ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

Национальный исследовательский университет «Высшая школа экономики»

Факультет гуманитарных наук

Образовательная программа «Фундаментальная и компьютерная лингвистика»

КУРСОВАЯ РАБОТА

На тему «Лингвистическая интерпретация моделей: визуализация больших данных»

Тема на английском “Linguistic Interpretation of NLP models: big data visualization”

Студентка 2 курса
группы № 212

Аванесян Алина
Армавиковна

Научный руководитель
Сериков Олег Алексеевич
Научный сотрудник
Института Искусственного
Интеллекта,
приглашённый
преподаватель Школы
Лингвистики НИУ ВШЭ

Москва, 2023 г.

Contents

1. Introduction

Probing experiments are aimed to estimate the efficiency of the language model. Probing is a method for analyzing the capabilities of models with the help of designed tasks. It is based on the external classifier model ("probe") that is trained to predict linguistic properties from the internal model representations (Hewitt and Liang, 2019; Ferreira et al., 2021). This approach allows to reveal the weaknesses of the model on the different languages and their levels, and, consequently, helps improve the accuracy of the neural model. For instance, it can be needed to know to what extent the representations generated by the model take into account morphosyntactic features, and is the model able to determine grammatical categories (Voice, VerbAspect, and etc.). By conducting probing experiments, the prediction of the specific linguistic features based on the hidden representations of a model can be obtained. So, depending on the received accuracy scores, it can be possible to postulate the model successfully or unsuccessfully copes with tasks of a certain kind.

Multilingual models positioned as models that are capable of processing multiple languages. However, training such models to show good results can be quite challenging because it demands combining differing techniques, thanks to which the linguistic features of the target languages (specifics of the grammar rules, word order, etc.) will be taken into account. Although probing helps to check how well the model can understand and handle various linguistic materials, the interpretation of its results is accompanied by some difficulties. To make meaningful conclusions, it is necessary to be able to find patterns in a huge array of data. It cannot be done when the user tries to estimate the model relying on the huge number of json-files (this format is used to store the probing results) or on the previous means of visualization that usually served for monolingual probing. For qualitative analysis, numerical data should be presented in the form of understandable graphs that can be interpreted. (Serikov et al., 2022) tells about the designed framework aimed to conduct probing experiments on a massive number of languages where visualization occupies a separate place.

Making a brief excursion into the GUI-assisted framework described in (Serikov et al., 2022), it is worth noting that its structure provides by the tool that perform following tasks step-by-step:

1) pre-processing data for probing

2) probing engine

3) visualization
At first, the probing framework gets CONLLU files (the source of these treebanks is the Universal Dependencies data (de Marneffe et al., 2021)) or a directory to such files and gives us files in SentEval format because probing tasks are mostly based on the SentEval methodology (Conneau and Kiela, 2018). Then these files are sent to the probing engine that consists of encoders that create embeddings for each element of the input, classifiers and metrics. Classification tasks are performed here based on the obtained grammatical categories. As classifiers, classification linear and non-linear models are used and, as metrics, accuracy score and weighted F1 score are chosen. The last step is aimed to facilitate the analysis with the help of the reflecting values on the graphs. They should transmit all the necessary stuff and not be overloaded at the same time.

Therefore, there is a need to visualize a massive data, a large quantity of raw numbers, to understand the pros and cons of the considered models. This paper suggests a dashboard that can be useful for simplifying the perception of big data resulting from multilingual probing.

The **purpose** of this work is not only to develop a probing visualization tool, but also with its help to conduct the analysis of several language models. The probing results, which will be visualized and discussed further, were received according to the probing methodology described above in the work (Serikov et al., 2022).

Such work is motivated by the following **research tasks**:

1. Based on the analysis of available visualization tools, highlighting their advantages and drawbacks, to select the most suitable graphs for displaying different aspects of probing results.
2. With the help of the designed dashboard, to deduce universals about each model: to identify categories, languages, and language families that are recognized poorly/well and to reveal the relationship between the result and these aspects parameters (to establish how similar the accuracy scores of languages with the same genealogical information are). We expect that the languages of some language families have comparable results.
3. To compare models by searching for common patterns and examining the average values of the model layers. Perhaps, all four models need to learn how to better define a particular category, a particular language, etc.). In general, it is about checking if they have some intersections: how similar are their probing results for the same languages and language families, if common strong and weak categories can be identified.

