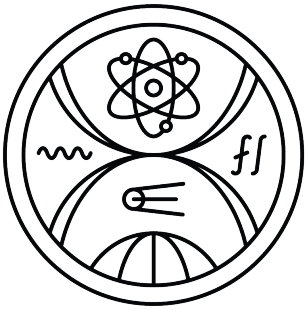
COMENIUS UNIVERSITY IN BRATISLAVA
FACULTY OF MATHEMATICS PHYSICS AND INFORMATICS

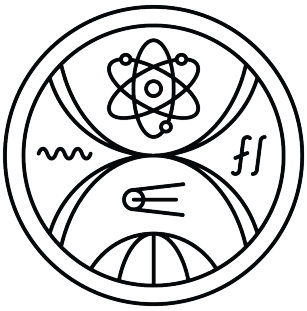


PREDICTION OF HEALTH STATUS DETERIORATION

Master thesis

2025 Bc. Marián Kravec

COMENIUS UNIVERSITY IN BRATISLAVA
FACULTY OF MATHEMATICS PHYSICS AND INFORMATICS



PREDICTION OF HEALTH STATUS DETERIORATION

Master thesis

Study program:	Applied informatics
Branch of study:	Applied informatics
Department:	Department of Applied Informatics
Supervisor:	MSc. František Dráček
Consultant:	

Bratislava, 2025 Bc. Marián Kravec

ZADANIE ZÁVEREČNEJ PRÁCE

Typ záverečnej práce: diplomová
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Predikcia zhoršenia zdravotného stavu
Prediction of Health Status Deterioration

Anotácia: V súčastnosti sa sektor zdravotníctva na Slovensku vyznačuje nizkou mierou využita dostupnych zdravotníckych dát. V rámci tejto prace je cieľom ukázať, že z existujúcjich dát je možné predikovať vyvoj dalšieho zdravotného stavu pacienta, poprípade odhadnúť vývoj budúcich nákladov za účelom lepšieho plánovania prerozdelenia financí v rámci sektoru.

Cieľ: Práca bude rozdelená na dve časti, v prvej študent urobí teoretické zhrnutie existujúchich metód spracovania dát a metód strojového učenia, ktoré sa budú dať potenciálne aplikovať na daný problém. V druhej časti navrhne a aplikuje predičkný model.

Literatúra: T. Sk, L. M. G, L. R. K and R. R. J, "Health Status Prediction using ML Techniques," 2022 6th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2022, pp. 1191-1196, doi: 10.1109/ICCMC53470.2022.9753766.

Jödicke, A.M., Zellweger, U., Tomka, I.T. et al. Prediction of health care expenditure increase: how does pharmacotherapy contribute?. BMC Health Serv Res 19, 953 (2019). https://doi.org/10.1186/s12913-019-4616-x

Vedúci: MSc. František Dráček
Konzultant: Ing. Lukáš Palaj
Katedra: FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry: doc. RNDr. Tatiana Jajcayová, PhD.

Spôsob sprístupnenia elektronickej verzie práce:
bez obmedzenia

Dátum zadania: 05.10.2023

Dátum schválenia: prof. RNDr. Roman Ďurikovič, PhD.
garant študijného programu

.....
študent

.....
vedúci práce

I hereby declare that I have written this thesis by myself, only with help of referenced literature, under the careful supervision of my thesis advisor.

.....

Bratislava, 2025 Bc. Marián Kravec

Acknowledgment

WRITE ACKNOWLEDGMENT

Abstract

ABSTRACT EN

Keywords: TODO

Abstrakt

ABSTRACT SK

Kľúčové slová: TODO

Contents

List of Figures

List of Tables

Terminology

Terms

Abbreviations

- **CPT** - Current Procedural Terminology.
- **EHR** - Electronic Health Records.
- **LLM** - Large language model.
- **LaBSE** - Language-agnostic BERT sentence embedding model.
- **BERT** - Bidirectional encoder representations from transformers.
- **KNN** - K-nearest neighbors algorithm.
- **ICD-10-CM** - International Classification of Diseases, Tenth Revision, Clinical Modification.
- **MKCH-10** - International Classification of Diseases, Tenth Revision (Medzinárodná klasifikácia chorôb).
- **ATC** - Anatomical Therapeutic Chemical.

