Investigation of Transitivity Relation in Natural Language Inference

Petro Zdebskyi[1], Andrii Berko [1] and Victoria Vysotska [1,2]

[1] *Lviv Polytechnic National University, S. Bandera Street, 12, Lviv, 79013, Ukraine*
[2] *Osnabrück University, Friedrich-Janssen-Str. 1, Osnabrück, 49076, Germany*

Abstract

Motivation of this work is a data-centric approach of improving model accuracy by improving data quality instead of improving model architecture. The idea is to improve dataset with transitivity relations to help machine learning model learn such dependencies. Alongside with enriching dataset investigate how good is previously trained model in catching such relations. So, basically study can be divided into two main parts investigating dataset and investigating machine learning model trained on such datasets. It was found that the existing model catches transitive dependencies well. It was also investigated that "entailment" relation is more directional that "contradiction" and "neutral".

Keywords

Natural Language Inference, Recognizing Textual Entailment, transitive relation

1. Introduction

The NLI task is a good benchmark for NLU research where two sentences passed to a model and asked to determine the relationship between them by selecting one of 3 options: entailment, neutral, and contradiction. Success in NLI doesn't demand complex machine learning skills, but rather an ability to understand the meaning of sentences, lexical and compositional semantics, as well as phenomena such as quantification, tense, beliefs, modality, and lexical and syntactic ambiguity.

The study solely relies on the MultiNLI dataset, which is considered the largest dataset for Natural Language Inference and supports multiple languages. MultiNLI consists of sentence pairs that are annotated with textual entailment information, and it covers different genres of text. [1]

Premise	Label	Hypothesis
Fiction		
The Old One always comforted Ca'daan, except today.	*neutral*	Ca'daan knew the Old One very well.
Letters		
Your gift is appreciated by each and every student who will benefit from your generosity.	*neutral*	Hundreds of students will benefit from your generosity.
Telephone Speech		
yes now you know if if everybody like in August when everybody's on vacation or something we can dress a little more casual or	*contradiction*	August is a black out month for vacations in the company.
9/11 Report		
At the other end of Pennsylvania Avenue, people began to line up for a White House tour.	*entailment*	People formed a line at the end of Pennsylvania Avenue.

Figure 1: MultiNLI samples example

COLINS-2023: 7th International Conference on Computational Linguistics and Intelligent Systems, April 20–21, 2023, Kharkiv, Ukraine
EMAIL: petro.v.zdebskyi@lpnu.ua (P. Zdebskyi); Andrii.Y.Berko@lpnu.ua (A. Berko); victoria.a.vysotska@lpnu.ua (V. Vysotska)
ORCID: 0000-0002-0478-2308 (P. Zdebskyi); 0000-0003-2892-9519 (A. Berko); 0000-0001-6417-3689 (V. Vysotska)

NLI task is direct relation [2]. Let's check relation in the opposite direction. Instead of making inference from first sentence to second sentence, swap them and make the inference from the second sentence to the first one.

Model that will be used called Roberta. It is huge deep learning model trained by Facebook and fine-tuned on previously mentioned dataset (MultiNLI). It is improved BERT model. Performance was improved by training the model on more data for a longer period of time; removed the next sentence prediction objective; and dynamically changed the masking pattern on to the training data. So, basically it has improved training procedure, which is called Roberta, and it achieves better results on various benchmarks, without finetuning for GLUE or any additional data for SQuAD.

Investigate dataset if second sentence on some samples are the same as first on other samples (strict and fuzzy matching) for creating dataset for analysing transitivity and check Roberta ability of learning transitivity relation on the dataset created on the previous step.

2. Related works. Transformer architecture

The Transformer is a neural network architecture designed for natural language processing (NLP) tasks. It has become one of the most widely used and popular architectures in NLP. This architecture is based on a self-attention mechanism, which allows to better understand the context and relationships between words because the model looks at different parts of the input sequence [1-3]. Unlike traditional NLP models such as convolutional or/and recurrent neural networks, the Transformer is faster and more efficient because it process the entire sequence in parallel [4-7].

It composed of an encoder and a decoder, each of which is made up of a stack of identical layers. The encoder takes the input sequence and produces a sequence of hidden states, while the decoder takes the encoder output and produces the final output sequence. Both the encoder and decoder use self-attention to process the input and produce the output.

One of the main advantages of the Transformer is its ability to handle variable-length input sequences without the need for padding or truncation. This is achieved through the use of positional encoding, which encodes the position of each word in the input sequence as a vector.

The Transformer has achieved state-of-the-art performance on a wide range of NLP tasks, including machine translation, language modeling, question answering, and sentiment analysis. Many popular NLP models, such as BERT, GPT-3, and RoBERTa, are based on the Transformer architecture.

Positional encoding is a technique used in NLP to encode the relative position of tokens in a sequence. In NLP, sequences of tokens are often processed by models such as the Transformer, which are based on attention mechanisms. Since these models don’t have any notion of order they require an additional signal to represent the order of the sequence. Positional encoding achieves this by adding an encoding vector to the embedding of each token that captures the token's position in the sequence. The encoding vector is calculated based on the position of the token in the sequence and the dimension of the embedding space. This encoding vector is added to the token's embedding, which allows the model to differentiate between tokens based on their position in the sequence. For tasks such as language modeling, machine translation, and text classification it is important to capture both the semantics of the tokens and their position in the sequence and the Transformer model with positional encoding can achieve this.