4. To check whether it is possible to detect layers with high and low accuracy that will be equally bad or good for all languages of a given language family within one or all categories.
5. To check whether there is a correlation between whether a category characterizes a nominal or verbal grammatical category.
6. To check if there is a correlation between the results and the structure of the pretraining dataset.
7. To check whether there is a correlation between the result and the volume of the training dataset prepared for probing.

In this paper, such language models as BLOOM (176B) , BLOOM-1B7 (le Scao et al., 2023), mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020) will be discussed. These models are referred to the transformers, the main feature of which is the attention mechanism that provides by the possibility of parallel processing what makes such a mechanism more efficient than traditional RNN's (recurrent neural networks) and CNN's (convolutional neural networks), which process inputs sequentially.

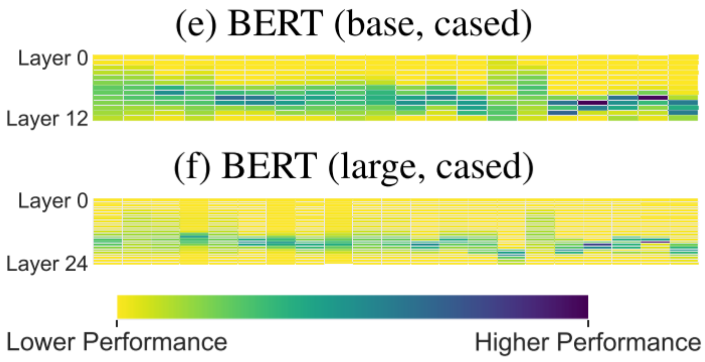
Transformers were first introduced in (Vaswani et al., 2017) where an innovative approach to processing sequential data was proposed by the authors. Thanks to the multi-head attention mechanism underlying this architecture, such models can cope with the long sequences more efficiently. The input is split into heads where each head has his own representation because they focus on different aspects. A set of weights is computed that indicate how much attention should be paid to each token in the sequence. This feature allows to capture more details of the discourse that makes the results of the models' work better because the relationships between words are taken into account. Therefore, multi-head attention improves the quality of the output because this mechanism allows for many nuances. Thus, models with a transformer architecture are of particular interest.

2. Related work

It turned out to be a difficult task to search for literature that would highlight the visualization of probing results. In many works, visualization is used as an intermediate link to analyze the model, among them such articles as (Rogers et al., 2020), (Durrani et al., 2020) and (Bhattacharya et al., 2022).

(Rogers et al., 2020) discusses BERT, provides an overview of its linguistic knowledge and technical aspects of the model. The authors use **heatmap** (see Figure 1) to visualize how various probing tasks work on 12 layers for BERT (base, cased) and on 24 layers for BERT (large, cased). The heatmap is accompanied by a color scale, where darker colors correspond to Higher Performance, and light colors correspond to Lower Performance. This method of visualization seems to be quite effective, because the user can visually assess how many light and dark areas were revealed as a result of probing. This creates a general impression, as well as a layer-by-layer comparison becomes available.

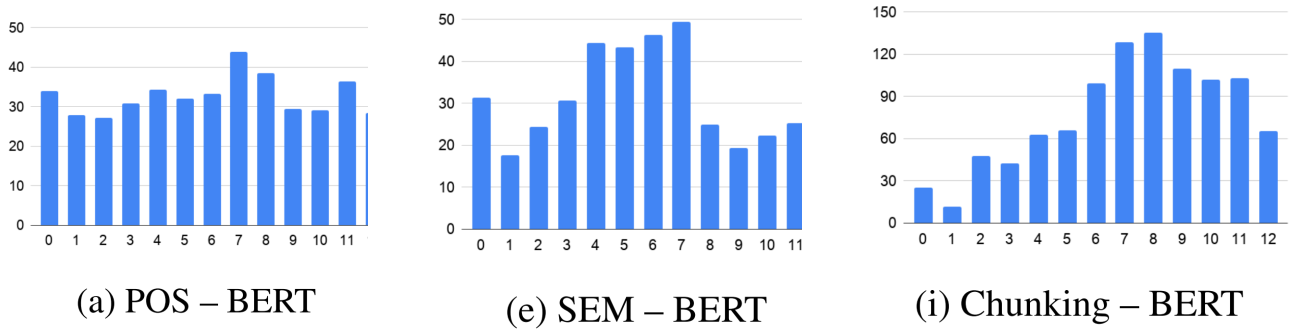
Figure 1: BERT layer transferability from Rogers et al. (2020)


