



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY

A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

TRANSFORMER NEURAL NETWORKS FOR NATURAL LANGUAGE PROCESSING

TRANSFORMER NEURAL NETWORKS FOR NATURAL LANGUAGE PROCESSING

SEMESTRÁLNÍ PRÁCE

SEMESTRAL THESIS

AUTOR PRÁCE

AUTHOR

Malinga Tembo

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Jiří Hošek, Ph.D.

BRNO 2024

Semestral Thesis

Master's study program **Communications and Networking (Double-Degree)**

Department of Telecommunications

Student: Malinga Tembo

ID: 243757

**Year of
study:** 2

Academic year: 2023/24

TITLE OF THESIS:

Transformer Neural Networks for Natural Language Processing

INSTRUCTION:

This master's thesis aims to apply Transformer Neural Networks to Natural Language Processing tasks, specifically focusing on text summarization, code generation and question-answering. The objectives of the thesis are as follows:

- Reviewing the state-of-the-art Transformer Neural Networks used in natural language processing. This involves studying existing approaches and understanding their application and performance.
- Comparing the effectiveness and performance of Transformer Neural Networks with traditional Neural Networks in the context of natural language processing tasks.
- Developing and implementing deep learning models based on Transformer Neural Networks for one natural language processing task, text summarization.

Master thesis:

- Developing and implementing deep learning models based on Transformer Neural Networks for multiple natural language processing tasks, such as code generation and question-answering.
- Evaluating and comparing the performance of the developed models on various real-world datasets.
- Evaluating the proposed model and comparing its performance with other models in the literature.

RECOMMENDED LITERATURE:

according to instruction of supervisor

**Date of project
specification:** 1.10.2023

**Deadline for
submission:** 15.1.2024

Supervisor: doc. Ing. Jiří Hošek, Ph.D.

doc. Ing. Jiří Hošek, Ph.D.
Chair of study program board

WARNING:

The author of the Semestral Thesis claims that by creating this thesis he/she did not infringe the rights of third persons and the personal and/or property rights of third persons were not subjected to derogatory treatment. The author is fully aware of the legal consequences of an infringement of provisions as per Section 11 and following of Act No 121/2000 Coll. on copyright and rights related to copyright and on amendments to some other laws (the Copyright Act) in the wording of subsequent directives including the possible criminal consequences as resulting from provisions of Part 2, Chapter VI, Article 4 of Criminal Code 40/2009 Coll.

Contents

Introduction	5
1 Natural Language Processing	7
Natural Language Processing	7
1.1 Problem definition	7
1.2 Proposed Solution	8
1.3 Natural Language Processing Tasks	9
1.3.1 Text Classification	9
1.3.2 Information Extraction and Information Retrieval	9
1.3.3 Question Answering	10
1.3.4 Text Summarization	10
1.3.5 Machine Translation	10
1.4 Natural Language Processing Methods	11
1.4.1 Rule-Based Methods	11
1.4.2 Deep Learning Methods	12
2 Text Summarization	19
2.1 Model Selection	19
3 Conclusion	21
Bibliography	23
Symbols and abbreviations	27

Introduction

Natural language can be defined as system of communication that consists of a set of symbols (words) and rules (grammar) used by individuals within a community to convey meaning, express thoughts, and share information [1]. It serves as a medium for human interaction, enabling the exchange of ideas, emotions, and knowledge. In an attempt at finding effective ways to communicate with machines, the field of NLP emerged as an amalgamation of the study of linguistics and artificial intelligence. The field of NLP constitutes computational techniques aimed at the analysis and representation of human natural language. The primary objective of NLP is to leverage computational methods, facilitating human-like language understanding and proficiency across a spectrum of tasks and applications [2].

NLP can be divided into two main branches: fundamental research and applicative research. Fundamental research addresses general language-related problems like language modeling, morphological analysis, syntactic processing, shallow parsing (chunking), semantic analysis and other low-level NLP base tasks. Applicative research focuses on practical tasks such as extracting information from texts, language translation, document summarization, automatic question answering, document classification and clustering temporal inferences/relationship extraction, word sense disambiguation, named entity recognition, and others [3].

Part one of this thesis in takes a high level overview of NLP, defining the problem and highlighting some common NLP tasks. Different approaches to NLP such as the rule-based approach and the deep learning approach are discussed. Further, an overview of the transformer architecture is presented. This is followed by a review of the state-of-the-art Transformer Neural Networks used in NLP. A Comparison of the effectiveness and performance of Transformer Neural Networks with traditional Neural Networks in the context of natural language processing tasks, from literature, is presented. To close Part one, an implementation of a deep learning model based on Transformer Neural Networks for text summarization task is presented.

1 Natural Language Processing

Natural Language Processing (NLP), also known as computational linguistics, is a branch of artificial intelligence that focuses on the interaction between humans and machines through human languages. It involves developing and deploying systems and algorithms to enable machines to understand, process, and analyze human language in an accurate and natural manner, reflecting NLP's ongoing efforts to achieve effective interaction between humans and machines through human language [4].