2.1. Encoder and Decoder

Encoder-decoder is a type of neural network architecture commonly used in natural language processing and machine translation tasks. It consists of two main components: an encoder and a decoder.

As it shown in Fig. 1, the encoder takes an input sequence and converts it into a fixed-length vector representation, which captures the essential information of the input. This vector is passed on to the decoder, which generates an output sequence based on the input vector.

Such architecture is well-suited for tasks where the input and output have different lengths because it can handle sequences of variable length for input and output. Examples of such tasks can be machine translation, text summarization.

The encoder contains 6 identical layers stacked together [1-7]. Each of them two sub-layers: a multi-head self-attention, and a fully connected feed-forward network. There is a residual connection around each of the two sub-layers, followed by the normalization layer [8-12].

The decoder takes the vector from encoder and generates an output sequence. The decoder often has residual connections and normalization similarly to encoder.

The encoder typically has a stack of recurrent or convolutional neural networks that process the input sequence one token at a time and produce a sequence of hidden states. It takes an input sequence such as a sentence in one language and transforms it into a fixed-length vector representation called a context vector.

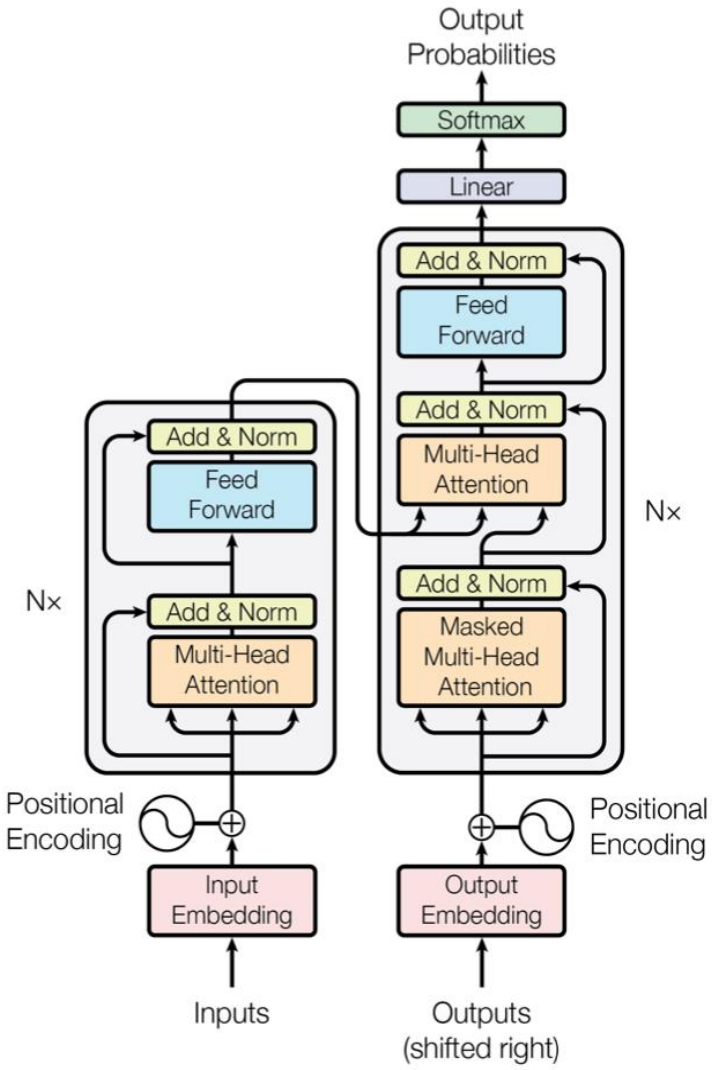

Figure 2: Transformer architecture

2.2. Scaled Dot-Product Attention

Scaled Dot-Product Attention is one of the main components of the Transformer. It is the attention mechanism that helps model to generate the output sequence better by concentrating on the most relevant parts of the input sequence. It computes a weighted sum of values based on a set of key-value pairs where the weights are computed by taking the dot product of the query with each key and then applying a softmax function to normalize the weights. Also, to prevent the gradients from becoming too large dot product is scaled by the square root of the dimension of the key vectors. As a result, weighted sum is then used as input to the next layer in the network.

Calculation of the attention on a set of the queries is stored in a matrix Q. Keys are stored in matrix K and values in matrix V. Formula for Dot-Product Attention:

$$Attention(Q, K, V) = softmax(\frac{QK^t}{\sqrt{d_k}})V \quad (1)$$

There are two attention functions that are used. First is additive and the second is multiplicative attention. In complexity these are similar but the product is faster and more efficient in terms of space because it's implemented with matrix multiplication code that is highly optimized.

If we have small values of d_k then two approaches works similarly but additive attention works better than dot product attention without scaling when d_k have bigger values. The dot products grow and pushes the softmax function where it has extremely small gradients when we have large values of d_k.

Such Attention mechanism is very useful in NLP tasks where the input sequences can be very long because it allows the model to focus only on the most relevant parts of the sequence and as a result it improves the accuracy of the model.

2.3. Multi-Head Attention

Multi-head attention is commonly used in neural network models for NLP tasks like machine translation and text classification. It allows the model to analyse simultaneously distinct parts of the sequence. It has improved ability to analyse difficult relationships between the parts of the input.