(e) BERT (base, cased)

(f) BERT (large, cased)

In (Durrani et al., 2020), **histograms** (see Figure 2) can be noticed where each column reflects the value for each layer. Such visualization seems too cumbersome and inconvenient. If there is a need to examine different probing tasks on the same graph, there are two ways: consider each task on a separate schedule (as in Figure 2) or add multiple plots for every task. The first one makes it difficult to compare results by various categories because it creates a large amount of graphs. In the second case, the columns corresponding to the certain categories overlap each other. This may not interfere with the perception of information as long as the number of parameters to be considered does not exceed three. For this reason, **linear charts** seem to be more appropriate for such data because the intersection of lines does not create serious difficulties in interpretation, and individual trends can be hidden.

Figure 2: Distribution of neurons in different layers for each task (X-axis = Layer number, Y-axis = Number of neurons selected from this layer) from (Durrani et al., 2020)


(a) POS – BERT

(e) SEM – BERT

(i) Chunking – BERT

Authors of (Bhattacharya et al., 2022) demonstrate the MLP dev accuracy of the tasks performed by BERT and use for this goal **error bars** (see Figure 3) that indicate the range of values for a certain measurement and errors that report the anomalies in trend behavior. Since there are multiple trends on one chart (for TF-IDF, Hadamard, Pooler, Mean and CLS), it is allowed to reveal that the 5th layer represents a great complexity for CLS representation, while for the others there is no such serious drops in accuracy relative to their values. This paper also appeals to the **heatmap** (see Figure 4) for demonstration results of TF-IDF baseline, BERT and GPT-2 by the BLiMP tasks referring to morphology and syntax-semantics.

Figure 3: MLP dev accuracy for *determiner noun agreement irregular 1* task of BLiMP benchmark for BERT representation across layers from (Bhattacharya et al., 2022)

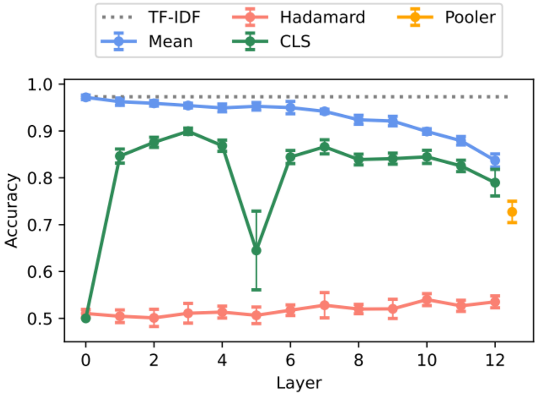
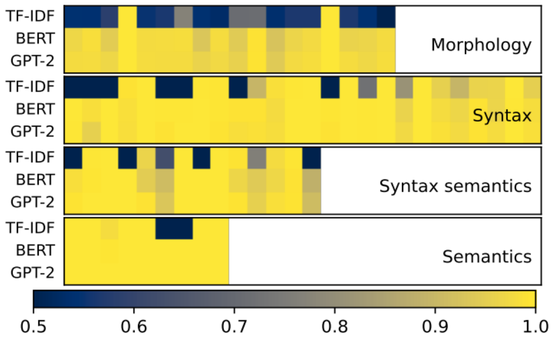


Figure 4: Accuracy on various BLiMP tasks from (Bhattacharya et al., 2022)



Thus, visualization of probing results has not been a central task of researchers until now. The visualization tools proposed earlier were necessary for interpreting the data and solving the questions posed, were an auxiliary but significant part of the study, without which it would be difficult to identify patterns or correlations between the variables. In the considered and other papers **line graphs, heatmaps, bar charts, error bars, scatter plots** and other types of figures were used. In the next section, a dashboard will be proposed, which includes the most successful solutions for visualizing probing experiments.