Motivation

Being able to accurately predict cost of patient in next period is important for both government and health insurance company.

Chapter 1

Introduction

Chapter 2

Similar studies

One of sub-task for prediction of patient future is to group medical procedures into clusters because there are many procedures that even thought have different codes they are essentially same or similar enough that leaving them separate would only cause issue for predicting model.

For this task Lorenzi et al. from Duke University in Durham developed novel algorithm called Predictive Hierarchical Clustering [12]. This algorithm was developed for agglomerative clustering of surgical CPT codes. This algorithm uses one-pass bottom-up approach where they utilize EHR, more precisely using 317 predictors like lab values and patients history, excluding CPT information for 3,723,252 patients and 3,132 CPT codes where each patient have one main surgical CPT code. For each CPT code then they create tree containing patients with that code. Then at each iteration, the algorithm considers merging all pairs of existing trees. To compare two trees they utilize two hypothesis, first hypothesis say that data in both trees are generated from same model, while second say data in each tree is generated from models with different parameters. Final value is weighted average of probabilities of these two hypothesis considering data in trees, where weigth is probability of first hypothesis 2.1.

$$p(D_k|T_k) = p(H_1^k)p(D_k|H_1^k) + (1 - p(H_1^k))p(D_i|T_i)p(D_j|T_j) \tag{2.1}$$

Where D_k is set of data in merged tree (merged T_i and T_j), T_k is merged tree, H_1^k is first hypothesis, D_i and D_j are data in trees T_i and T_j.

From perspective of prediction of patient future one of similar studies is study called Deep Patient by Riccardo Miotto et al. [13] where they were predicting which disease would patient have in the future based on his current state. Their input data were contained general demographic details such as age, gender and race, and common clinical descriptors such as diagnoses, medications, procedures, and lab tests. To predict future diseases they use random forest model with one-vs.-all learning. Study focus primarily on improving results of model by reducing noise in data by reducing their

dimensionality. They compared standard approaches like principle component analysis, Gaussian mixture model or K-means, but main focus was approach using stack of denoising autoencoders. Model using stack of denoising autoencoders to reduce dimension showed significantly better results compared to both model using original dataset and models using other dimensionality reduction techniques.

Another similar study, is study by Caballer-Tarazona er al. [6] in which they tried to predict future cost of the patient primarily using what they called "Aggregated Clinical Risk Group 3" computed from standardized Clinical Risk Group (CRG). This variable consist of the parts, first in one of nine grouped CRGs and second part is one of six levels of severity.

Chapter 3

Medical data

To train and verify model we used anonymized data obtained from Slovak National health information center also known under abbreviation NCZI. Our data consisted of two dataset:

- Records of medical procedures from ambulatory health care
- Records of prescribed medicines

3.1 Records of medical procedures from ambulatory health care

Each row in this dataset contains information about single medical procedure done to patient. Each record consist of these variable:

- date of the procedure - date when procedure was performed
- code of the patient - identification code unique for the patient
- age of the patient - age of the patient at the time of procedure
- gender of the patient
- code of the diagnosis - identification code unique for the diagnosis for which procedure was prescribed
- code of the procedure - identification code unique for the medical procedure
- cost of the procedure - cost associated with performing of the procedure

For our prediction we use most of these information, date of the procedure combined with patient age is used to create timestamp information used to order all records for patient as well as one of the dimension of record embedding. Identification code of

patient is used to be able to gather all records for single patient. Identification codes for diagnosis and procedure are matched with their corresponding numerical vector and embedded into vector corresponding to record (see 4.1). Cost is encoded into cost category and used as information of cost associated with embedding of record.

3.2 Records of prescribed medicines

Similarly to dataset containing procedures, each row of this dataset contains informations about single prescription of drug to specific patient. Each record consist of these variable:

- date of the prescription - date when drug was prescribed
- code of the patient - identification code unique for the patient
- age of the patient - age of the patient at the time of procedure
- gender of the patient
- code of the diagnosis - identification code unique for the diagnosis for which drug was prescribed
- code of the drug - identification code unique for the medical procedure
- cost of the drug - cost associated with performing of the procedure

Use of these variable is also similar to procedures, with only difference that instead of using encoding procedure into record embedding we encode drug.

