1

Detailed planning of the TER2021_66

Classification automatique de questions d'entrainement en langue naturelle dans le domaine médical selon la Taxonomie de Bloom

YACOUB Rémi & YACOUB Nabil

SophiaAntipolis, Nice, France
nabil.yacoub@etu.unice.fr
remi.yacoub@etu.unice.fr

Abstract. In this work, our goal is to help to automate a classification of medical questions with the Bloom's taxonomy. To provide a classifier able of level recognition of Bloom's taxonomy, we first analyzed pretrained models and choose the best that we could use, Wikipedia4G. Later, we investigated a way to increase the efficiency of a model by modifying one parameter at a time. To do that, we observe pattern unique to each level. Then, we observe that the specialist, who labeled some questions, also look at their answers. So we concatenate questions with answers. The results shows no amelioration of the models. After that, we replace all digits values inside each question. The results also shows no benefit for the model. Finally, we look at the weight of each word inside the questions to see if some domains of medicine were more present on some levels than others. The results are not conclusive enough for now.

Keywords: NLP/TALN & Taxonomie de Bloom & Recommandation & Classification de textes & e-Education.

1 Introduction

1.1 Goal

To provide medical students with intelligent learning services, recommending practice questions tailored to their profile and learning goals is the keystone of personalized learning. In this context, the level of complexity and the cognitive objective associated with a question is an important criterion for the recommendation.

The objective of the project is to use machine learning techniques, natural language processing models (NLP / NLP) and advanced data preparation to be able to classify questions according to a taxonomy of cognitive levels.

The taxonomy used is Bloom's revised taxonomy with 6 cognitive levels: remember, understand, apply, analyze, evaluate and create. To do that, we use OntoSIDES[5], Ontology-based student progress monitoring on the national evaluation system of French Medical Schools.

⋆ Université Côte d'Azur . Sophia Antipolis

2 State-of-the-Art

2.1 Bloom's Taxonomy

Bloom's Taxonomy is a classification of the different objectives and skills that educators set for their students (learning objectives). The taxonomy was proposed in 1956 by Benjamin Bloom, an educational psychologist at the University of Chicago. The terminology has been recently updated to include the following six levels of learning. These 6 levels can be used to structure the learning objectives, lessons, and assessments of your course. :

Remembering : Retrieving, recognizing, and recalling relevant knowledge from long-term memory.

Understanding : Constructing meaning from oral, written, and graphic messages through interpreting, exemplifying, classifying, summarizing, inferring, comparing, and explaining.

Applying : Carrying out or using a procedure for executing, or implementing.

Analyzing : Breaking material into constituent parts, determining how the parts relate to one another and to an overall structure or purpose through differentiating, organizing, and attributing.

Evaluating : Making judgments based on criteria and standards through checking and critiquing.

Creating : Putting elements together to form a coherent or functional whole; reorganizing elements into a new pattern or structure through generating, planning, or producing.

Like other taxonomies, Bloom's is hierarchical, meaning that learning at the higher levels is dependent on having attained prerequisite knowledge and skills at lower levels.

2.2 Transformers

A transformer is model architecture consists of a multi-head self-attention mechanism combined with an encoder-decoder structure. It extracts the features for each word using a self-attention mechanism to know the importance of each word in the sentence.

CamemBERT This project use CamemBERT [4] , a French version of the Bi-directional Encoders for Transformers also called BERT [1]. It is based on Facebook's RoBERTa [3] model released in 2019. It is a model trained on 138GB of French text. It can be used for executing part-of-speech tagging, dependency parsing, named entity recognition and natural language inference tasks.