3. Methods

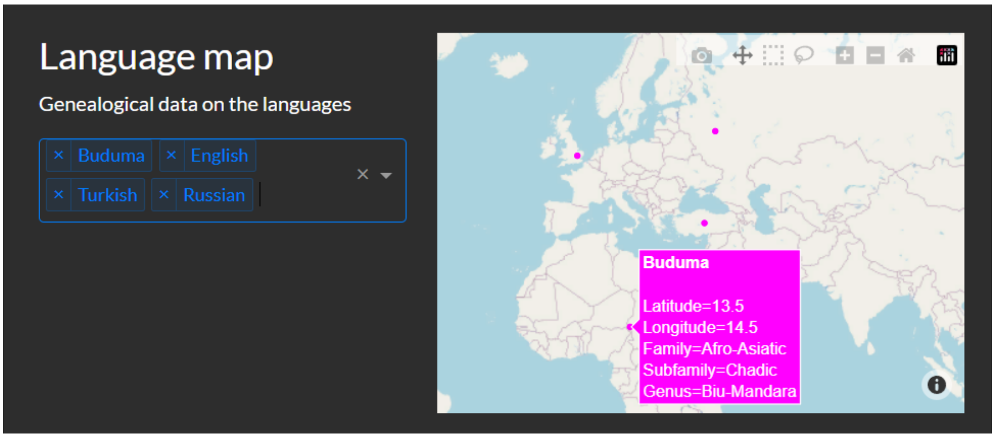
The presented dashboard is based on the Plotly architecture – Dash, and Dash in turn is built on the Plotly.js, React and Flask. Compared to Matplotlib and Seaborn graphics, graphs from Plotly are more modern both in appearance and functionality. Plotly is more economical and organic in terms of the amount of code needed to create complex graphs. For while

Matplotlib is more demanding, and the interface it provides is rather outdated and low-level despite it shares a vast collection of graphs. Although Seaborn as built on the Matplotlib represents a more contemporary library which manifests itself in the requirements for fewer lines of code and a more attractive design, it still gives way to Matplotlib having a collection with a smaller variety of graphs, it is limited. Consequently, Plotly has competitive advantages. Plotly offers easy interaction graphs and a web application that is convenient for integration with other tools. Tableau is also a fairly spread and powerful data visualization tool. However, Plotly can be integrated with programming languages, while Tableau offers a possibility of creating a dashboard without any coding skills, it creates graphs by the uploaded data. Since our work is based on Python, Tableau was not chosen as inconsistent with our purposes. Moreover, Plotly suggests more flexibility in terms of customization.

As for the functionality of the suggested dashboard, it can receive as input the results of probing of various models, so there is a possibility to switch between models in the dashboard using dropdown. Accordingly, the user receives a set of graphs for the selected model, which is accompanied by a manual (see Appendix A.2) describing all the graphs, which helps to better understand the abilities and purposes of each added analytics tool.

To identify the regularities in how the model copes with the categories of certain languages, it is important to have a source of information about the genealogical data on the language, such as family, subfamily and genus. The **scatter mapbox** (see Figure 5) solves this problem by showing the distribution area of the chosen language(s) and the corresponding data.

Figure 5: Language map



Language map

Genealogical data on the languages

× Buduma × English × Turkish × Russian

Buduma

Latitude=13.5
Longitude=14.5
Family=Afro-Asiatic
Subfamily=Chadic
Genus=Biu-Mandara

Secondly, it is necessary to obtain average indicators on how accurately the model copes with certain categories. Displaying such data seems convenient in the form of a

horizontal bar chart (see Figure 6 in Appendix A.1), where the categories are arranged in descending order of their average values. Thus, it will be possible to make an initial assessment. Average values are also offered on two charts, which are represented by a combination of **bar chart** and **boxplot**. The first one (see Figure 7 in Appendix A.1) shows the average values, as well as the spread of average values for each layer. The second (see Figure 8 in Appendix A.1) graph reflects the average values of the languages arranged in descending order of average accuracy.

Nevertheless, deep statistical analysis cannot be carried out if the language composition for each category is unknown: it is necessary to know in which and how many languages this grammatical category has been tested to compare variables correctly. To do this, the *"Average values for each "category-language" pair"* graph (see Figure 9 in Appendix A.1) was built, which is presented as a **heatmap**. Here, on the X–axis, all languages that have been probed are clearly shown, on the Y-axis, are task categories, and at the intersection are arithmetic averages (the value in each layer/the number of layers). This graph is necessary because it is impossible to take into account the values of the previous graph without knowing whether it is relevant to compare categories with different numbers of languages. For instance, the AdvType category is represented by four languages, and Voice, a more common category, is represented by thirty–three, so the average value of AdvType is formed from a small number of languages in comparison with the Voice, and Voice may have variations due to linguistic diversity. Using heatmap to display such pairs in the form of color-coded cells is reasonable because it facilitates the task of detecting the areas with low values by darker colors among many variables.