Chapter 4

Proposed method

Task of predicting future cost of a patient can be split input multiple sub-tasks which follow each other. The sub-tasks are these:

1. Embed patient history into numerical vectors
2. Compute expected number of records patient would have in next year
3. Predict future records for patient
4. Predict cost of each future record
5. Complete total cost of patient for next year

4.1 Embedding of patient

First of sub-tasks for prediction of patient future costs is to embed each patient record into numerical vector that would be understandable for neural network. Some of patient information like age are easy to embed as they are ordered numerical values to begin with, while others are much more tricky. More specifically we needed to figure our how to embed three main information: diagnosis, medical procedure and prescribed drug.

4.1.1 Diagnosis embedding

Base diagnose information we embed was ICD-10-CM code of disease.ICD-10-CM stands for "International Classification of Diseases, Tenth Revision, Clinical Modification" and is used to code and classify medical diagnoses [7] most precisely version of this code that is used in Slovakia and is better known by the acronym MKCH-10-SK (Medzinárodná klasifikácia chorôb) [4].

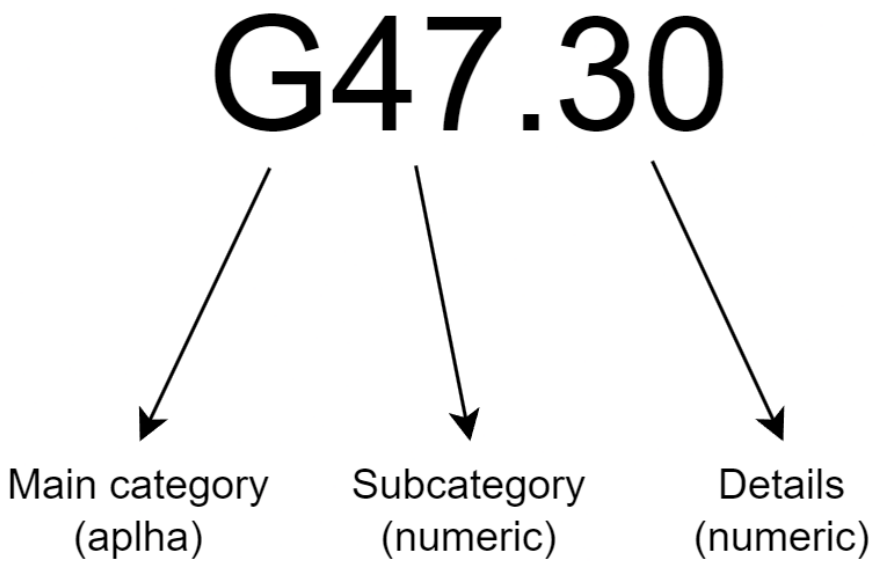


Figure 4.1: Structure of MKCH-10 code

This code consist of three parts as shown on image 4.1. First part is one letter that encodes main categories of diseases also known as chapter, for example codes starting with G are diseases of the nervous system. After that there is two numeric characters that further specify subcategory of disease such as codes from G40 to G47 which are episodic and paroxysmal disorders and specifically G47 are sleep disorders. We can see that episodic and paroxysmal disorders are only up to G47, meaning that theoretically there can exist subgroups G48 and G49 which have 4 in second position but does not belong to same G4 subcategory as G40 or G47 . Thankfully this is not the case and in cases like this when higher lower subcategory (like G4) does not have 10 lower level subcategories (like G47) these subcategories does not exist at all and in case it have more than 10 lower level subcategories it gets multiple consecutive high level subcategories, for example disorders of other endocrine glands spans from E20 to E35. Then code contains dot after which there are characters that further describe details of disease such as etiology, anatomic site and severity. Official documentation of ICD-10-CM codes stands that codes can be up to 7 characters long meaning that after first three characters specifying category there can be up to 4 alphanumeric to further specify the disease CITE, however Slovak version MKCH-10 codes contains at most two numeric characters to specify disease cite and these details are organized in a way where the first position conveys higher-level information than the second, for example G47.3 is sleep apnea and G47.30 is primary central sleep apnea.