It is better to linearly project queries, keys, and values the number of times h with different projections on d_k, d_k, and d_v sizes instead of using single attention with d_{model}-sized keys, values, and queries. Attention function is performed in parallel on predicted versions of queries, keys and values. With that we obtain output values which are d_v-dimensional.

The weighted sum of the input sequence is then computed for each head, and the results are concatenated together to form a single output vector. After that multi-head attention output is passed through a basic feedforward neural network layer. Multi-head focus allows models to share information from different view subspaces in different positions. By averaging we control this because one head of attention is used.

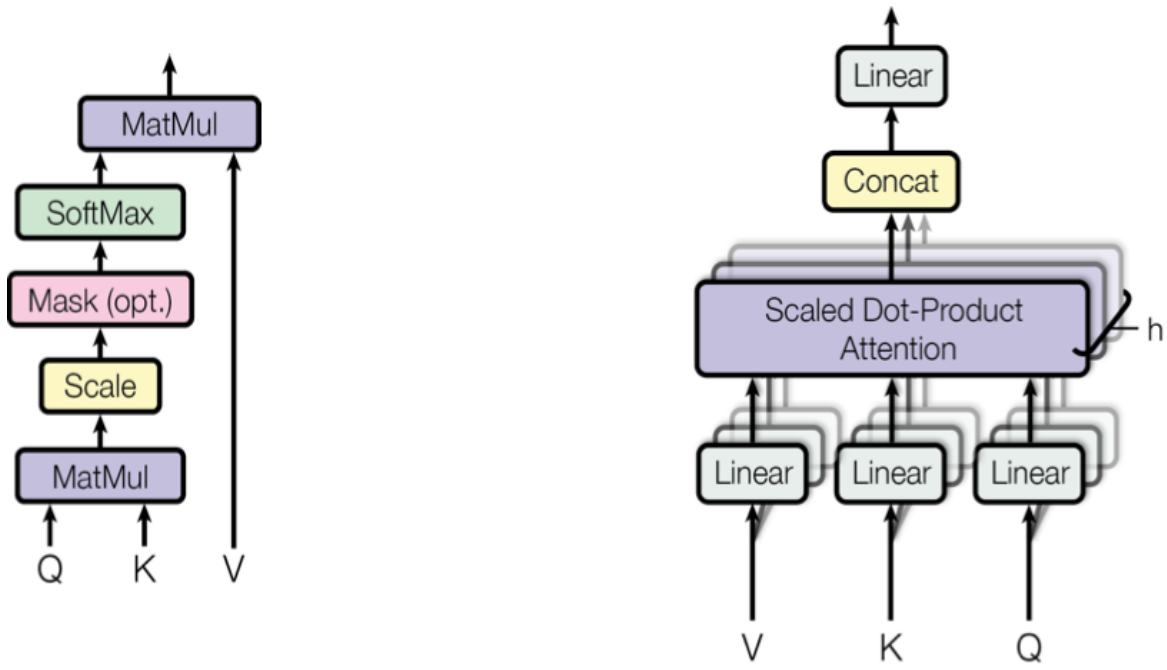

Figure 3: Scaled Dot-Product Attention and Multi-Head Attention

2.4. Application of Attention in the Model

There are three different ways in which multi-headed attention can be used in Transformer:

- In the encoder-decoder attention, keys and memory values passed from the output of the encoder but requests from the decoder last level. Specifically, at each time step in the decoder, the attention mechanism computes an attention weight vector over the encoded input sequence. This allows each position in the decoder to visit all positions in the input sequence. This is very similar to the common approaches of the encoder-decoder attention in the seq-to-seq models.
- *In Self-Attention encoder* approach it has the same attention to itself. In the level of self-awareness everything is coming from the one place. In this approach the output is taken from the encoder previous level. E*ach head of the attention mechanism attends to a different part of the input sequence and computes an attention weight vector*. In the encoder each position can be in any or all positions of the decoder's previous level.
- Similarly, the decoder self-attention allows decoder to be in all positions of decoder until this position. At each time step, the decoder attends to the previously generated tokens in the target sequence to compute an attention weight vector. It's implemented inside the scaled attention of the product point and all values at the softmax are masked input to prevent the left flow of the information.

The Transformer is can to capture complex relation between the input and output by using multi-head attention in these ways, that is why it's one of the most successful model in NLP. [3]