This dashboard attempts to establish generalizations based on language families. In this regard, the graph *"Average values of language families"* (see Figure 10 in Appendix A.1) is given, which is represented by a **scatter plot**. The idea is that the dashboard analyzes all the languages that it receives as input, determining whether each of them belongs to a language family (see Table 1 in Appendix B). The framework counts the number of languages of each language family. This indicator affects the size of the dot on the scatter plot. So, the point corresponding to the Indo-European family is likely to be more massive than the rest, since this language family is rich in its composition, which makes it more likely that the language will be assigned to this category. This graph is accompanied by a sidebar with statistical data on how many files a particular language family is represented by.

It must be said that the attempt to consider languages as part of a larger group continues in the dashboard block *"Statistics on language families"*.

It starts with a **dropdown** with a list of all the language families found in the downloaded files, which opens access to the graphs for the selected option. Next comes a part with a brief overview of the language family: the number of languages in the language family, the most poorly and accurately recognized category among the languages of this language family.

It seems important to show the structure of the language family, so **treemap** (see Figure 11 in Appendix A.1) was found to be convenient in perception, where the size of each plate corresponding to the language depends on the number of files in which this language is represented, and the exact number of files is available when hovering over the plate. The next graph of this block is **boxplot** (see Figure 12 in Appendix A.1). It reflects the distribution of the average values for each category. The boxplot is accompanied by a **switch** *"Show languages"*, which, when clicked, adds points to the graph corresponding to the languages with their values. So, we get a combination of a boxplot and a scatter plot. Boxplot shows the range of values and marks the first quartile (Q1) (the median value of the lower half of the data), the third quartile (Q3) (the median value of the upper half of the data), the minimum, maximum value and median. Using these indicators, it is possible not only to identify categories with low accuracy, but also to understand the relationships between languages within a category (which of them demonstrate high accuracy, which are close to the median of the category under consideration, and which are outliers). Thus, it becomes clear which languages can be grouped by values, because they produce a similar result, and which are on the periphery.

When preliminary conclusions are made about how similar some languages are to others in terms of average values, there is a need to compare their exact values. For that purpose, the following **three columns of linear graphs** (see Figure 13 in Appendix A.1) serve: the most similar trends by category, the most dissimilar trends by category, Comparison is impossible, the number of languages is too small. Each grammatical category is represented by certain languages with values on each layer, that is, each language within one category can be shown as a trend on a linear graph. Then we need a mechanism that can compare these trends with each other and group them. Comparing each trend with all available ones is a rather laborious process. Therefore, a comparison can be made with a certain **pattern line** – the median trend. The pattern line is calculated as follows: in each layer, the median is selected among the values of all languages in this category. All these medians make up the Y-axis coordinates for the pattern line, while X-axis is about the layers. As a measure of central location, it is the median that is chosen, and not the average value, since the sample by layers can be very heterogeneous, and does not always represent a normal distribution. For example, the Degree category includes completely different languages, in the first layer more than 20 languages have an F1-score close to 0.8-0.9, but Basque gives a value of 0.133, so the average value does not take into account the asymmetry of the data, while the median is not distorted by outliers. Trends are compared by calculating the **Frechet distance** between the pattern line and the trends of all languages. The Frechet distance evaluates the measure of similarity of curves, taking into account the number and order of points along the curves. Based on the results of mathematical calculations, a rating of trends most closely located to the pattern line is compiled. The user in the dropdown selects the number of trends available for display, which he wants to see on one chart. For example, the user chose the number three. For the graphs of the first group, the first three trends are selected in the list sorted by increasing the distance to the pattern line, for the graphs of the second group, the last three are selected, and the dropdown value does not affect the graphs of the third group, since these include those categories that are represented by a small number of trends, which means it is incorrect to compare them. These graphs complete the block dedicated to language families.

As part of the discussion of language families, it is worth mentioning another graph – the **linear graph** *"Average values of language families"* (see Figure 14 in Appendix A.1) with multi select dropdown. It allows you to look at the average values for all languages and categories within the selected language families in each layer. Therefore, using this graph, it is possible to compare the nature of trends (how strong are the variations of values in the layers and whether fluctuations are frequent), find out which family has the lowest values, and also determine weak layers for all or certain language families (to determine whether the drops in certain layers coincide for selected families or is it a special feature).