To embed this code we firstly split it into it's three parts and embed each part separately. After that we concatenated embedding of each part to get final embedding of disease. To embed main category we generated vector containing random numbers

from interval [-0.5, 0.5] using uniform distribution for each letter of English alphabet (all main categories). We used random vectors in order to have relatively similar distance between any two main categories since there are no particular relationships between these categories, we were thinking also about using one-hot encoding which would make distance between each two categories perfectly same but we didn't do it since it would have restricted length of vector. In embedding second part which is subcategory we used different approach, where to each number between 00 and 99 (all possible values of this part) we assign linearly number between -0.5 and 0.5, meaning subcategory 00 would get -0.5 category 50 would get 0 and category 99 would get 0.5, we choose these boundaries in order to match mean and variance of of numbers generated for main category. This assigned number was then repeated multiple times to create vector. This approach has advantages and disadvantages. Advantage is that we can be sure that closely related disease subgroups like G46 and G47 would get close embedding since their subgroup codes are close on number line. However there are also two disadvantages, first is that G49 and G50 would be similarly close as G46 and G47, but thankfully in a most cases either X9 code doesn't exist at all, creating a gap, or if X9 code exist it belong to category that go past X on as higher level subgroup. Another disadvantage is that distance between two higher level subgroups can vary quite dramatically even though in reality there might not be reason for that difference in distance. For example, using this approach higher level subgroup G40-G47 is much closer to subgroup G50-G59 than to G80-G83. Finally to embed details we decided to use same approach as for subgroups since these codes work similarly, only difference was that not all codes had second level details, in such cases we add 5 as a proxy in order to minimize average distance from all potential codes with same first level details code that contain second level detail information while also maximizing average distance to different first level detail codes. Final after embedding each part final embedding is created as their concatenation, to encode importance of each part in final embedding we gave them different lengths, this works thanks to the fact that each value in each of vector have same mean and same variance. Main category, the most important part, got vector of length 26, subcategory part got length 9 and finally details got length 3. Showcase of resulting embedding can be seen on image 4.2 where each part is highlighted by different color and all values are rounded to two decimal. In order to confirm that our embedding has desired properties we computed similarity of embedding of multiple codes. As similarity function we choose simple multiplicative inverse of Euclidean distance. In table 4.1 we can see results. Highest similarity was between codes G47.30 and G40.09 which is expected since they belong to same main category and very close subcategory, second highest was between H40.09 and H18.80 which are only other combination that belong to same main category, this confirms that main category has biggest impact since this similarity is significantly higher than

that between G40.09 and H40.09 which differ only main category.

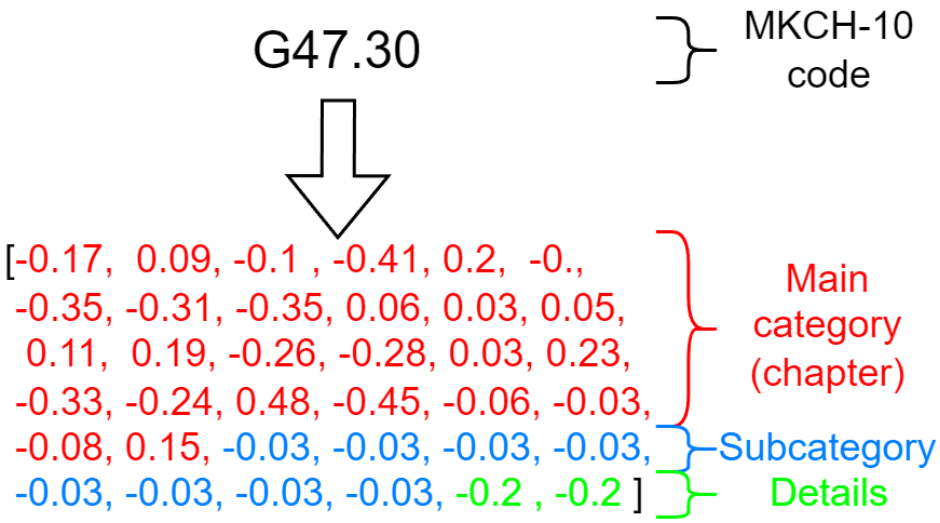


Figure 4.2: Showcase of resulting embedding of specific diagnosis (rounded to two decimal places)