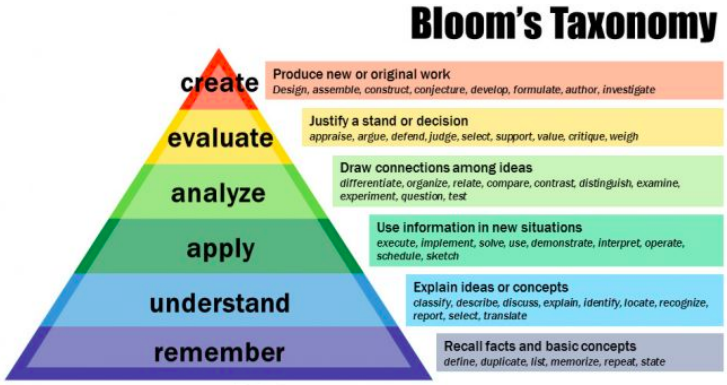

Fig. 1. Pyramidal hierarchy of Bloom's Taxonomy.

Sentence Transformers SBERT [7]] fine-tunes BERT in a siamese / triplet network architecture. It is a modification of the pretrained BERT network to derive semantically meaningful sentence embeddings that can be compared using cosine-similarity. This reduces the effort for finding the most similar pair from 65 hours with BERT / RoBERTa to about 5 seconds with SBERT, while maintaining the accuracy from BERT. SBERT [6] adds a pooling operation to the output of BERT / RoBERTa to derive a fixed sized sentence embedding.

2.3 Metrics

Cosinus Similarity The Cosine Similarity measurement begins by finding the cosine of the two non-zero vectors. This can be derived using the Euclidean dot product formula which is written as:

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$

Then, given the two vectors and the dot product, the cosine similarity is defined as:

$$\mathbf{A} \cdot \mathbf{B} = \|\mathbf{A}\| \, \|\mathbf{B}\| \cos\theta$$

The output will produce a value ranging from -1 to 1, indicating similarity where -1 is non-similar, 0 is orthogonal (perpendicular), and 1 represents total similarity.

Heatmap Heatmaps visualize the data in 2-D colored maps making use of color variations like hue, saturation, or luminance. Heatmaps describe relationships between variables in form of colors instead of numbers.

These variables are plotted on both axes. The color changes describe the relationship between two values according to the intensity of the color in a particular block.

2.4 ROC Curves

A Receiver Operator Characteristic (ROC) curve is a graphical plot used to show the diagnostic ability of binary classifiers. It was first used in signal detection theory but is now used in many other areas such as medicine, radiology, natural hazards and machine learning.

A ROC curve is a plot of the true positive rate (Sensitivity) in function of the false positive rate (100-Specificity) for different cut-off points of a parameter. Each point on the ROC curve represents a sensitivity/specificity pair corresponding to a particular decision threshold. The Area Under the ROC curve (AUC) is a measure of how well a parameter can distinguish between two given groups.

Classifiers that give curves closer to the top-left corner indicate a better performance. As a baseline, a random classifier is expected to give points lying along the diagonal (FPR = TPR). The closer the curve comes to the 45-degree diagonal of the ROC space, the less accurate the test.

2.5 Tf-idf

TF-IDF (Term Frequency Inverse Document Frequency of records) is the calculation of how relevant a word in a series or corpus is to a text. The meaning increases proportionally to the number of times in the text a word appears but is compensated by the word frequency in the dataset.

3 Methods

3.1 Data Structure

The dataset is made up of questions obtained from the SIDES platform (Intelligent Health Education System) which prepares medical students for the ECNi (French National Computerized Classifying Tests).

The small subset of multiple choice questions, has been annotated by several medical professors referred as experts. On one hand, we have the expert-dataset containing 104 questions labeled through level 1 (to remember), level 2 (to understand) or level 3 (to apply). On the other hand, there is 16893 unlabeled questions.

The questions should thus be all be labeled in accordance to the Bloom's taxonomy.

Everything was written in French.

3.2 Choosing the pretrained model

Firstly, we use pretrained model on questions of medical order. They will be classified based upon the standard Bloom's Taxonomy : To do so, we will retrieve the questions and use the labeled dataset with pretrained models to test upon the unlabeled data, in order to make predictions of these data. The labels chosen will be level 1 for remembering, level 2 for understanding, and level 3 for applying. We compared the "camembert-base" model with other pretrained model on the SBERT sandbox. This includes the following pretrained from Huggingface models called :

- camembert/camembert- ...
  - ··· -large
  - ··· -base-wikipedia-4gb
  - ··· -base-oscar-4gb
  - ··· -base-ccnet
  - ··· -base-ccnet-4gb

camembert/camembert-base-wikipedia-4gb will be shortened as wikipedia4gb in the following notes.
https://huggingface.co/camembert-base contains all the models mentioned above.