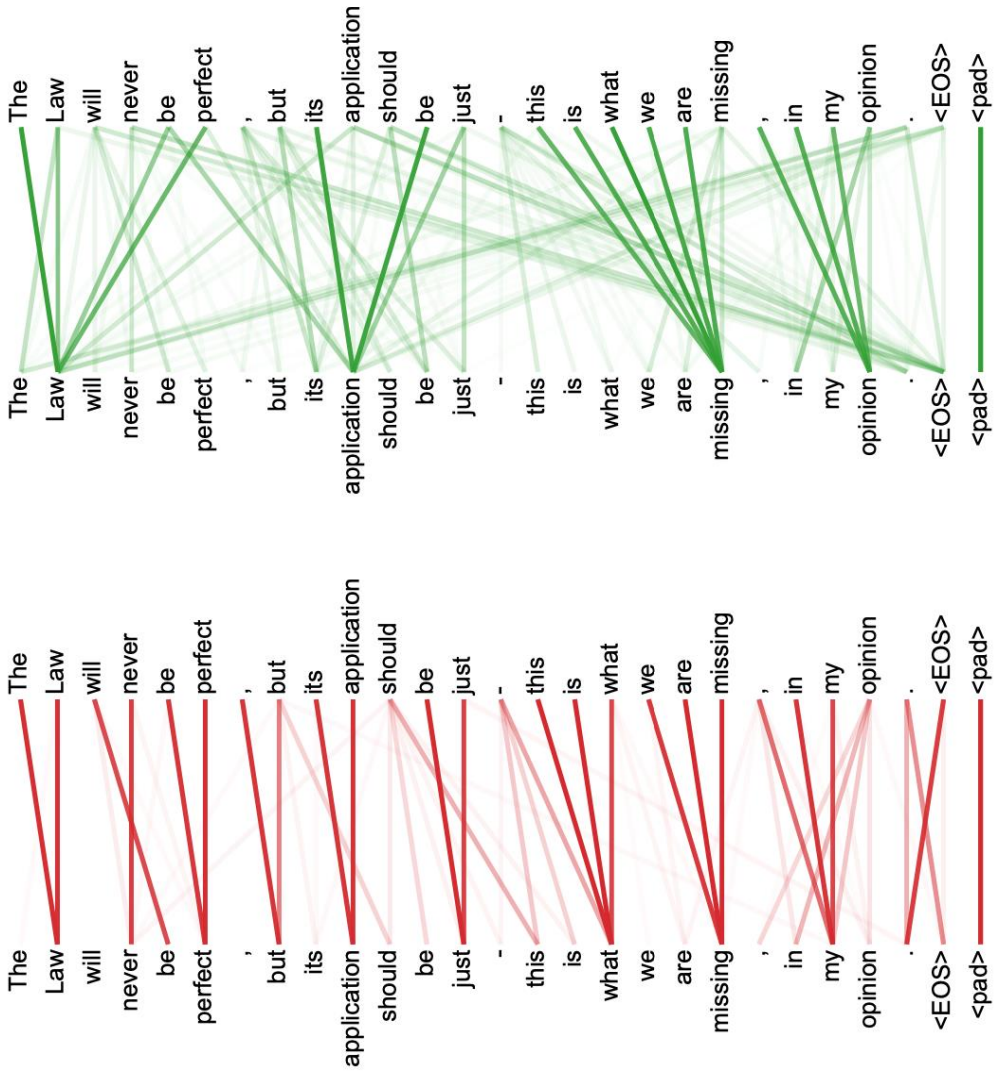

Figure 4: Attention mechanism visualization

3. Methods and technology

3.1. BERT

In this section there is details about BERT implementation. In general, there are two steps in the structure of the model. First is pre-training and the seconds is fine-tuning. The model is trained in unsupervised manner (unlabeled data) in different tasks tasks during pre-training. The model is initialized with trained parameters and then all parameters are adjusted using supervised approach (labeled data) for fine-tuning. For each task there is different fine-tuned models despite the fact that they are initialized with the identical pre-trained parameters.

A distinctive feature of such model is its that for different tasks is uses the same architecture. It means that there is only slight difference between the final architecture and pre-trained architecture.

The BERT model architecture is a multilayer Transformer encoder which is bidirectional. The database for BERT have the same model size as OpenAI first generation GPT model. Although, the GPT Transformer has limited self-awareness so it can only look to the left in the context while the BERT has bidirectional self-awareness. BERT is trained on a task called Masked Language Modeling (MLM), where it is given a sentence with some of the words randomly masked and is asked to predict the masked words based on the context of the other words in the sentence [4]. This helps model to learn the relationships between words and to understand the meaning of a sentence [11]. BERT is also trained on a task called Next Sentence Prediction (NSP), where it is given a pair of sentences and is asked to predict whether the second sentence is the next sentence in the document after the first sentence or not.

The representation of the input can represent a pair of sentences and a single sentence to make model work with different subsequent tasks. The sequence in this context is sequence of tokens passed to model that can be one sentence or two sentences together.

Embedding that is used in BERT called WordPiece embedding. The is special marker at the start of each sequence. There are two ways differentiate between sentences. First approach is to use special character to separate. Another is adding to each marker the learned embedding this says if it belongs to one sentence or another. [4, 11, 13-15]

3.2. Roberta

RoBERTa (Robustly Optimized BERT Pre-training Approach) is a variant of the BERT (Bidirectional Encoder Representations from Transformers) model. But it has massive dataset for pre-training compared to BERT (160 GB of text data). It uses dynamic masking which randomly masks out different tokens at each training epoch. RoBERTa is trained on a masked language modeling task instead of Next Sentence Prediction.

In the fist implementation, random replacement and masking is made once at the beginning and stored during the training but the data is duplicated in practice so for each sentence the mask is not always the same [5]. NSP or Next Sentence Prediction is a binary classification task to predict if two sentences comes one after another in the text. Examples for training crated by extracting consecutive sentences from the training text data while negative created using sentences from different documents.

The Next Sentence Prediction task is used to improve the performance of tasks like as natural language inference that require understanding the relationship between sentences. [5, 12, 16-18]

3.3. MultiNLI dataset

The MultiNLI (Multi-Genre Natural Language Inference) dataset is a popular benchmark dataset for natural language understanding tasks in machine learning. It has 433,000 examples and it makes it one of the biggest dataset for inference in natural language. MultiNLI has higher complexity by using data from 10 genres of written and spoken English. That allows evaluate models for almost complete complexity of the language and provides an obvious settings for assessing domain adaptation.