Broadly speaking, NLP consists of two major components: Natural Language Generation (NLG) and Natural Language Understanding (NLU) (see Fig. 1.1). NLU seeks to achieve comprehension of natural language. This process involves identifying the intended meaning or semantic form from various possible meanings that can be inferred from a given natural language expression [5]. NLU incorporates analysing the linguist elements of language such as phonology, morphology, semantics, pragmatics, syntax e.t.c. in achieving its aims

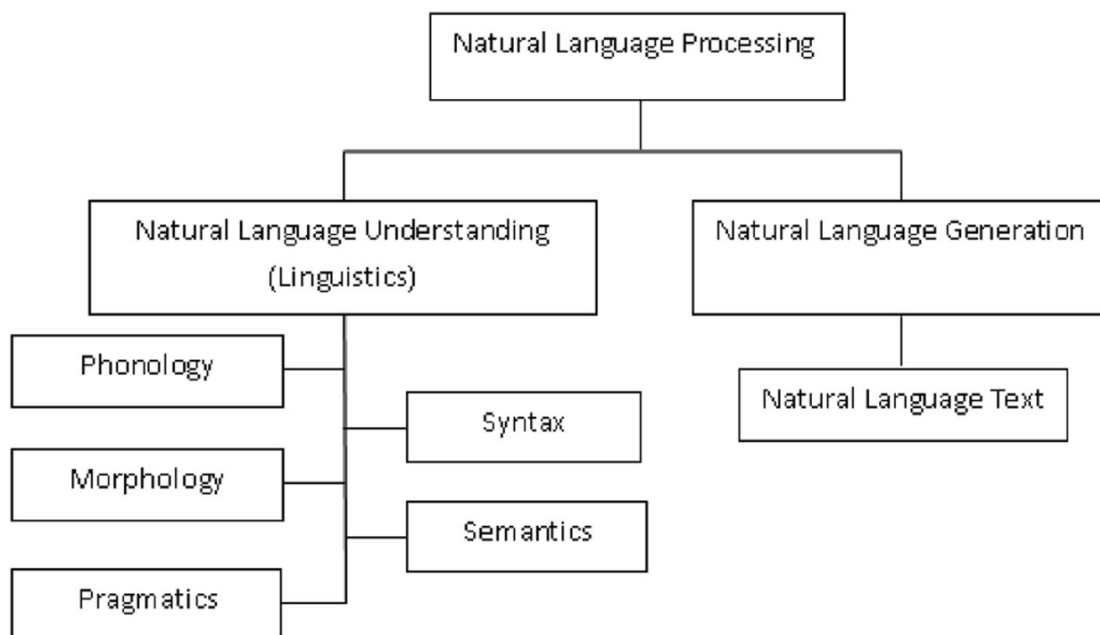


Fig. 1.1: Classification of NLP sub-fields [6].

1.1 Problem definition

At AT&T a significant challenge has emerged from the ever increasing engineering manual documentation. With this increase, significant challenges have arisen, to do with efficiently managing and utilizing an ever-growing repository of diverse and

complex documents such as internal manuals, equipment procedure manuals, user equipment manuals, configuration files e.t.c. This expanding volume of documentation, varying in format and content, has led to critical issues including information overload, inefficient manual processing, challenges in manual inference, and limited interconnectivity between different information sources. These issues not only hinder effective decision-making and workflow efficiency but also slow down operations, leading to potential errors and impeding a comprehensive understanding of interconnected data.

This thesis aims to address these challenges by developing an advanced system leveraging NLP techniques to transform how engineers at AT&T interact with and process this wealth of engineering documentation.

1.2 Proposed Solution

The proposed solution utilizes a transformer-based Large Language Model (LLM) to perform NLP tasks. Figure 1.2 outlines a high-level overview of the steps and elements of the architecture of the proposed solution.

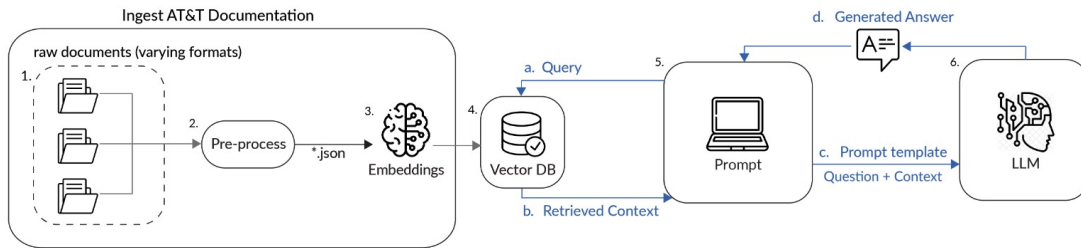


Fig. 1.2: Proposed Solution

Documentation processing:

1. Documentation retrieval using retrieval-augmented generation (RAG)
2. Documentation preprocessing
3. Text-to-vector embedding
 - Transforms text into vectors in semantic space.
 - Vectors reflect contextual meaning and relationships of the original text.

LLM-based document processing and inference:

4. Vector Database
 - Stores generated semantic embeddings of documents.
 - Enables faster and more accurate document retrieval based on context.
5. Prompt to LLM by human
6. Transformer-based LLM
 - Understands prompts and generates human-like responses.

1.3 Natural Language Processing Tasks

In the realm of Natural Language Processing (NLP), the tasks mentioned below frequently emerge due to extensive research and study.

1.3.1 Text Classification

Text classification involves the assignment of predefined labels or categories to text. At its core, text classification involves training a model to learn patterns and features from a labeled dataset, allowing it to generalize and predict the category of new, unseen text. The labeled dataset typically consists of text samples, each associated with a predefined category or label.