3.3 Pattern-recognition in levels

We will count the word in each levels by measuring frequencies inside the questions' csv (with 6 files for each level separately), with the expert-labeled file (discarding levels hierarchy)

3.4 Concatenating questions and answers

After realizing a thorough reading of each question and its answers of the labeled dataset. We took notice of how the vast majority of experts actually meticulously read all the answers. They made their own conclusion on which correct level was needed, and more often than not ; they focused on the answers rather than its question to do so. Therefore, by taking into consideration that most of them annotate levels by answers and consciously take them into account. We decided to add these answers to our model to make it more adequate; we did so by concatenating them to the questions, creating an enriched model to train upon. From now on, we will refer to it as : entire_phrase

3.5 Replacement of digital values with a constant

Because of how frequent some words were in some models we had to take into consideration that the new word replacing those digits was always going to add more weight to each words akin to this newly picked constant. Thus, we replaced every digit by a constant value to see its outcome on different runs. Some of those tests consisted in choosing the unique word : des, plusieurs, nombreux. We made a replacement with blank (a deletion), or with another digit like : 0.
Those changes were made regardless of the sentence's level. Doing it for both the labeled and unlabeled dataset.

3.6 Tf-idf

After noticing that each levels' recurring words were mainly attributed to the domain of several medical fields, we looked at the word's importance in each separate file (regardless of which levels it belonged to). Hence, we retrieve the tf-idf weights in the labeled dataset, and compare the words' weights with the values of diverse digits' weights.

3.7 ROC curve

We compare two pretrained model : wikipedia4gb and oscar4gb, together with their ROC curve, on the CamemBERT sandbox.

4 Experiments

4.1 Experiments dataset

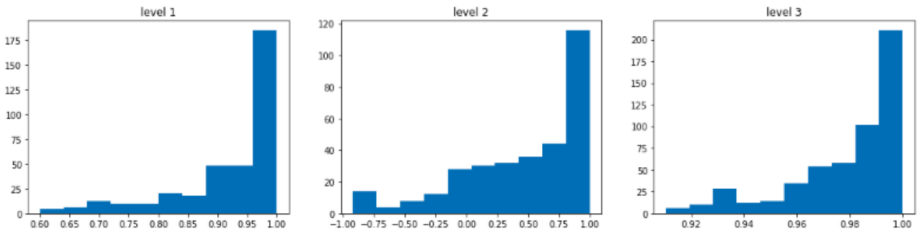

Fig. 2. Barplots of the similarity cosine values with the model : entire_phrase wiki

Heatmap Barplots of cosine similarity We are evaluating the barplot level by level. We will be especially focusing on the first x : the minimum ; that we will attempt to maximize especially for the level 3 and more importantly for the level 2 .

Comparing with heatmap and barplot, the cosine similarity values are especially good in this model with the level 3, This is translated visibly on the heatmap thanks to a color gradient, however these values are in different scales from one another, making the colors and the heights hardly comparable in terms of level performance. That's why the barplots display is more suited for the comparing different, especailly on the x-axis.

Finally, we can only rely on their minimum values of their cosine similarity, and we will discard the heatmap in further comparisons, because a way to match all colors in several heatmaps is yet to be found.

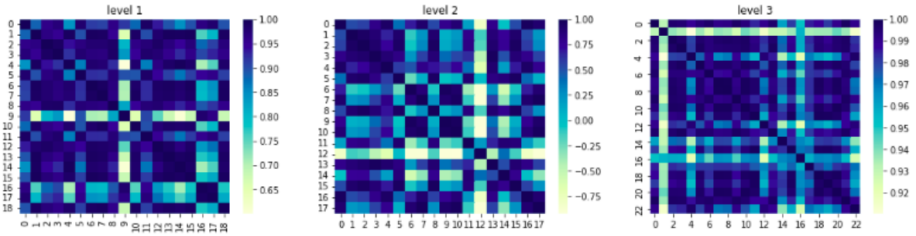

Fig. 3. Heatmap of the similarity cosine values for the wikipedia_4gb model with entire_phrase wikipedia_4gb model.