Code A	Code B	Similarity
G47.30	G40.09	2.77
G47.30	H40.09	0.53
G47.30	H18.80	0.46
G40.09	H40.09	0.54
G40.09	H18.80	0.45
H40.09	H18.80	0.84

Table 4.1: Similarities of embedding of multiple chosen MKCH-10 codes

4.1.2 Drug embedding

Similarly to diagnosis, to embed drug information we embed international code associated to these drug. In case of drugs it was Anatomical Therapeutic Chemical classification system also known under abbreviation ATC. In a same way as MKCH-10 code this code can be split into multiple parts where each next part contains finer information. It contains of 5 parts or levels. First level encodes main anatomical or pharmacological groups. There are fourteen such groups, encoded by single letter, which are shown in the figure 4.3. Then second level encodes pharmacological or therapeutic subgroup using two digit number, after that there two levels that further specify pharmacological, therapeutic or even chemical subgroup, these two levels are both encoded using single letter each. Final fifth encoded with two digit number contains information about specific chemical substance inside drug.

Embedding was done in very similar way as in diagnosis embedding, meaning each level was embedded separately and final embedding was done as concatenations of

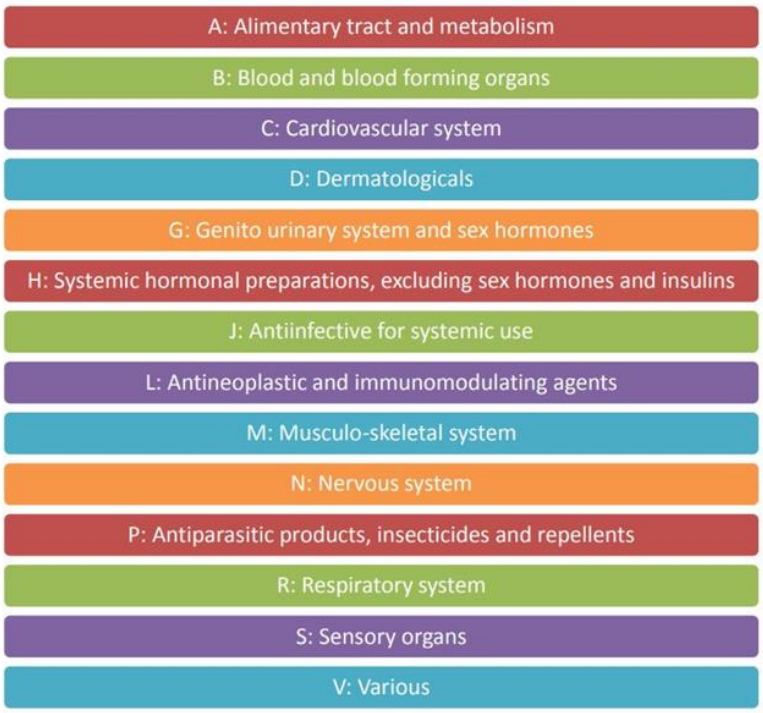


Figure 4.3: Fourteen main anatomical or pharmacological groups and their corresponding first level ATC code [1]

them. In this case each level was embed using random vector from uniform distribution with interval [-0.5, 0.5]. We used random vectors for each level since none of levels contains any internal sub-groupings similar to subgroups of diagnosis (see 4.1.1 subgroup codes). To encode importance of levels we again used lengths of random vectors, where with higher level vectors shortens. Lengths of vectors for each level can be seen in 4.2. So total length of embedding is same as embedding for diagnosis.In case code is incomplete, meaning it's missing higher levels, missing part is substituted with zeros which we consider neutral elements.