The uses of text classification in NLP applications include assigning thematic labels to texts (topic labeling). For instance, news articles can be automatically categorized into topics like politics, sports, or technology. Categorizing questions for more precise responses (question answering), and discerning sentiments within textual data (sentiment analysis), are other NLP examples that use text classification.

1.3.2 Information Extraction and Information Retrieval

Information Extraction involves the automated retrieval of structured information, encompassing entities, relationships between entities, and attributes describing entities, from unstructured sources [7]. Techniques such as named entity recognition and relationship extraction utilize machine learning and pattern recognition to identify entities, their connections, and relevant attributes in unstructured text. For instance, named entity recognition algorithms can classify entities like names, organizations, or locations, enhancing the extraction of valuable information from diverse textual sources.

In Information Retrieval, achieving streamlined access to intended information involves the use of algorithms like Term Frequency-Inverse Document Frequency (TF-IDF) and BM25, which assess the relevance of documents based on word occurrences and document structure. Additionally, advanced retrieval models, including vector space models and neural information retrieval, contribute to the efficiency of retrieving relevant information. These algorithms enable the rapid identification and retrieval of documents that best match user queries, supporting applications such as Question Answering in NLP.

1.3.3 Question Answering

The Question Answering (QA) task, requiring the precise delivery of accurate responses to user queries sourced from an extensive repository of documents or a database. Positioned as a subset of information retrieval, a QA system autonomously furnishes correct answers to human-posed questions in natural language. This achievement is realized through the utilization of a meticulously structured database or an amalgamation of natural language documents. Notably distinct from conventional search engines, a QA system selectively presents only the pertinent information sought, thereby obviating the necessity to sift through entire documents for relevant content [8, 9].

1.3.4 Text Summarization

Text summarization in NLP is accomplished through a combination of sophisticated algorithms and linguistic models. These algorithms scrutinizing the relationships between words, sentences, and paragraphs. In extractive summarization, statistical and machine learning methods identify and extract the most informative sentences directly from the source material. On the other hand, abstractive summarization involves generating new sentences that convey the essential meaning of the original text. This process often includes paraphrasing and synthesizing information to create a more condensed yet coherent representation.

NLP-driven text summarization models often utilize techniques such as sentence scoring, where sentences are assigned weights based on their importance, and attention mechanisms, which enable the model to focus on specific parts of the text. These models can also incorporate semantic understanding to ensure that the summarized version not only captures key information but also maintains the context and nuances present in the original text. By using linguistic analysis and algorithmic efficiency, NLP-based text summarization provides a condensed version of the content, allowing users to quickly grasp the main points and crucial details while preserving the essential meaning.

1.3.5 Machine Translation

Machine Translation (MT) is a task dedicated to the automated conversion of text or speech from one language to another [10]. It encompasses both rule-based and data-driven approaches. Rule-based systems adhere to predefined linguistic rules, striving to replicate grammatical nuances, while statistical and neural machine translation rely on extensive parallel corpora to learn associations between words and

phrases, enabling dynamic and context-aware translations. Similar to summarization, MT can be either literal or creative. Literal translation aims for word-for-word equivalence, resembling extractive summarization, while creative translation involves generating contextually relevant sentences, akin to abstractive summarization techniques.

In essence, NLP-driven MT utilizes linguistic analysis and algorithmic efficiency to provide accurate and contextually rich translations. This not only facilitates global communication by overcoming language barriers but also respects the intricacies of each language, a parallel goal to text summarization in distilling information coherently. Various techniques, such as sentence alignment, sentence scoring, and attention mechanisms—commonly employed in summarization are crucial in MT. Sentence alignment ensures coherence, scoring mechanisms assign importance weights, and attention mechanisms enable focused translation. Semantic understanding plays a pivotal role, extending beyond mere word matching to capture contextual nuances and idiomatic expressions.

1.4 Natural Language Processing Methods

1.4.1 Rule-Based Methods

In the early stages of NLP, rule-based approaches were employed [11]. Rule-based systems, also known as production systems or expert systems, are a basic form of artificial intelligence. By relying on rules as the fundamental method for knowledge representation and encapsulating encoded information, these systems are inherently deterministic in their operation [12, 13]. Rule-based methods are designed to imitate the problem-solving logic employed by human experts in tackling knowledge-intensive problems. The rules are typically expressed as **if-then** statements, outlining how the system should act based on the given conditions. Rule-based systems are often used in various NLP applications, providing a structured and interpretable way to handle linguistic patterns and make decisions based on predefined rules. Early attempts at using rule based approaches in NLP were performed in the domain of machine translation for word to word translation of Russian to English [14]. However, this naive approach to language translation was clearly not robust enough to handle homographs and metaphorical expressions. For example the biblical verse, “*The spirit is willing, but the flesh is weak*” translated to “*The vodka is agreeable, but the meat is spoiled*” [15].

Noam Chomsky’s important 1956 theoretical analysis of language grammars explained the difficulties of language modeling [16]. This analysis led to the creation

of Backus-Naur Form (BNF) notation in 1963, a tool pivotal in specifying context-free grammars (CFG). A grammar is a set of rules that determines how elements in a language can be combined [1]. Specifically, a CFG describes a language by specifying how strings are generated from the initial symbol through a sequence of rewriting steps based on defined rules [17].

Widely used in representing programming-language syntax. Chomsky’s insights extended to the identification of more restrictive regular grammars, serving as the foundation for regular expressions used in text-search patterns. This, in turn, significantly impacted language implementation by generating code and lookup tables for lexical and parsing decisions.

