	-0.056	-0.056	-0.082
xlm-roberta-xnli	0.0	-0.077	-0.077	0.059

Table 5.7: Correlations between modified output of the NLI model and selected metrics - abstracts/cnc.

Models	Correlations			
	rge-1-cs/nli_y	rge-2-cs/nli_y	rge-l/nli_y	mms-cs/nli_y
xlm-roberta-large	0.123	0.130	0.130	0.321
bert-base-model	0.123	0.130	0.130	0.245
xlm-roberta-fever	0.123	0.130	0.130	0.277
FERNET-C5	0.123	0.130	0.130	0.332
xlm-roberta-large-xnli	0.123	0.130	0.130	0.280

Table 5.8: Correlations between modified output of the NLI model and selected metrics - abstracts/cnc.

5.3 Discussion

In this part, we have presented and briefly discussed the results of the NLI-fact-checking pipeline. The mos testamentary part shall be the last one - evaluating abstracts over CNC. Since the original summarization model *mbart* was trained on the CNC dataset, we have in general supposed higher permormance (i.e. correlations) with the selected metrics than for out-of-set SumeCzech. By looking at tables 5.7, 5.8, 5.3 and 5.4 we conclude that the **FERNET-C5** may be considered as the best performing from the seleted pre-trained models, with high correlations with annotated date and often outperforming standard ROUGE metric. Moreover, we conclude that the MEMES-CS model-based metric shows great potential as the correlations with the NLI model were particularly high when the model correlated with the annotations.

Chapter 6

Conclusion

The thesis was subjected to several guidelines. Firstly, the goal was to prepare an extensive overview of the state-of-art NLP approaches. This was covered more than extensively in the chapter 2. Moreover, in the chapters 2 and 3 we discuss the possibilities and challenges towards Czech language summarization.

In the implementation part, we have successfully implemented and fact-check-NLI pipeline with which we have assessed five different NLI models for the task of assessing facticity. Due to the fact that there was a slight overlap of guidelines and goals with the thesis of Šimon Zvára, we have taken a different approach, and instead of copying or redoing his work, we have taken his work into account, and we have successfully compared it to the selected NLI models with the conclusion that both of these metrics had promising results in terms of correlations that were following the annotated data.

To summarize, this work has added new findings and research to the work of the team from AIC that might be utilized for future development and improvements in Czech summarization.

6.1 Future work

Here we list some topics for future research and improvements:

- Deep sets - examining the behaviour and possible utilization of deep sets algorithms for factual verification of abstractive generated summaries,
- Improving MEMES-CS - the model-based metric is without a doubt the future of evaluation metrics as it is more resistant to changing environments in terms of the used language. One of the future goals would be to continue the work of Šimon Zvára and merge it with the

usage of NLI models. A metric semi-based on pre-trained BERT-based models and NLI models.

- Model-based backtrack correction - after extensive research, the field of ATS is still at the beginning of exploring model-based evaluation approaches. One of the first ones that were found during the research was a backtrack correction model that could help not to evaluate but to improve the summarization task itself.

Appendix A

Bibliography

[1] Meng Cao et al. “Factual Error Correction for Abstractive Summarization Models”. In: *CoRR* abs/2010.08712 (2020). arXiv: 2010.08712. URL: https://arxiv.org/abs/2010.08712.

[2] Jacob Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *CoRR* abs/1810.04805 (2018). arXiv: 1810.04805. URL: http://arxiv.org/abs/1810.04805.

[3] Dr. Michael J. Garbade. *A quick introduction to text summarization in machine learning*. Online; accessed 2022-05-20. URL: https://towardsdatascience.com/a-quick-introduction-to-text-summarization-in-machine-learning-3d27ccf18a9f.

[4] Alexander R. Fabbri et al. “SummEval: Re-evaluating Summarization Evaluation”. In: *CoRR* abs/2007.12626 (2020). arXiv: 2007.12626. URL: https://arxiv.org/abs/2007.12626.

[5] Som Gupta and S. K Gupta. “Abstractive summarization: An overview of the state of the art”. In: *Expert Systems with Applications* 121 (2019), pp. 49–65. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2018.12.011. URL: https://www.sciencedirect.com/science/article/pii/S0957417418307735.