After analyzing the language families, there is a need for tools for global comparison of categories. The second **heatmap** (see Figure 15 in Appendix A.1) accompanied by a dropdown is designed for this purpose. In the dropdown, a category is selected, and then all the languages in alphabetical order with the selected category appear on the X-axis, and columns corresponding to the layers appear on the Y-axis. This allows users to select individual categories and, thanks to such a comprehensive view of the data, notice what problems arise in specific layers, whether it occurs systematically or is it the specifics of a particular language.

And finally, the two remaining charts, but some of the most remarkable: **line graphs** with two dependent **dropdowns** where the last one is multi select. The first graph is called

“Values of the languages represented in the selected category” (see Figure 16 in Appendix A.1). One category is selected initially in the list, then the contents of the second dropdown dynamically reacts and offers a list of languages with such a category. Such visualization is suitable for plotting multiple lines (here languages) chosen by the condition (here category) on the same graph. It can be easily interpreted because the number and the character of lines is manually configured. The second figure has the title *“Values of the categories of the selected language”* (see Figure 17 in Appendix A.1) and it is inverse to the previous one because at first it suggests choosing the language and then the list of required categories. The idea is the same: there are multiple lines of different colors on the same graph, so target trends can be tracked. However, a significant advantage of this plot is the addition in the form of an informational block about the composition of train, validation and test datasets for the chosen categories. This tool is one of the most useful in this dashboard, because it allows you to take into account the possibility of underfitting or overfitting of models, which may correlate with the results of probing.

Hence, the dashboard includes various types of visualization: numerous line graphs, boxplot, heatmaps, horizontal bar chart, scatter plots, treemap and even the combination of types of graphs. All of them are suitable for specific goals:
– heatmaps are good at showing the relationships between dependent variables due to the colored cells and gradient color scale, so, there is no difference how many variables, it doesn’t interfere with perception, which makes heatmap indispensable for big data representation;
– line graphs let us overlay multiple trends on one figure, where the trends can be independent or can demonstrate intersections;
– boxplots are aimed to reflect the spectrum of values corresponding to the accuracy obtained as a result of probing experiments, and etc.

Many of the charts have dropdowns that make an interface interactive and user friendly. To help beginners in mastering the technology, there is a button that refers to a pop-up window with instructions for use. The features of Plotly based graphics, all charts have such functions as zoom out/in, autoscale, downloading plot as a png and others.

Now that the dashboard is described as a tool for multi-lingual probing, an attempt will be made to interpret the results of probing several models placed on the Hugging Face in the library of Transformers, including Bloom, Bloom-1b7, mBert (cased) and XML-R.

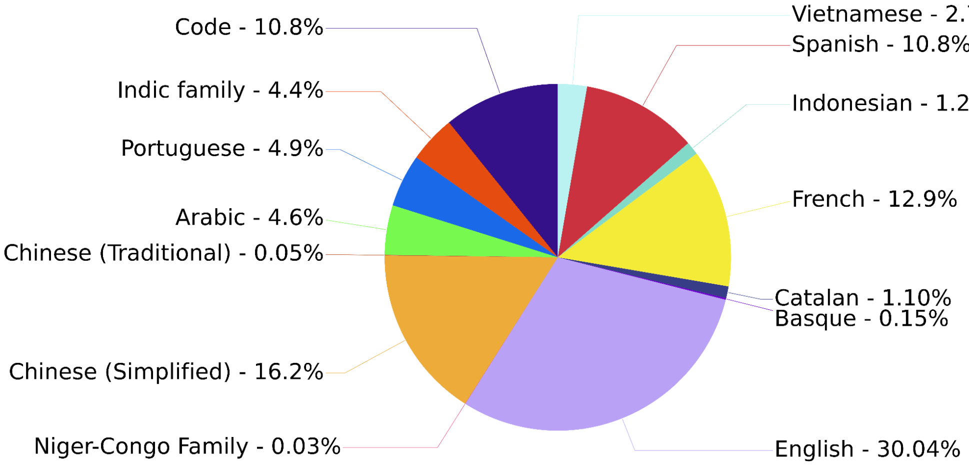
4. Results

It seems quite a difficult task to cover absolutely all categories and languages for analysis. However, in this part, an attempt will be made to analyze some frequency categories. First, each of the four models will be briefly examined, and then a comparison of the results will be given. The models can be grouped by how similar the architecture they have: BLOOM and BLOOM-1 B7, mBERT and XLM-R. This means that, most likely, models within pairs will produce very similar results.