The MultiNLI dataset consists of sentence pairs drawn from various genres, including fiction, government reports, and conversational speech. Each sentence pair is labeled with one of three relationship types: entailment, contradiction, or neutral.

The NLP tasks depend on understanding natural language or NLU to succeed [19-21]. A lot of work has been done to advance applied understanding natural language tasks so that the model can succeed in these issues. In general model must be accurate in NLU as well as in additional machine learning

tasks like memory access or structured prediction. Given that it makes it pretty difficult to make the correct assessment on how NLP models understand the meaning of language [22-27].

The entailment relationship indicates that the first sentence logically entails the second sentence, meaning that if the first sentence is true, then the second sentence must also be true. The contradiction relationship indicates that the first sentence contradicts the second sentence, meaning that if the first sentence is true, then the second sentence must be false. The neutral relationship indicates that there is no logical relationship between the two sentences [28-36].

Methodology for data collection in MultiNLI is very similar to the SNLI. It has pair of sentences from the previous source and then the annotator was asked to make a new sentence. The dataset was created to encourage research into natural language understanding across multiple genres, rather than just a single genre, which was the case with many previous datasets. This makes it more challenging and realistic for models to perform well on this dataset.

The text for MultiNLI assumption sentences comes from 10 sources of fully available text. Therefore it should be diverse and represent American English. The sentence pairs in the MultiNLI dataset were drawn from ten different genres, including telephone conversations, travel guides, and government documents. This ensures that the dataset covers a wide range of topics and writing styles [37-41].

The gold mark was assigned for each pair of sentences that represent the majority of votes between the mark assigned by the original annotator and the four marks assigned by the verifying annotators. Some small number of sentences did not have a consensus. Such sentences should not be used in standard estimates and are included in the distributed housing but they have a dash as a gold label. As it shown in Fig. 5 dataset has sentence pair, genre, gold label and all labels assigned by annotators. [1]

Met my first girlfriend that way.	FACE-TO-FACE **contradiction** C C N C	I didn't meet my first girlfriend until later.
8 million in relief in the form of emergency housing.	GOVERNMENT **neutral** N N N N	The 8 million dollars for emergency housing was still not enough to solve the problem.
Now, as children tend their gardens, they have a new appreciation of their relationship to the land, their cultural heritage, and their community.	LETTERS **neutral** N N N N	All of the children love working in their gardens.
At 8:34, the Boston Center controller received a third transmission from American 11	9/11 **entailment** E E E E	The Boston Center controller got a third transmission from American 11.
I am a lacto-vegetarian.	SLATE **neutral** N N E N	I enjoy eating cheese too much to abstain from dairy.
someone else noticed it and i said well i guess that's true and it was somewhat melodious in other words it wasn't just you know it was really funny	TELEPHONE **contradiction** C C C C	No one noticed and it wasn't funny at all.

Figure 5: Examples from the development set of MultiNLI dataset which was randomly chosen

3.4. Properties of binary relations

The binary relation can be defined this way "There any sets A and B. Binary relation called R from A to B and can be described formally as R : A × B. Also, this relation is a subset of A × B set."

Reflexive is a relation R on a set A is called reflexive if (a, a) ∈ R for every element a ∈ A. Every vertex has a self-loop.

Figure 6: Reflexive relation

Figure 7: Symmetric relation

If (b, a) ∈ R and (a, b) ∈ R then relation considered symmetric, for all a, b ∈ A. It mean that there is end in the opposite direction if there is edge from one vertex to another. In this case antisymmetric a relation such that for all a, b ∈ A, if (a, b) ∈ R and (b, a) ∈ R.

Antisymmetric relation if for all a, b ∈ A, if (a, b) ∈ R and (b, a) ∈ R. So between distinct vertices there is at most one edge.

Figure 8: Antisymmetric relation

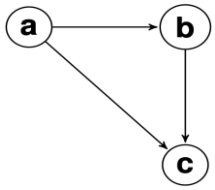

Figure 9: Transitive relation

A binary relation is called transitive if (a, b) ∈ R and (b, c) ∈ R, then (a, c) ∈ R, for all a, b, c ∈ A. It means that is there is any edge from one vertex to another then there is path from one vertex to the second one. [6]

3.5. Textual Entailment as a Directional Relation

In textual entailment task is premise and hypothesis relation is considered directional. For example, let's take "Last year I was in Paris" as a premise and "I was abroad a year ago". In the example we clearly can see that relation is directional, we can entail hypothesis from the premise, but cannot do other way around because if someone was abroad doesn't mean he was in Paris. Only few authors exploited the directional character of the entailment relation, which means that if T → H it is unlikely that the reverse H → T also holds. From a logical point of view, the entailment relation is alike to the implication which, contrary to the equivalence, is not symmetric. [2, 8, 9]

4. Check of how many hypothesis are premises in other samples

Idea is to check if there are hypothesises (second sentences) which are the premises (first sentences) in other samples. That will allow us to create dataset to validate how good is models in catching transitive relation. Strong matching is used, but fuzzy matching can be utilized as well. Sample of the dataset was checked because the dataset is huge and it will take a lot of time to make calculations on the whole dataset.

$$TimeElapsed = (IterationsNumber \cdot IterationTime)^2 \quad (2)$$

Iteration time on my computer was 0.0028, so we can plot the graph of calculation time. So, even for half of the dataset we need 84 hours of calculations.