4.2 Experiments settings

Pattern Recognition We compare the patterns recognized level's by level from the csv files in the labeled datasets, to `IQ_rbt_all_combined` csv which contain all levels regrouped together

	levels1	levels2	levels3	IQ_with_predictions
ans	2454	10123	13918	58
que	7291	7226	9427	99
comme	68	388	1115	2
quand	0	67	102	3
comment	6	27	32	0
pourquoi	1	19	16	0
concernant	2181	1760	1744	13
radio	118	679	3253	0
neuro	102	578	336	0
rhum	49	138	144	0
pédia	163	894	1288	2
médecine	10	54	54	0
médecin	34	1215	521	3
ophtal	15	24	21	0
allerg	1162	176	171	4
imagerie	42	119	3120	1

Fig. 4. Word count frequency among numerous dataset.

Occurrences of the most common adverbs used to make questions in French: Que(quelles,quel,quels, quels), comment comme(comment, comme), quand, pourquoi
For the adverb "que", its number of occurrences is almost the same for each level, but we have about 30
For the adverb "comment", we observe occurrences in the three levels, but it is approximately 5 times more present in level 2 and 15 times more for level 3. Level 3 has almost three times as many occurrences as level 2.
For the adverb "comment", we observe occurrences in the three levels, but it is about 4 times more present in levels 2 and 3.
For the adverb "pourquoi", it is observed that it is present in the same quantity in level 2 and 3 and almost absent in level 1.
Occurrences of the most represented medical fields: radio (radiography, radiology, radiography...), imaging, neuro(neurology, neurography...), rhum, pedia(pediatrics..), allerg(allergology...), ophtal(ophthalo...),medecin,medicine
Radio and imagerie are mainly present at level 3.
Allerg is mainly present at level 1.
Ophtal,, rhum , pedia are mostly present in level 2 and 3 but in general slightly more in level 3

4.3 Model accuracy

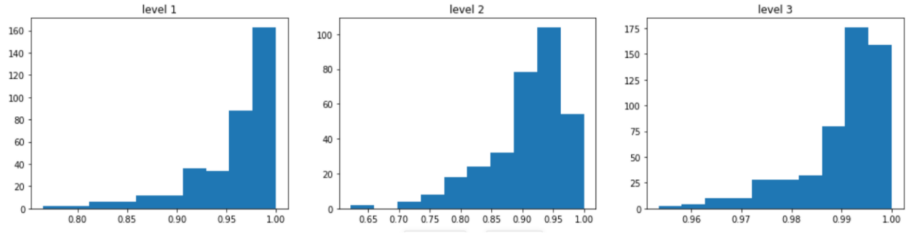

Fig. 5. Barplots of similarity values for each level's with question's embeddings with intact wikipedia_4gb model.

The figures 5 and 6 represents respectively the intact_wikipedia4gb and the replaced_digits_with_plusieurs_wikipedia4gb models displaying the cosine similarity values of each questions' within its own levels.

We are evaluating which one of the 2 plots; among the plots having the same level, has the highest x leftmost values. This means we will be especially focusing on the first x : the minimum . For example : the level 2 plots illustrate very well that the intact_wikipedia4gb is better than the model replacing figits here, because the min is negative for replaced_with_plusieurs, while it displays a positive min for intact_wikipedia4gb one.