4.1. BLOOM (176B)

BLOOM (BigScience Large Open-science Open-access Multilingual Language Model) is trained on 46 languages and 13 programming languages. As for parameters, this model has 70 layers and 112 attention heads. According to the (Le Scao et al., 2023)[1], the ratio of languages in the pretraining dataset is following (see Figure 18):

Figure 18: The distribution of languages in pretraining data from https://huggingface.co/bigscience/bloom



Consequently, to a greater extent, the model was trained in English and Romance languages, Chinese also accounts for a large share in the pretraining data. As for the Indic family, the vast majority of the examples are devoted to Hindi (0.7 percentage) and Bengali (0.5 percentage).

It is curious to see which categories this model recognizes with high and low accuracy. In order to identify such categories, it is necessary to check as many languages as possible in which this category manifests itself. Table 2*,* located below, presents the categories and their average values, which were tested in at least 10 languages, and the language composition was heterogeneous (a variety of language families were taken). Also, when choosing categories, it was taken into account whether most languages have high/low values (that is, these

[1] https://huggingface.co/bigscience/bloom

categories were selected not just by average values, but also by how uniquely bad or good the category is for all the languages that are being tested). Some of them will be discussed in more detail.

Table 2: Categories that are well and poorly recognized by BLOOM

	Well recognized categories			**Poorly recognized categories**			
	Mood	*Polarity*	*Degree*	*Gender*	*Case*	*Tense*	*PronType*
average f-1 score	0.706	0.668	0.601	0.314	0.363	0.413	0.448

4.1.1. Gender

Gender is the feature of nouns and it is marked in agreement with other parts of speech (such as adjectives, verbs).

The worst situation is with Kazakh Gender (values are close to zero on all 70 layers – 0.039). According to WALS[2], there is no grammatical gender in Kazakh. The absence of gender is confirmed by the data of the grammar of the language (Dotton and Doyle Wagner, 2018).Words differ semantically by gender (therefore, the markup in datasets was most likely based on logical opposition), but grammatically this is not expressed in any way. Basque also has no grammatical gender, but the average accuracy is 0.524. These cases demonstrate the imperfection of samples and methods. In particular, treebanks have inaccuracies, and the model, training on erroneous data, adjusts to them and even gives good results.

Manx is recognized best of all. It belongs to the Indo-European family of the Celtic group (together with Irish, Welsh, Breton, etc.). The other languages of this group have values much lower (±0.5-0.6 and layers with drops to about 0.1). The volumes of train datasets are comparable, so it cannot be said that Gender is well defined throughout the Celtic group.

Significant weaknesses are revealed in many layers of English, where the value drops from 0.751 on the other layers to 0.036. Unlike Manx, English has not two, but three genders (masculine, feminine +neutral). Note that Tamil is one of those languages in which gender is consistently imprecisely recognized (the values range from 0.366 to 0.06), there are also three genders, and only two are marked in the train dataset. In languages such as Icelandic, Albanian, Latin, Upper Sorbian, Marathi, Beja, etc. there are also three grammatical genders and the gender category manifests itself poorly on all layers. Therefore, the languages of the

[2] WALS Online - Feature 30A: Number of Genders

Afro-Asiatic family (Arabic, Assyrian, Coptic, Beja, Amharic) with two grammatical genders have high rates.

It turns out that for most languages, the rule applies: the more genders in the language (the more extensive the grammatical category), the less accuracy of the BLOOM model.

4.1.2. Case

Case marking serves as a means of reflecting the relations between the syntactic elements.

High values are only in those languages in which the case system is not branched (where, for example, there are only 4 cases: nominative, accusative, genitive and vocative), these are languages like English, Irish, Portuguese, worse already with Spanish. In Slavic languages, where there can be 8 cases, the model works poorly. In Turkic languages, which are agglutinative, which simplifies machine processing, the score is still low, and the number of cases is more than 4. Farsi demonstrates great meanings, but there is no diverse case paradigm in this language. It is worth noting that the value of case does not correlate with the number of examples on which the model was trained, so it is possible to postulate the weakness of the model, and not the unrepresentativeness of the sample.