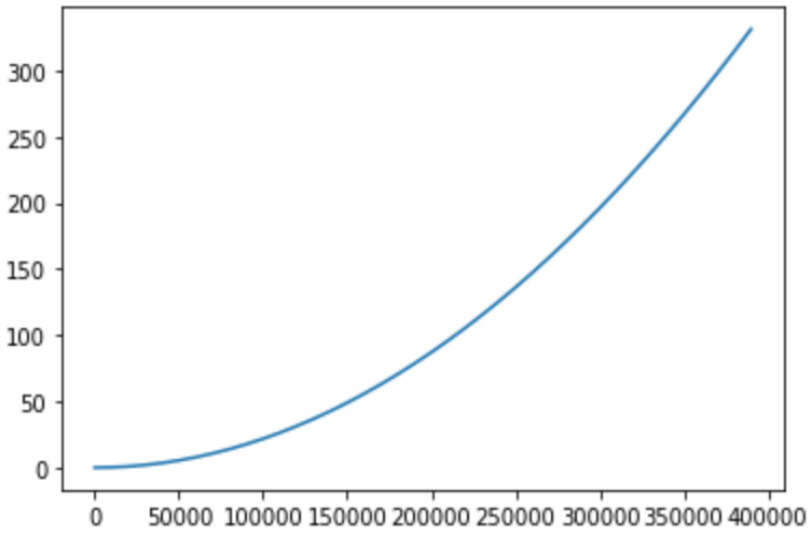

Figure 10: Time complexity graph

5. Discussion and conclusions

5.1. Validation of model entailment with transitive on data

From all of these rules only first one can be considered strictly. If have entailment in first pair and entailment in second one then there should be entailment relation between premise of first pair and hypothesis of second pair. For example first pair: “I have been to Paris” and “I have been to France”, second pair: “I have been to France” and “I visited France”, then “I have been to Paris” and “I visited France” is an entailment. Similarly we can explain for other labels.

Based on domain knowledge such rules were created:

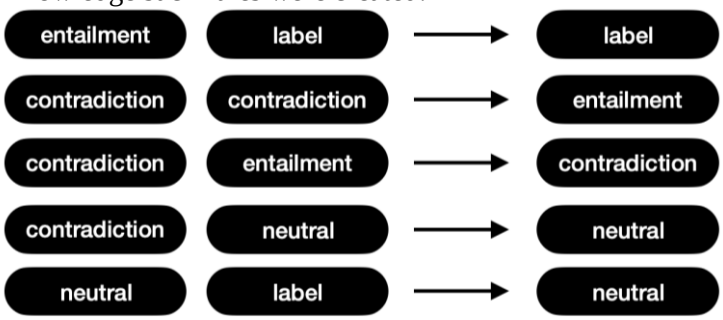

Figure 11: Transitive relation rules

Table 1
Classification of original samples from the dataset

First pair				Second pair			
Premise	Hypothesis	Predicted label	True label	Premise	Hypothesis	Predicted label	True label
I'm afraid not, sir.	I don't think so.	*entailment*	*entailment*	I don't think so.	You are looking in the wrong place.	*neutral*	*neutral*
I'm afraid not, sir.	I don't think so.	*entailment*	*entailment*	I don't think so.	I have no real idea.	*neutral*	*contradiction*
you know and he you know i don't know	I don't know.	*entailment*	*entailment*	I don't know.	I know.	*contradiction*	*contradiction*
Not at all--or at least I don't think so.	I don't think so.	*entailment*	*entailment*	I don't think so.	You are looking in the wrong place.	*neutral*	*neutral*
Not at all--or at least I don't think so.	I don't think so.	*entailment*	*entailment*	I don't think so.	I have no real idea.	*neutral*	*contradiction*
uh-huh that's true that's true yeah	That's right.	*entailment*	*entailment*	That's right.	That's correct.	*entailment*	*entailment*
Quite.'	Absolutely.	*entailment*	*entailment*	Absolutely.	There's no doubt that I'll do it.	*neutral*	*neutral*
Quite.'	Absolutely.	*entailment*	*entailment*	Absolutely.	Definitely.	*entailment*	*entailment*

Table 2
Classification of created samples with transitive relation

Premise	Hypothesis	Predicted label	True label
I'm afraid not, sir.	You are looking in the wrong place.	*neutral*	*neutral*
I'm afraid not, sir.	I have no real idea.	*contradiction*	*neutral*
you know and he you know i don't know	I know.	*contradiction*	*contradiction*
Not at all--or at least I don't think so.	You are looking in the wrong place.	*neutral*	*neutral*
Not at all--or at least I don't think so.	I have no real idea.	*contradiction*	*neutral*
uh-huh that's true that's true yeah	That's correct.	*entailment*	*entailment*
Quite.'	There's no doubt that I'll do it.	*neutral*	*neutral*
Quite.'	Definitely.	*entailment*	*entailment*

In these tables wrong classifications are highlighted with grey background. We can use accuracy metric here, because classes are balanced. Accuracy here equals 75% (6/8*100). Accuracy is calculated based only on 8 samples so it cannot be considered reliable from statistical point of view, but there is restriction related to computational complexity. When we go deeper in analysis of the result we can see that there are 2 errors for the samples in which there are errors on the previous table. For other classes samples is accurate. So, there are troubles only with samples for which model failed even on the original dataset, but if we look at the samples they are really hard to decide that this two sentences are really contradiction. As a conclusion, we can say that in general model catched transitive relation, except for the samples in which model fails on the original dataset. Possible improvement will be to validate samples with class "contradiction" in the dataset model was trained on.