[6] Hugging Face. *all-MiniLM-L6-v2*. Online; accessed 2022-07-28. URL: https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2.

[7] Hugging Face. *distilbert-base-uncased-finetuned-sst-2-english*. Online; accessed 2022-07-28. URL: https://huggingface.co/distilbert-base-uncased-finetuned-sst-2-english.

[8] Hugging Face. *Fill-Mask*. Online; accessed 2022-07-28. URL: https://huggingface.co/tasks/fill-mask.

[9] Hugging Face. *gp2*. Online; accessed 2022-07-28. URL: https://huggingface.co/gpt2.

[10] Hugging Face. *Question answering.* Online; accessed 2022-07-28. URL: https://huggingface.co/tasks/question-answering.

[11] Hugging Face. *roberta-base for QA.* Online; accessed 2022-07-28. URL: https://huggingface.co/deepset/roberta-base-squad2.

[12] Hugging Face. *Sentence similarity.* Online; accessed 2022-07-28. URL: https://huggingface.co/tasks/sentence-similarity.

[13] Hugging Face. *Summarization.* Online; accessed 2022-07-28. URL: https://huggingface.co/tasks/summarization.

[14] Hugging Face. *Text classification.* Online; accessed 2022-07-28. URL: https://huggingface.co/tasks/text-classification.

[15] Hugging Face. *Text generation.* Online; accessed 2022-07-28. URL: https://huggingface.co/tasks/text-generation.

[16] Hugging Face. *Token classification.* Online; accessed 2022-07-28. URL: https://huggingface.co/tasks/token-classification.

[17] IBM Cloud Education. *Natural Language Processing (NLP).* Online; accessed 2022-05-20. URL: https://www.ibm.com/cloud/learn/natural-language-processing.

[18] Wojciech Kryscinski et al. "Evaluating the Factual Consistency of Abstractive Text Summarization". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP).* Online: Association for Computational Linguistics, Nov. 2020, pp. 9332–9346. DOI: 10.18653/v1/2020.emnlp-main.750. URL: https://aclanthology.org/2020.emnlp-main.750.

[19] Faisal Ladhak et al. "WikiLingua: A New Benchmark Dataset for Cross-Lingual Abstractive Summarization". In: *Findings of the Association for Computational Linguistics: EMNLP 2020.* Online: Association for Computational Linguistics, Nov. 2020, pp. 4034–4048. DOI: 10.18653/v1/2020.findings-emnlp.360. URL: https://aclanthology.org/2020.findings-emnlp.360.

[20] 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.* Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 228–231. URL: https://aclanthology.org/W07-0734.

[21] Sweta P. Lende and M. M. Raghuwanshi. "Question answering system on education acts using NLP techniques". In: *2016 World Conference on Futuristic Trends in Research and Innovation for Social Welfare (Startup Conclave).* 2016, pp. 1–6. DOI: 10.1109/STARTUP.2016.7583963.

[22] Mike Lewis et al. “BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension”. In: *CoRR* abs/1910.13461 (2019). arXiv: 1910.13461. URL: http://arxiv.org/abs/1910.13461.

[23] Elizabeth D. Liddy. “Natural language processing.” In: (2001).

[24] Chin-Yew Lin. “ROUGE: A Package for Automatic Evaluation of Summaries”. In: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, July 2004, pp. 74–81. URL: https://aclanthology.org/W04-1013.

[25] Yinhan Liu et al. “Multilingual Denoising Pre-training for Neural Machine Translation”. In: *CoRR* abs/2001.08210 (2020). arXiv: 2001.08210. URL: https://arxiv.org/abs/2001.08210.

[26] Krotil Marian. “Text Summarization Methods in Czech”. Bachelor’s Thesis. Prague, Czech Republic: Czech Technical University in Prague, Faculty of Electrical Engineering Department of Cybernetics, 5/2022.

[27] Harvard Business Review - Joe McKendrick. *AI Adoption Skyrocketed Over the Last 18 Months*. Online; accessed 2022-05-20. 2019. URL: https://hbr.org/2021/09/ai-adoption-skyrocketed-over-the-last-18-months.