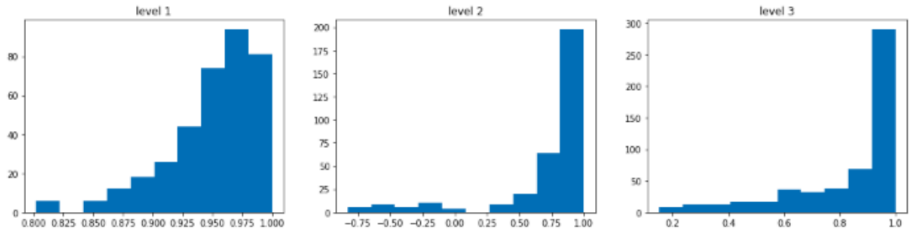

Fig. 6. Barplots of similarity values for each level's with question's embeddings replaced_digits_with_plusieurs wikipedia_4gb model.

This comparison translates that the Wikipedia4gb pretrained model, has higher values across all levels compared to the replaced_digits model, they are most accurate : which means, its questions are more similarly packed up together.

4.4 ROC Curve

In order to compare some pretrained model performance on other models, we will look at the the ROC curves with the CamemBERT sandbox. First of all, the AUROC values is almost the same for all of the levels, making it hard to draw any conclusions; so let's look at the ROC instead :

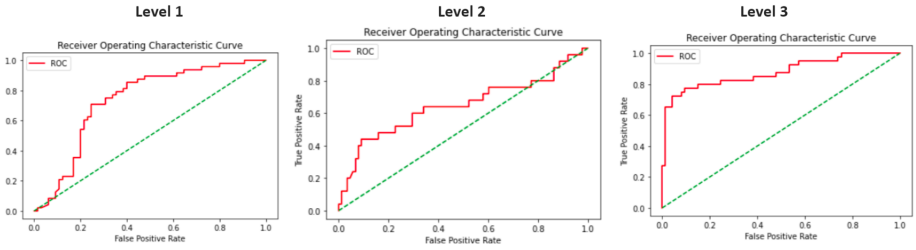

Fig. 7. ROC curves for each level's with CamemBERT sandbox on the wikipedia_4gb model.

The ROC curve of the wikipedia model for level 1 is always above the random prediction (dotted lines), which means that it presents good measures for its predictive accuracy.
The ROC curve for level 3 is also mainly above the random prediction (dotted lines), meaning that it displays good accuracy for its predictions.
Unfortunately, the ROC curve for level 2 is showing some values below the random prediction (dotted lines), even though it is only for a few of its values, it will demonstrate bad measures for its predictions. Therefore, we should consider this level2 as the one possessing the most margins of improvement in future runs.

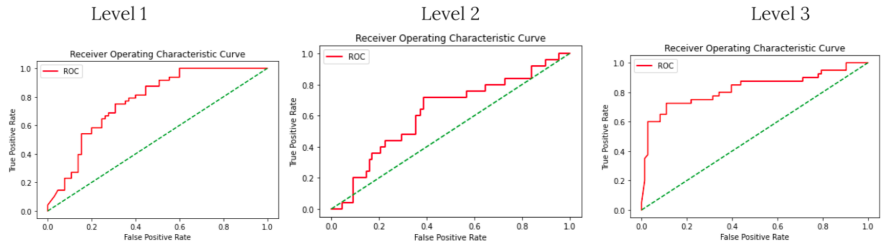

Fig. 8. ROC curves for each level's with CamemBERT sandbox with the model oscar_4gb.

With the oscar4gb pretrained model, we observe that the roc curve is always above the threshold, so we conclude that the level 1 is well classified. Level 2's plot shows rates going under the random prediction (dotted lines). This highlights again the importance of this level 2 since it is the most unstable across all levels. The level 3 shows rates going always above the random prediction, which makes it well classified.

4.5 Comparisons of models

To compare the effectiveness of the different models that we tested, we computed different pretrained models. Table 9 presents all the minimum cosine similarities within each questions' embeddings that we obtain for each model. It shows that Wikipedia4gb is the best in level 3 and 2. Moreover, large has higher results in level 1. So Wikipedia4Gb is better for questions around understanding and applying. With the model : Large ; the remembering questions fares slightly better than Wikipedia4gb pretrained model.
However, the overall level-average criteria performs better with the pretrained Model : Wikipedia4Gb.