5.2. Model entailment from hypothesises to premises

We've decided to check if what we will get if we swap first and second sentence. Second sentence will be considered as premise and first one as hypothesis [10]. Because the dataset is huge random subset containing 10000 samples was taken from the dataset (calculation time was more than 2 hours).

Table 3
Metrics of classification from hypothesis to premise

	From premise to hypothesis	From hypothesis to premise (reverse order)	Decreased by
Accuracy	0.95	0.63	1.51
Accuracy for class "similar"	0.98	0.33	2.97
Accuracy for class "neutral"	0.90	0.77	1.17
Accuracy for class "contradiction"	0.98	0.78	1.26

In the table we can see that accuracy decreased by a factor of 2.97, 1.17, 1.26 for class similar (entailment), neutral, and contradiction respectively. From this we can infer that similar (entailment)

class is the most directional relation, then contradiction, and then neutral. These number are actually intuitive because if one of the sentences contradicts to another one usually means that it can be in reverse order from second sentence to the first one. For class “neutral” accuracy drops even less compared to “contradiction” because neutral sentences are usually even more often neutral in other direction. As for “entailment” class direction is very strong and we can support that with the numbers, because accuracy became lower approximately by a factor of 3.

6. References

[1] Williams, N. Nangia, S. R. Bowman, A broad-coverage challenge corpus for sentence understanding through inference. arXiv preprint arXiv:1704.05426 (2017).

[2] D. Tatar, G. Serban, A. D. Mihis, R. Mihalcea, Textual entailment as a directional relation. Journal of Research and Practice in Information Technology 41(1) (2009) 53-64.

[3] A. Vaswani, et al. Attention is all you need. Advances in neural information processing systems, 30 (2017).

[4] J. Devlin, M. W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018).

[5] Y. Liu, et al. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019).

[6] William McDaniel Albritton, ICS 241: Discrete Mathematics II, Waterloo, 2015.

[7] P. Zdebskyi, V. Lytvyn, Y. Burov, Z. Rybchak, P. Kravets, O. Lozynska, R. Holoshchuk, S. Kubinska, A. Dmytriv, Intelligent System for Semantically Similar Sentences Identification and Generation Based on Machine Learning Methods, CEUR workshop proceedings Vol-2604 (2020) 317-346.

[8] A. L. Kalouli, A. Buis, L. Real, M. Palmer, V. De Paiva, Explaining simple natural language inference. In Proceedings of the 13th Linguistic Annotation Workshop, 2019, pp. 132-143.

[9] T. Chen, Z. Jiang, A. Poliak, K. Sakaguchi, B. Van Durme, Uncertain natural language inference. arXiv preprint arXiv:1909.03042, 2019.

[10] A. Poliak, A survey on recognizing textual entailment as an NLP evaluation. arXiv preprint arXiv:2010.03061, 2020.

[11] Y. You, et al. Large batch optimization for deep learning: Training bert in 76 minutes. arXiv preprint arXiv:1904.00962, 2019.

[12] L. Z. Liu, Y. Wang, J. Kasai, H. Hajishirzi, N. A. Smith, Probing across time: What does RoBERTa know and when?. arXiv preprint arXiv:2104.07885, 2021.

[13] N. Khairova, A. Shapovalova, O. Mamyrbayev, N. Sharonova, K. Mukhsina, Using BERT model to Identify Sentences Paraphrase in the News Corpus, CEUR Workshop Proceedings, Vol-3171 (2022) 38-48.

[14] P. Zweigenbaum, P. Jacquemart, N. Grabar, B. Habert, Building a text corpus for representing the variety of medical language, Studies in Health Technology and Informatics 84 (2001) 290–294.

[15] H. Livinska, O. Makarevych, Feasibility of Improving BERT for Linguistic Prediction on Ukrainian Corpus. CEUR workshop proceedings Vol-2604 (2020) 552-561.

[16] V. Lytvyn, P. Pukach, V. Vysotska, M. Vovk, N. Kholodna, Identification and Correction of Grammatical Errors in Ukrainian Texts Based on Machine Learning Technology. Mathematic 11, (2023) 904. https://doi.org/10.3390/math11040904

[17] N. Sharonova, I. Kyrychenko, I. Gruzdo, G. Tereshchenko, Generalized Semantic Analysis Algorithm of Natural Language Texts for Various Functional Style Types, CEUR Workshop Proceedings, Vol-3171 (2022) 16-26.

[18] V. Shynkarenko, I. Demidovich, Natural Language Texts Authorship Establishing Based on the Sentences Structure, CEUR Workshop Proceedings, Vol-3171 (2022) 328-337.

[19] O. Kuropiatnyk, V. Shynkarenko, Automation of Template Formation to Identify the Structure of Natural Language Documents. In: CEUR Workshop Proceedings, Vol-2870 (2021) 179-190.