However, these methods fell short in effectively capturing the expansive and unrestrained characteristics of natural language. Their primary shortcomings include the inability to derive meaning from text and to handle spoken prose that, while easily comprehensible to humans, may lack grammatical correctness [15]. Hand-crafted approaches often require human experts to define rules, needs to be domain expert and programmer and have sound understanding of linguistics, essential to craft robust extraction rules. Handcrafted approaches often require individuals who are both domain experts and proficient programmers, possessing a sound understanding of linguistics to define the necessary rules for robust extraction. These NLP systems, like other AI systems of that time, were based on constructing task-specific rules. They were built specifically for the tasks at hand. However, this approach rendered them not generally useful beyond their designated functions [10].

1.4.2 Deep Learning Methods

With increases in computational power, more novel methods in NLP based on neural networks have emerged. The introduction of recurrent neural networks (RNNs) vastly improved upon the ability to account for sequential dependencies in NLP tasks [18]. RNNs found application in many supervised NLP tasks mainly to do with classification and regression thanks to long short term memory (LSTM) [19]. However, the biggest drawbacks with RNNs is that their sequential nature hinders the ability to parallelize them on multiple processes. They also suffer from the problem of vanishing and exploding gradients during backpropagation. Consequently, RNNs perform poorly in NLP tasks where they have to capture long term dependencies [20].

The transformer architecture, by Vaswani et al., in the paper “Attention Is All You Need” [22], proposed a new architecture to mitigate RNN bottlenecks adopting a non-recurrent architecture that depend on attention. Attention can be defines as, “the capacity to selectively concentrate on a chosen stimulus, maintaining that

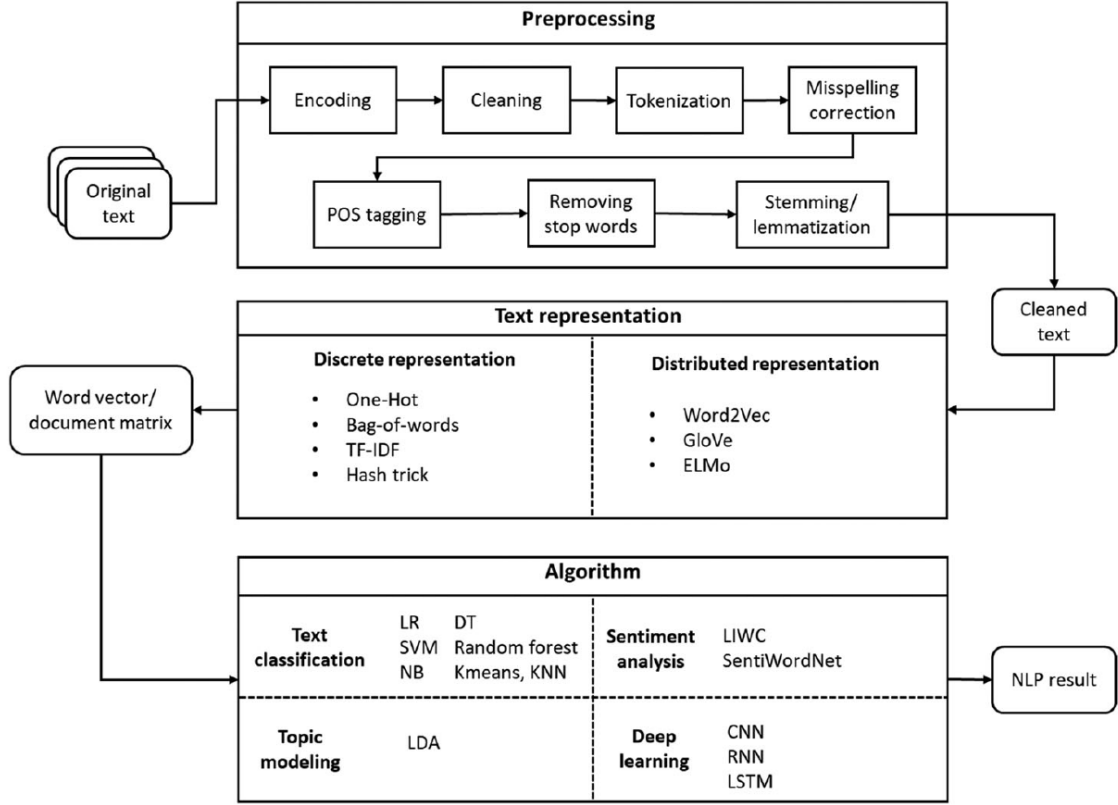


Fig. 1.3: Processing pipeline in statistical and deep learning-based NLP [21].

concentration, and having the flexibility to shift it as desired” [23]. Self-attention is a mechanism that allows the model to weight the importance of the words in any given sequence to all other words in the sequence. These weights inform the model on which words to pay attention to thereby accounting for intricate patterns and contextual relationships in language.

Recurrent Neural Networks

Neural networks are a class of machine learning models inspired by the structure and functioning of the human brain. In the context of neural networks, the term “network” highlights the fact that the artificial neurons, or perceptrons are densely interconnected and organized into multiple layers. The strength of connectivity between the neurons is determined by the weights of the edges connecting two neurons [24] as depicted in Fig. 1.4. The weights are adjusted during the training procedure by passing the input data to the input layer, propagating intermediate computations through the network, computing the output, and comparing it to the desired output. The weights can be further optimized by computing derivatives of the weights using the backpropagation procedure and performing gradient descent.