Pretrained models	min level 1	min level 2	min level 3	all levels averaged	
Model : camembert-base	**0.3**	-0.1	-0.24	0	
Model : oscar_4gb	**-0.82**	**-0.94**	**0.95**	-0.27	
Model : ccnet4gb	-0.61	-0.76	**0.93**	-0.15	
Model : large	**0.88**	-0.29	-0.08	0.17	
Models : Wikipedia4GB	0.76	**0.62**	**0.95**	**0.78**	Best of all
entire phrase (questions & answers)	0.59	-0.02	-0.82	-0.08	
entire_phrase without plurals, nor T/F	**0.56**	0.55	-0.51	0.2	
deleting_digits	**0.65**	-0.79	**-0.7**	-0.28	
replaced digits with : quelques	**0.93**	-0.55	0.55	0.31	
replaced digits with : des	**0.81**	-0.64	0.57	0.25	
replaced digits with : plusieurs	**0.8**	-0.83	0.15	0.04	
replaced digits with : nombreux	0.7	-0.02	**0.92**	0.53	
replaced digits with : 0	-0.68	-0.35	**0.93**	-0.03	

blue bold : maximum among 3 levels
highlight in yellow : maximum within level's column
Lowest : within level's columns

Pretrained model comparison
Wikipedia4gb model is used for those highlights

Fig. 9. Minimums of cosine similarity within each questions' embeddings of

Entire phrase stands for sticking the answers to the questions The plural forms like in : la(les) ; and the T/F (True)/(False), were erased in order to see if those made any difference when deleting all parentheses.

In the all_levels_averaged values' column : the Wikipedia4GB pretrained model (with questions-only) is the best overall.

Firstly, we compare the levels with each change made to the sentence : when we compare the sentence, when we combine the question and answers : we realize that it is showing worse results than with the questions alone. We focused on the numerical values, which are mainly present in level 2 and 3 's We replaced those digits by fixed words, to do so, we made several runs with different words (by replacing decreasingly less commonly used equivalents), or 0, or simply deleting them.*
However, we came to the conclusion that even with a rarely occurring word such as "nombreux", we still can't come close to our golden standard (intact wikipedia4gb) on level 2, (the other levels are better). Thus, we think that keeping this model is the wiser approach (so far..)

Pretrained models	avg level 1	avg level 2	avg level 3	all levels averaged	
Model : camembert-base	0.88	0.64	0.68	0.73	
Model : oscar_4gb	0.65	0.3	0.99	0.65	
Model : ccnet4gb	0.62	0.48	0.97	0.69	
Model : large	0.96	0.75	0.65	0.79	
Models : Wikipedia4GB	0.95	0.9	0.99	0.95	Best of all
entire phrase (questions & answers)	0.85	0.76	0.67	0.76	
entire_phrase without plurals, nor T/F	0.85	0.87	0.58	0.77	
deleting_digits	0.96	0.44	0.71	0.7	
replaced digits with : quelques	0.98	0.57	0.89	0.81	
replaced digits with : des	0.93	0.52	0.93	0.79	
replaced digits with : plusieurs	0.95	0.69	0.83	0.82	
replaced digits with : nombreux	0.96	0.8	0.98	0.91	
replaced digits with : 0	0.68	0.78	0.98	0.81	

Pretrained model comparison
Wikipedia4gb pretrained for those highlights as well

Fig. 10. Averages of the cosine similarity within each questions' levels

We observe the same remarks, with the exception of the entire_phrase model, meaning we shouldnt only focus only on the minimums when trying to translate the predictions produced. Since their cosine similarity values can actually be pretty low (for their minimum), while keeping a good average values in the end. So, adding more substance to the sentence's shows good results for level 2, but it comes at the detriment of level 3, which is an undesired outcome that has yet to be fixed.

Even if at times, we achieve better performance in level 1, it is at the cost of worsening level 2 and 3. This is due to the fact that some replacements drastically change each sentence's meaning, which is why a more case-by-case factoring of those newly generated artifacts is required.

We conclude that the Wikipedia4GB is the best regarding level 2 and level 3, and the second best for level 1. Then, we iterate over many ideas using it by default, to see how we can improve this model farther.