[28] Monkey learn. *Natural Language Processing (NLP): What Is It & How Does it Work?* Online; accessed 2022-07-28. URL: https://monkeylearn.com/natural-language-processing/.

[29] Monkey learn. *Text Classification: What it is And Why it Matters*. Online; accessed 2022-07-28. URL: https://monkeylearn.com/text-classification/.

[30] Prakash M Nadkarni, Lucila Ohno-Machado, and Wendy W Chapman. “Natural language processing: an introduction”. In: *Journal of the American Medical Informatics Association* 18.5 (Sept. 2011), pp. 544–551. ISSN: 1067-5027. DOI: 10.1136/amiajnl-2011-000464. eprint: https://academic.oup.com/jamia/article-pdf/18/5/544/5962687/18-5-544.pdf. URL: https://doi.org/10.1136/amiajnl-2011-000464.

[31] Ramesh Nallapati, Bing Xiang, and Bowen Zhou. “Sequence-to-Sequence RNNs for Text Summarization”. In: *CoRR* abs/1602.06023 (2016). arXiv: 1602.06023. URL: http://arxiv.org/abs/1602.06023.

[32] Jun-Ping Ng and Viktoria Abrecht. “Better Summarization Evaluation with Word Embeddings for ROUGE”. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, Sept. 2015, pp. 1925–1930. DOI: 10.18653/v1/D15-1222. URL: https://aclanthology.org/D15-1222.

[33] 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*. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 74–84. DOI: `10.18653/v1/W17-4510`. URL: `https://aclanthology.org/W17-4510`.

[34] Diyah Puspitaningrum. “A Survey of Recent Abstract Summarization Techniques”. In: *CoRR* abs/2105.00824 (2021). arXiv: `2105.00824`. URL: `https://arxiv.org/abs/2105.00824`.

[35] Zhe Quan et al. “An Efficient Framework for Sentence Similarity Modeling”. In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 27.4 (2019), pp. 853–865. DOI: `10.1109/TASLP.2019.2899494`.

[36] Roger Brown. *Where is artificial intelligence used today*. Online; accessed 2022-05-20. 2019. URL: `https://becominghuman.ai/where-is-artificial-intelligence-used-today-3fd076d15b68`.

[37] Horacio Saggion and T. Poibeau. “Automatic Text Summarization: Past, Present and Future”. In: *Multi-source, Multilingual Information Extraction and Summarization*. 2013.

[38] Victor Sanh et al. “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter”. In: *ArXiv* abs/1910.01108 (2019).

[39] Noam Shazeer et al. “Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer”. In: *CoRR* abs/1701.06538 (2017). arXiv: `1701.06538`. URL: `http://arxiv.org/abs/1701.06538`.

[40] Zvára Šimon. “Assessing Facticity in Abstractive Summarization Methods”. Bachelor’s Thesis. Prague, Czech Republic: Czech Technical University in Prague, Faculty of Electrical Engineering Department of Computer Science, 5/2022.

[41] Milan Straka et al. “SumeCzech: Large Czech News-Based Summarization Dataset”. In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA), May 2018. URL: `https://www.aclweb.org/anthology/L18-1551`.

[42] Ashish Vaswani et al. *Attention Is All You Need*. 2017. DOI: `10.48550/ARXIV.1706.03762`. URL: `https://arxiv.org/abs/1706.03762`.

[43] Thomas Wolf et al. “Transformers: State-of-the-Art Natural Language Processing”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. DOI: `10.18653/v1/2020.emnlp-demos.6`. URL: `https://aclanthology.org/2020.emnlp-demos.6`.

Appendix B

Glossary

Symbol	Meaning
AI	artificial intelligence
NLP	natural language processing
ML	machine learning
NLI	natural language inference
ATS	abstractive text summarization
QA	question answering
POS	part-of-speech
NER	named entity recognition
RCI	research center for informatics
RNN	recurrent neural networks
T2A	text to abstract
T2H	text to headline
A2H	abstract to headline
CNC	czech news center
ROUGE	recall-oriented understudy for gisting evaluation
BERT	bidirectional encoder representations from transformer
AIC	artificial intelligence center
CTU	czech technical university