[20] V. Shynkarenko, I. Demidovich, Authorship Determination of Natural Language Texts by Several Classes of Indicators with Customizable Weights. In: CEUR Workshop Proceedings, Vol-2870 (2021) 832-844.

[21] O. Kuropiatnyk, V. Shynkarenko, Text Borrowings Detection System for Natural Language Structured Digital Documents, CEUR workshop proceedings Vol-2604 (2020) 294-305.

[22] V. Lytvyn, S. Kubinska, A. Berko, T. Shestakevych, L. Demkiv, Y. Shcherbyna, Peculiarities of Generation of Semantics of Natural Language Speech by Helping Unlimited and Context-Dependent Grammar, CEUR workshop proceedings, Vol-2604 (2020) 536-551.
[23] O. Bisikalo, Y. Ivanov, N. Karevina, Encoding of Natural Language Information on the Basis of the Power Set. In: 2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT 2018 - Proceedings, 2018, 2, pp. 17–20.
[24] O. Bisikalo, I. Bogach, V. Sholota, The Method of Modelling the Mechanism of Random Access Memory of System for Natural Language Processing. In: Proceedings - 15th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering, TCSET 2020, 2020, pp. 472–477.
[25] O.V. Bisikalo, O.M. Vasilevskyi, Evaluation of uncertainty in the measurement of sense of natural language constructions, International Journal of Metrology and Quality Engineering, 2017, 8.
[26] S. Albota, Linguistically Manipulative, Disputable, Semantic Nature of the Community Reddit Feed Post. In: CEUR Workshop Proceedings, Vol-2870 (2021) 769-783.
[27] S. Albota, Semantic analysis of the Reddit vaccination news feed in the Reddit social network. In Computer science and information technologies: proceedings of IEEE 16th International conference CSIT, Lviv, Ukraine, 22–25 September, 2021, pp. 56–59.
[28] S. Albota, War Implications in the Reddit News Feed: Semantic Analysis. In Computer science and information technologies: proceedings of IEEE 17th International conference CSIT, Lviv, Ukraine, 10–12 November, 2022, pp. 99–102.
[29] I. Khomytska, V. Teslyuk, The Multifactor Method Applied for Authorship Attribution on the Phonological Level, CEUR workshop proceedings, Vol-2604 (2020) 189-198.
[30] I. Khomytska, V.Teslyuk, I. Bazylevych, Y. Kordiiaka, Machine Learning and Classical Methods Combined for Text Differentiation, CEUR Workshop Proceedings, Vol-3171 (2022) 1107-1116.
[31] I. Khomytska, V. Teslyuk, K. Prysyazhnyk, N. Hrytsiv, The Lehmann-Rosenblatt test applied for determination of statistical parameters of Charles Dickens's authorial style. In Computer Science and Information Technologies (CSIT): Proceedings of IEEE XVIth Scientific and Technical Conference. Lviv, Ukraine, 22–25 Sept. 2021, Vol. 2, pp. 64–67.
[32] I. Khomytska, V. Teslyuk, I. Bazylevych, Yu. Kordiiaka, Machine learning and classical methods combined for text differentiation, CEUR Workshop Proceedings, Vol-3171 (2022) 1107-1116.
[33] I. Khomytska, V. Teslyuk, I. Bazylevych, The statistical parameters of Ivan Franko's authorial style determined by the chi-square test. In Computer Science and Information Technologies (CSIT): Proceedings of IEEE XVIIth Scientific and Technical Conference, Lviv, Ukraine, 10–12 November, 2022, pp. 73–76.
[34] V. Husak, O. Lozynska, I. Karpov, I. Peleshchak, S. Chyrun, A. Vysotskyi, Information System for Recommendation List Formation of Clothes Style Image Selection According to User's Needs Based on NLP and Chatbots, CEUR workshop proceedings, Vol-2604 (2020)788-818.
[35] O. Veres, I. Rishnyak, H. Rishniak, Application of Methods of Machine Learning for the Recognition of Mathematical Expressions, CEUR Workshop Proceedings 2362 (2019) 378-389.
[36] Yu.M. Furgala, B.P. Rusyn, Peculiarities of melin transform application to symbol recognition. In: 14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering, TCSET, 2018, pp. 251-254.
[37] E. Fedorov, O. Nechyporenko, Method for Recognizing Linguistic Constructions Based on Stochastic Neural Networks, CEUR Workshop Proceedings, Vol-3171 (2022) 104-115.
[38] S. Kubinska, R. Holoshchuk, S. Holoshchuk, L. Chyrun, Ukrainian Language Chatbot for Sentiment Analysis and User Interests Recognition based on Data Mining, CEUR Workshop Proceedings, Vol-3171 (2022) 315-327.
[39] K. Smelyakov, A. Chupryna, D. Darahan, S. Midina, Effectiveness of Modern Text Recognition Solutions and Tools for Common Data Sources, CEUR Workshop Proceedings 2870 (2021) 154.
[40] R. Martsyshyn, et al. Technology of speaker recognition of multimodal interfaces automated systems under stress. In: International Conference: The Experience of Designing and Application of CAD Systems in Microelectronics, CADSM, 2013, pp. 447-448.
[41] P. Zhezhnych, O. Markiv, Recognition of tourism documentation fragments from web-page posts. In: 14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering, TCSET, 2018, pp. 948-951.