We observe that all of them have worst performance than the one we have with our original golden standard : questions-only Wikipedia4GB

5 Discussion

In the summary table, our results shows that level 2 of Bloom's taxonomy is the hardest level to classify. When we try to improve our model, the level 2 criteria is the one showing us if the modification is significant. Some modifications like the replacement of digital values can improve the level 1 just a little at the cost of a drastically lowering the accuracy of level 2.

Also, we started working on rule-based matching and dependency tree of syntactic relation generated with spaCy, by illustrating how the loss of digits breaks pre-established links, and how it would change the sentence's meaning. This could explain how the deleting_digits model have lower performance than replacement models. Because these models would still simulates the preexisting structure, preventing those structures to be broken.

5.1 Tokens' importance weights

FALSE	0.9454775637
jour	0.6303183758
2	0.6303183758
7	0.3151591879
21	0.3151591879
6	0.3151591879
semaine	0.3151591879
minute	0.3151591879
heure	0.3151591879
1	0.3151591879
4	0.3151591879
TRUE	0.3151591879

Fig. 11. extract of tf-idf-weights from the labeled dataset

As we can see, numbers have weights just like words, and some have greater weight than others. This means that they shouldn't all be replaced by a unique value. Instead, we should consider them by groups of digits, according to the importance of their weight.

6 Conclusion

In the works, we experimented with various parameters to observed how it influenced how close each levels were within among themselves in their own embeddings, we can conclude that the Wikipedia4G has achieved the highest results of all the pre-trained model in varied aspects. We try to finetune the model with pattern recognition of each level. We also try to directly modify the questions to see if the model could be improved. As a consequence, concatenating question with their answer actually made the model worst. Digits values replacements are not making the model better than before, but the resulting outcomes tend to lean towards exploring the use of rarer (that is to say : with more sustained language) when we have to replace by constant words

We haven't used the tf-idf weights with the kmeans, but it could be explored if we want to rate the accuracy of our model in the long run.

7 Acknowledgements

We would like to thank Anna Bobasheva and Catherine Faron-Zucker for their technical and research support. We also want to thank Oscar Rodríguez Rocha for all the help in his research works on extracting learning objectives from training questions in the medical field.

[2]

References

1. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs] (May 2019), http://arxiv.org/abs/1810.04805, arXiv: 1810.04805
2. Kudo, T., Richardson, J.: SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. arXiv:1808.06226 [cs] (Aug 2018), http://arxiv.org/abs/1808.06226, arXiv: 1808.06226
3. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692 [cs] (Jul 2019), http://arxiv.org/abs/1907.11692, arXiv: 1907.11692
4. Martin, L., Muller, B., Suárez, P.J.O., Dupont, Y., Romary, L., de la Clergerie, V., Seddah, D., Sagot, B.: CamemBERT: a Tasty French Language Model. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics pp. 7203–7219 (2020). https://doi.org/10.18653/v1/2020.acl-main.645, http://arxiv.org/abs/1911.03894, arXiv: 1911.03894

5. Palombi, O., Jouanot, F., Nziengam, N., Omidvar-Tehrani, B., Rousset, M.C., Sanchez, A.: OntoSIDES: Ontology-based student progress monitoring on the national evaluation system of French Medical Schools. Artificial Intelligence in Medicine **96**, 59–67 (May 2019). https://doi.org/10.1016/j.artmed.2019.03.006, https://linkinghub.elsevier.com/retrieve/pii/S0933365718301295
6. Piao, G.: Scholarly Text Classification with Sentence BERT and Entity Embeddings. In: Gupta, M., Ramakrishnan, G. (eds.) Trends and Applications in Knowledge Discovery and Data Mining. pp. 79–87. Lecture Notes in Computer Science, Springer International Publishing, Cham (2021). https://doi.org/10.1007/978-3-030-75015-2$_8$
7. Reimers, N., Gurevych, I.: Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. arXiv:1908.10084 [cs] (Aug 2019), http://arxiv.org/abs/1908.10084, arXiv: 1908.10084