Level	Length
1	20
2	10
3	5
4	2
5	1

Table 4.2: Lengths of random vectors assigned to each information level of ATC code

With this embedding we should get codes whose similarity is more dependent on whether lower more important levels match than higher ones. To confirm this we do similar check as we did with diagnosis code and compute similarity of four chosen ATC

codes and see if our theory holds. To compute similarity we again use multiplicative inverse of Euclidean distance. We choose C01EB15, C01CA04, C10AA07 and J01CA04, we would expect first two to be most similar since they match on first levels, than first two compared to third should have slightly lower similarity since they match on only first level and finally we expect that all three would be least similar to fourth one since first level is different, even though that second and forth match on all other levels. We can see results in table 4.3. We can see that results met our expectation with highest similarity between first two and lowest between any of first three and fourth, even in case of second and forth that match on all other levels except first.

Code A	Code B	Similarity
C01EB15	C01CA04	1.24
C01EB15	C10AA07	0.54
C01EB15	J01CA04	0.45
C01CA04	C10AA07	0.64
C01CA04	J01CA04	0.48
C10AA07	J01CA04	0.38

Table 4.3: Similarities of embedding of multiple chosen ATC codes

4.1.3 Medical procedure embedding

Final and most difficult part to embed was medical procedures. In this case there is no structured code that can be used and is implemented in Slovak healthcare systems. So what we have done is to embed description of the procedures. For that we used large language model (LLM) trained on multiple languages including Slovak. Dimensionality of resulting embedding was then reduced using PCA, to get rid of dimensions with small variance meaning they encode small amount of information.

For LLM we choose language-agnostic BERT sentence embedding also known under abbreviation LaBSE. It's a model developed by Goggle to encode text into high dimensional vectors. This model was trained 109 languages including Slovak. Main goal of this model is to generate similar representation to pairs of sentences which have same meaning and are only translations in two different language [3]. This approach should produce better results compared to standard text embedding models trained solely on Slovak language, since LaBSE model is during training comparing embedding not only to similar sentences in Slovak language but also their translation in other which could mitigate a relatively small amount of Slovak language data compared to other more commonly used languages. Additionally this model could know domain specific words in our case medical terms which are left in foreign language and would most likely not be found in Slovak only corpus. To confirm this expectation, we compare results to

Word2Vec model trained on purely Slovak language using corpus containing 110 million words [2]. As a way of comparison we choose to firstly group embedding into categories using K-nearest neighbors algorithm (KNN), and visually asses whether description in resulting categories show any resemblance or whether they seem random.
Architecture of LaBSE model consist of 4 parts [5]:

1. Transformer (BERT model)
2. Pooling layer
3. Dense layer
4. Normalization layer

Transformer (BERT model)

First and most important part of LaBSE model is transformer, which is a deep learning architecture. More specifically LaBSE uses BERT so bidirectional encoder representations from transformers model which is encoder-only transformer architecture meaning this model does not contain decoder found in standard transformer which is usually used for prediction, because of this BERT model is focused in extracting contextual information from input text. Architecture of standard BERT model looks like this:

1. Tokenizer layer
2. Embedding layer
3. Encoder
4. Task layer

First layer is tokenizer, this layer takes input text split it into tokens which in case of BERT model is called PieceWise tokenizer which split text into subwords so something close to syllables. PieceWise tokenizer has advantage compared to different tokenizers that use either words or characters. Compared to character wise tokenizing, subwords contain more information than characters, and compared to word tokenizer is that there a lot less subwords than words and are more similar across multiple languages, creating much smaller vocabulary which is especially important for multilingual models. After split this layer assign integer number to each unique token, LaBSE model vocabulary distinguishes around 500 000 different tokens.

After that comes embedding layer which assign real number vector to each token, more specifically BERT model compute three different embeddings and add them together and normalize result to get final one. First is token type embedding, which is basic embedding where each token in vocabulary is assigned it's unique embedding. Second is positional embedding, as name suggest this embedding contain information about where in the sequence token is found giving additional information. Third and final embedding is segment type, which encodes information about to which segment, usually sentence token belong, important for embedding input text consisting of multiple sentences.