In neural network, every neuron only receives the results of computations at the previous layer and does not memorize its previous states. It uses only the weights adjusted during the training procedure and relies solely on the current input information for predictions, expressed as $p(y_t|x_t)$. Neural networks typically consist of numerous processors functioning concurrently and organized into layers. The input layer processes the raw input information, while subsequent layers receive their inputs from the previous ones. The lack of ability to memorize the sequences of previously seen data during the training procedure limits the scope of its applications only to those problems, where such temporal dependencies can be disregarded.

RNNs address this limitation by forming connections between units in a directed cycle as demonstrated in Fig. 1.5a. This cyclic structure creates an internal state within the network, enabling it to demonstrate dynamic behavior. In contrast to the feed-forward neural networks discussed previously, RNNs leverage their internal memory as an additional context for neurons about the previous input. The ability to retain memory about previous states makes RNNs particularly effective in handling sequential data, allowing them to capture and utilize context from preceding inputs, $p(y_t|x_t, h_{t-1})$.

In feed-forward neural networks, the input layer neurons receive data and convey

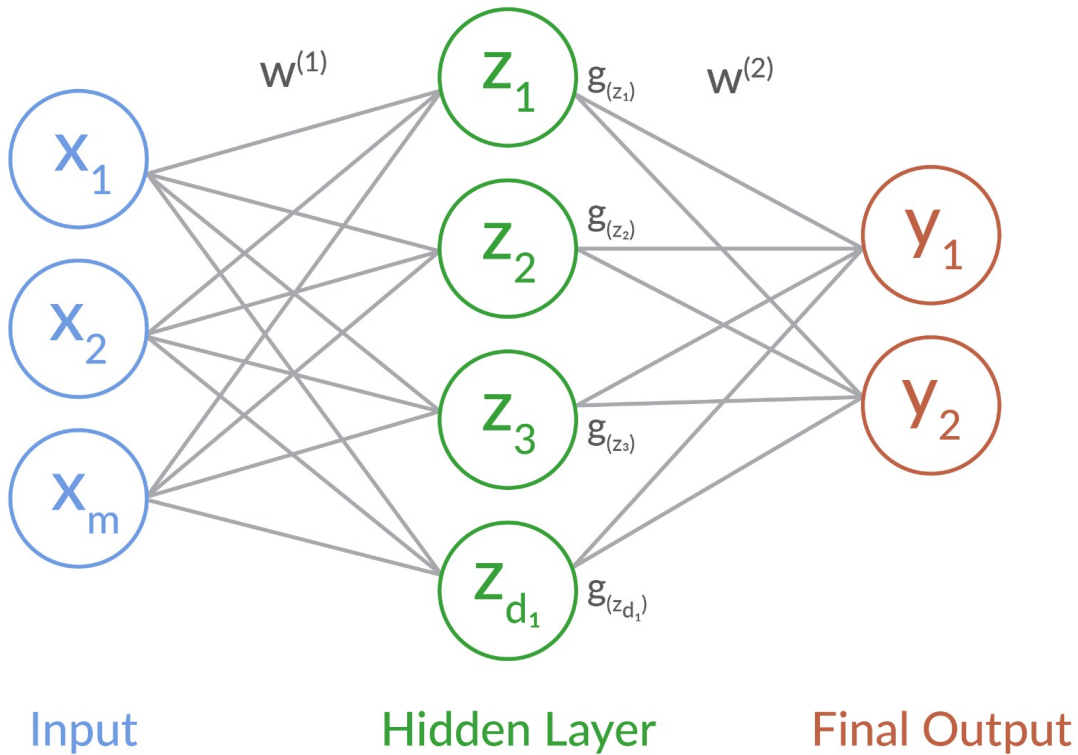


Fig. 1.4: General structure of neural network with one hidden layer

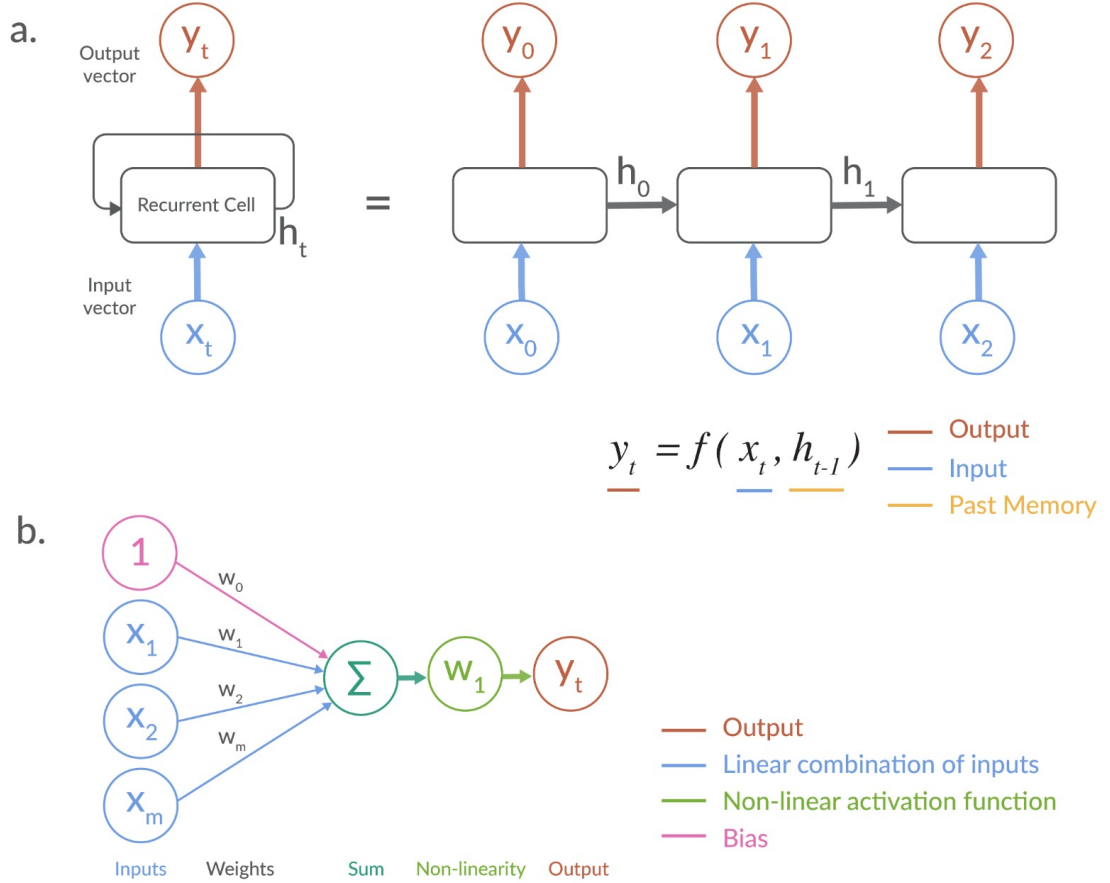


Fig. 1.5: a. Unrolled RNN [20]. b. Perceptron, forward propagation

it to neurons in the initial hidden layer through weighted connections. Subsequently, the data undergo mathematical processing (see eq. 1.1), and the results are transmitted to neurons in the subsequent layers until the output layer. The output of the network is determined by the neurons in the final layer, where the results are first passed through a non-linear function such as sigmoid before being outputted. The computation performed by the z -th neuron in a hidden layer involves calculating the weighted sum and incorporating a bias term $w_{0,i}^{(1)}$ following the expression

$$z_i = w_{0,i}^{(1)} + \sum_{j=1}^m x_j w_{j,i}, \quad (1.1)$$

The output is given by

$$y = g \left(w_{0,i}^{(1)} + \sum_{j=1}^{d1} z_j w_{j,i} \right), \quad (1.2)$$

where $g(z)$ is a non-linear activation function, in the case the sigmoid function $\sigma(z) = \frac{1}{1+e^{-z}}$ [24].

Neural networks are defined by three essential elements: i) the architecture, which determines the number of layers and nodes in each layer, ii) the learning mechanism that governs the adjustment of connection weights, and iii) the activation functions $g(z)$ introducing non-linearity and determining the output of each node within different layers [24].

The training procedure of RNNs consists in measuring the disparity between the predicted output and the actual target values. The goal is to minimize this loss, effectively enhancing the network’s ability to make accurate predictions. Common loss functions include Mean Squared Error for regression tasks and Cross-Entropy Loss for classification tasks. The backpropagation algorithm is used for the training of neural networks. It propagates the measured discrepancy backward through the network, attributing the error contribution of each neuron in each layer. This information is then used to adjust the neural network parameters in a way that they minimize the overall loss. The iterative application of backpropagation and weight adjustments refines the network’s ability to generalize and make accurate predictions on new, unseen data.

The potential of the hidden state, h_t , in each Recurrent Neural Network (RNN) cell to theoretically preserve information over an indefinite duration seems like a promising feature. However, the practical execution of updating the hidden state at each time step, $p(y_t|x_t, h_{t-1})$, presents challenges associated with handling long-term dependencies during the training process. A high number of time steps can lead to the accumulation of gradients during backpropagation. This accumulation, if unchecked, results in exploding gradients, where the gradients become excessively large, causing numerical instability and hindering the convergence of the model. The core of this issue lies in the inherent nature of the chain rule concerning the sequential architecture of RNNs. As the gradients are propagated backward through time, they undergo repeated multiplication. When this multiplication involves numbers greater than one, the gradients grow exponentially, leading to the explosion phenomenon and when the gradients are less than one, leading to vanishing gradients product.

To tackle the challenge of exploding gradients in RNNs, practitioners frequently turn to gradient clipping as a solution. This method introduces a cap on the backpropagation gradients, ensuring they do not surpass a predefined threshold. By restraining the magnitude of gradients, clipping acts as a stabilizing mechanism, preventing the learning process from being disrupted by excessively large gradients. Gradient clipping is not universally adopted due to task and network-specific effectiveness. In scenarios with minimal exploding gradient concerns, it is seen as unnecessary, adding complexity. Selecting an optimal threshold is challenging, risking under-constraint or ineffective mitigation. Specialized architectures like LSTM or GRU networks offer alternatives, addressing gradient challenges without explicit

clipping.

Applied to an NLP task, the input x_t may represent either a string of characters or a single character, while h_t denotes the output state of the RNN. The objective is to utilize h_t as the output and assess its alignment with the test data, typically a limited subset of the original dataset. Subsequently, an error rate is computed based on this comparison and employed in the process of backpropagation through time. This facilitates the adjustment of weights to enhance the model's performance and produce improved results [20].