Third and most important layer is encoding. This is the layer in which contextual information are mined from the text. This consist of multiple attention block stacked one after the other. In case of BERT used in LaBSE there are 12 such blocks. Each attention block consist of two parts, multi-headed self-attention layer and feed forward neural network. Each head of self-attention layer takes a input set of embeddings and to compute new set of embeddings which should encode not only original information but also information about relationship between original ones. To do that it compute for each input embedding three vectors usually called key, query and value by multiplying input embedding with three matrices which values are learned during training. After that to get new embedding query vector is multiplied with matrix created from key vectors so resulting vector is vector of dot product of single query and all keys, this vector then goes through softmax function to normalize result. In some transformer models before softmax this vector get masked. Masking is done by setting values where key vector belongs to later embedding than query to minus infinity, this way after softmax this values become zeros. This is done so that later embedding does not affect previous ones. It's mostly useful in models trained to predict next token in order for model to learn to predict only based on tokens from past on not future. However in case of model like BERT where emphasis is on extracting as much information from input text masking is not done. Final step to get new embedding is to multiply vector we got after applying softmax with matrix composed of value vector to get linear combination of value vectors. This resulting vector is new embedding. This is done for all query vectors.

Chapter 5

Software design

Chapter 6

Implementation

Chapter 7

Research

Chapter 8

Results

Conclusion

REFERENCE SHOWCASE: 3

Bibliography

[1] Anatomical Therapeutic Chemical (ATC) Classification — who.int. https://www.who.int/tools/atc-ddd-toolkit/atc-classification. [Accessed 25-09-2024].

[2] GitHub - essential-data/word2vec-sk: Vector representations of Slovak words trained using word2vec — github.com. https://github.com/essential-data/word2vec-sk. [Accessed 19-10-2024].

[3] Google | LaBSE | Kaggle — kaggle.com. https://www.kaggle.com/models/google/labse/tensorFlow2/labse/1?tfhub-redirect=true. [Accessed 18-10-2024].

[4] Medzinárodná klasifikácia chorôb - MKCH-10 — nczisk.sk. https://www.nczisk.sk/Standardy-v-zdravotnictve/Pages/Medzinarodna-klasifikacia-chorob-MKCH-10.aspx. [Accessed 16-09-2024].

[5] sentence-transformers/LaBSE · Hugging Face — huggingface.co. https://huggingface.co/sentence-transformers/LaBSE. [Accessed 19-10-2024].

[6] Vicent Caballer-Tarazona, Natividad Guadalajara-Olmeda, and David Vivas-Consuelo. Predicting healthcare expenditure by multimorbidity groups. *Health Policy*, 123(4):427–434, 2019.

[7] CDC. ICD-10-CM — cdc.gov. https://www.cdc.gov/nchs/icd/icd-10-cm/index.html. [Accessed 16-09-2024].

[8] Yuriy Chechulin, Amir Nazerian, Saad Rais, and Kamil Malikov. Predicting patients with high risk of becoming high-cost healthcare users in ontario (canada). *Healthcare Policy*, 9(3):68, 2014.

[9] Edward Choi, Cao Xiao, Walter Stewart, and Jimeng Sun. Mime: Multilevel medical embedding of electronic health records for predictive healthcare. *Advances in neural information processing systems*, 31, 2018.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

[11] Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. Language-agnostic bert sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, 2022.

[12] Elizabeth C Lorenzi, Stephanie L Brown, Zhifei Sun, and Katherine Heller. Predictive hierarchical clustering: Learning clusters of cpt codes for improving surgical outcomes. In *Machine Learning for Healthcare Conference*, pages 231–242. PMLR, 2017.

[13] Riccardo Miotto, Li Li, and Joel T Dudley. Deep learning to predict patient future diseases from the electronic health records. In *Advances in Information Retrieval: 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20–23, 2016. Proceedings 38*, pages 768–774. Springer, 2016.

[14] Mohammad Amin Morid, Olivia R Liu Sheng, Kensaku Kawamoto, Travis Ault, Josette Dorius, and Samir Abdelrahman. Healthcare cost prediction: Leveraging fine-grain temporal patterns. *Journal of biomedical informatics*, 91:103113, 2019.