RNNs are a compelling option for NLP tasks such as next-word prediction. Their recurrent nature allows them to maintain a memory of past words at each time step, resulting in output that is influenced by previous computations. This characteristic makes RNNs well-suited for tasks requiring an understanding of sequential dependencies in language. However, RNNs are limited by how far back they can look in a sequence, making it challenging, for example, to make a correct word prediction for based on words that occur at the beginning of a long sentence. To solve this bottleneck, RNN are used with LSTM. LSTM is a memory mechanism that saves memory that keeps track of long term relationships or dependencies within the serial data.

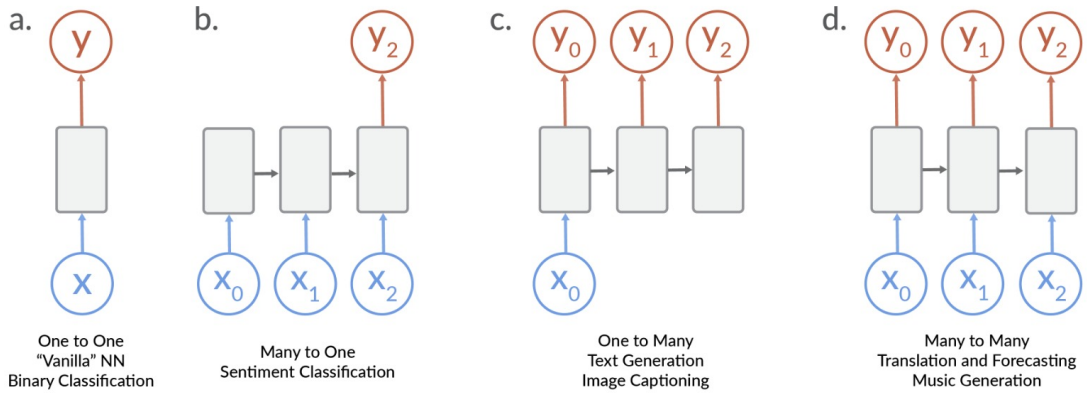


Fig. 1.6: RNN configuration for different NLP tasks [25]

2 Text Summarization

Text summarization is the process of condensing long documents into shorter versions while retaining the main ideas and key information. It can be categorized into two main branches: extractive and abstractive summarization.

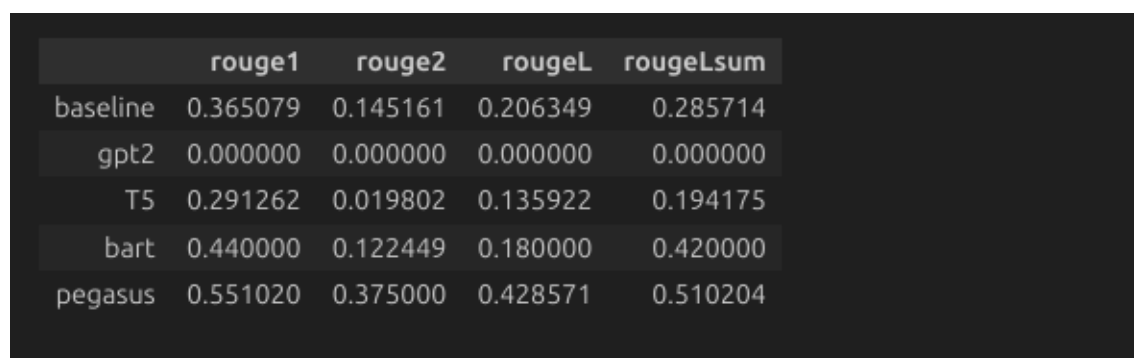
Extractive summarization involves selecting important sentences or phrases from the source text based on their relevance to the theme of the document section.

Abstractive summarization, on the other hand, aims to produce shorter documents conveying the salient ideas of the source document, often containing sentences and phrases that may not be present in the original text.

This section demonstrates automatic text summarization and evaluates it using two transformer-based LLMs: Google-Pegasus and Facebook-BART, with the evaluation criteria being the ROUGE and BLEU scores. The dataset used for this evaluation was the CNN-Daily Mail dataset.

2.1 Model Selection

Pre-trained LLMs GPT-2, T5, Google-Pegasus and Facebook-BART, were initially used to summarize news articles from the CNN-Daily Mail dataset. The evaluation was based on calculating the rouge1, rouge2, rougeL and rougeLsum matrices of the summarization. The base-line shows the rouge scores of human, hand-crafted summarization. The results of the comparisons are shown in the figure 2.1 below.



	rouge1	rouge2	rougeL	rougeLsum
baseline	0.365079	0.145161	0.206349	0.285714
gpt2	0.000000	0.000000	0.000000	0.000000
T5	0.291262	0.019802	0.135922	0.194175
bart	0.440000	0.122449	0.180000	0.420000
pegasus	0.551020	0.375000	0.428571	0.510204

Fig. 2.1: Initial Results

The following is a high level overview of the meaning of evaluation criteria: **ROUGE-1**: Measures the overlap of unigrams (individual words) between the system-generated summary and the reference summary. It computes the precision, recall, and F1 score based on this overlap.

ROUGE-L: Measures the longest common subsequence (LCS) between the system-generated summary and the reference summary. It considers the precision, recall, and F1 score based on the length of the LCS.

ROUGE-Lsum: Is a variant of ROUGE-L that considers multiple reference summaries. It computes the ROUGE-L score for each reference summary separately and then averages these scores to give the final result.

3 Conclusion

From the comparison of the rouge scores against human summarized baselines, the Google-Pegasus and Facebook-BART showed better results than the GPT-2 and T5 models.

It is not surprising that the GPT-2 LLM gave the worst performance initial results owing to the fact, it is primarily an autoregressive language model in contrast to Google-Pegasus and Facebook-BART which incorporate mechanisms such as encoder-decoder architectures, attention mechanisms, and task-specific modifications better suited for text summarization tasks. In-addition, PEGASUS, BART, and T5 are designed with specific pre-training objectives that are more aligned with text summarization tasks whereas GPT-2, is a more general language model.

From these initial results, the LLM models Google-Pegasus and Facebook-BART will be explored further for text summarization in the final part of the thesis.

Bibliography

- [1] John E Hopcroft and Jeffrey D Ullman. *Formal languages and their relation to automata*. Addison-Wesley Longman Publishing Co., Inc., 1969.
- [2] Elizabeth D Liddy. *Natural language processing*. 2001.
- [3] Prakash M Nadkarni, Lucila Ohno-Machado, and Wendy W Chapman. Natural language processing: an introduction. *Journal of the American Medical Informatics Association*, 18(5):544–551, 09 2011. [arXiv:https://academic.oup.com/jamia/article-pdf/18/5/544/5962687/18-5-544.pdf](https://academic.oup.com/jamia/article-pdf/18/5/544/5962687/18-5-544.pdf), doi:10.1136/amiaajnl-2011-000464.
- [4] Ivano Lauriola, Alberto Lavelli, and Fabio Aiolli. An introduction to deep learning in natural language processing: Models, techniques, and tools. *Neurocomputing*, 470:443–456, 2022.
- [5] Brojo Kishore Mishra and Raghvendra Kumar. *Natural language processing in artificial intelligence*. CRC Press, 2020.
- [6] Diksha Khurana, Aditya Koli, Kiran Khatter, and Sukhdev Singh. Natural language processing: State of the art, current trends and challenges. *Multimedia tools and applications*, 82(3):3713–3744, 2023.
- [7] Sunita Sarawagi et al. Information extraction. *Foundations and Trends® in Databases*, 1(3):261–377, 2008.
- [8] Sweta P Lende and MM 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)*, pages 1–6. IEEE, 2016.
- [9] Asma Ben Abacha and Pierre Zweigenbaum. Means: A medical question-answering system combining nlp techniques and semantic web technologies. *Information processing & management*, 51(5):570–594, 2015.
- [10] Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta, and Harshit Surana. *Practical natural language processing: A comprehensive guide to building real-world NLP systems*. O’Reilly Media, 2020.
- [11] Alan Mathison Turing. Mind. *Mind*, 59(236):433–460, 1950.
- [12] Crina Grosan, Ajith Abraham, Crina Grosan, and Ajith Abraham. Rule-based expert systems. *Intelligent systems: A modern approach*, pages 149–185, 2011.

- [13] Pradeepta Mishra. Model explainability for rule-based expert systems. In *Practical Explainable AI Using Python: Artificial Intelligence Model Explanations Using Python-based Libraries, Extensions, and Frameworks*, pages 315–326. Springer, 2021.
- [14] John Hutchins. The first public demonstration of machine translation: the georgetown-ibm system, 7th january 1954. *noviembre de*, 2005.
- [15] Prakash M Nadkarni, Lucila Ohno-Machado, and Wendy W Chapman. Natural language processing: an introduction. *Journal of the American Medical Informatics Association*, 18(5):544–551, 2011.
- [16] Noam Chomsky. Three models for the description of language. *IRE Transactions on information theory*, 2(3):113–124, 1956.
- [17] Fabio Mascarenhas, Sérgio Medeiros, and Roberto Ierusalimschy. On the relation between context-free grammars and parsing expression grammars. *Science of Computer Programming*, 89:235–250, 2014.
- [18] Adji B Dieng, Chong Wang, Jianfeng Gao, and John Paisley. Topicrnn: A recurrent neural network with long-range semantic dependency. *arXiv preprint arXiv:1611.01702*, 2016.
- [19] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, nov 1997. doi:10.1162/neco.1997.9.8.1735.
- [20] Kanchan M Tarwani and Swathi Edem. Survey on recurrent neural network in natural language processing. *Int. J. Eng. Trends Technol*, 48(6):301–304, 2017.
- [21] Yue Kang, Zhao Cai, Chee-Wee Tan, Qian Huang, and Hefu Liu. Natural language processing (nlp) in management research: A literature review. *Journal of Management Analytics*, 7(2):139–172, 2020.
- [22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [23] Pierre Baldi and Roman Vershynin. The quarks of attention: Structure and capacity of neural attention building blocks. *Artificial Intelligence*, 319:103901, 2023.
- [24] Magdi Zakaria, AS Mabrouka, and Shahenda Sarhan. Artificial neural network: a brief overview. *neural networks*, 1:2, 2014.

- [25] Daniel Jurafsky and James H. Martin. *Third Edition*. Stanford University Press, Stanford, CA and Boulder, CO, 2020. Draft.

Symbols and abbreviations

NLP	Natural Language Processing
NLG	Natural Language Generation
NLU	Natural Language Understanding
TF-IDF	Term Frequency-Inverse Document Frequency
QA	Question Answering
MT	Machine Translation
NLP	Natural Language Processing
CFG	Context-Free Grammars
RNN	Recurrent Neural Network
LLM	Large Language Model