4.1.3. PronType

PronType (pronominal type) is the feature that “typically applies to pronouns, pronominal adjectives (determiners), pronominal numerals (quantifiers) and pronominal adverbs”[3].

The Romanian language consistently shows poor results in all layers (0.031). Although it belongs to the group of Romance languages, it cannot be said that bad values extend to the entire Romance group: Portuguese shows permanently high values on all layers (0.967), Spanish permanently reaches 0.514. In the group of Romance languages, behavior of Romanian is more similar to the French (0.042 on most layers). The similarity of Romanian and French trends is also evident in such a category as Definite. But in Romanian *Gender* it resembles Portuguese.

Arabic gets the highest score – 0.991 on all layers. Arabic is the only one in the Afro-Asiatic family that has the *PronType* category, so it cannot be compared with other languages of its family. The two languages of the Tupian language family are also best recognized: Karo (0.934 on average) and Mbya Guarani (0.706). Other Tupian languages have lower values Guajajara (0.629), Tupinamba (0.211). It can be assumed that the values

[3] https://universaldependencies.org/u/feat/PronType.html

differ due to the volume of the train dataset, but if this correlation works, then only in the Tupian language family. Considering the entire category, no such pattern is revealed.

4.1.4. Mood

Mood is the feature of verb forms expressing the modality.

This category is very well recognized on average for many languages. Some languages (such as Turkish) show unstable trends: high values mixed with drops. Most likely, this is due to the branched system of this category (10 different grammatical moods), there are few examples for each type, so it may have been poorly trained. However, Western Armenian also demonstrates low values, although there are only 4 types of mood in this language (but there are also few examples). There are other languages in which the volume of datasets is small, but the values are permanently high – Spanish, Komi Zyrian, etc. In Classical Chinese, there are a lot of falls despite the adequate size of the train dataset and the share in the pretraining data of the BLOOM model (according to the Figure 11, 0.05%).

Therefore, the model perceives the mood of the following languages poorly: Persian, Turkish, Western Armenian, Yupik. The set of languages with high values is also diverse (among them is Spanish, Hindi, Finnish, Breton, and etc.). Consequently, there is no strong dependence between the results and language families.

4.1.5. Person

Person is the feature of nominal parts of speech (pronouns, determiners). This category is considered because it is impossible to say unequivocally about it that all languages have high or low values, the spread of results within this category is large.

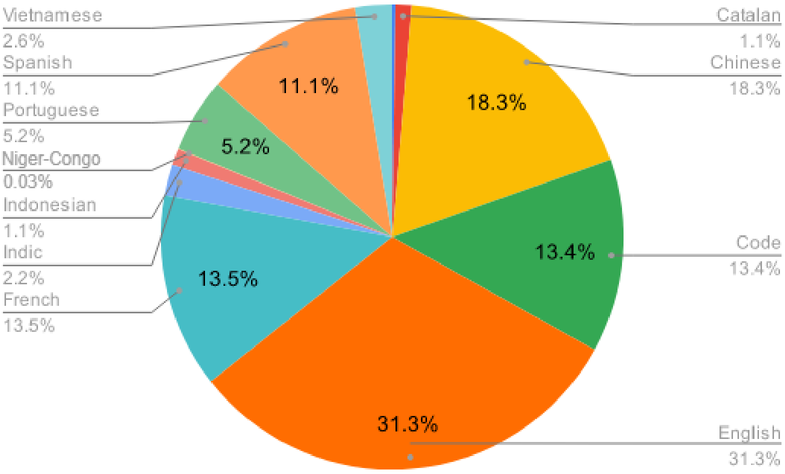
The lowest average value for *Person* is Irish (Celtic group) (according to the boxplot, it is 0.157), followed by Slovak. The Celtic branch consists of Irish, Manx, Breton, Welsh. Their trends differ from the average trends of the Indo-European group (graph “The most dissimilar trends by category, where category is person”). Irish behaves similar to Welsh, their averages are lower in the group (0.157 for Irish and 0.385 for Welsh), and fluctuations are higher. Galician manifests itself best in the group (average score is 0.848), Galician and Manx are more stable, there are few falls.

Urdu is recognized best of all. This language belongs to the Indo-Aryan group (Hindi, Sanskrit, Bengali, etc.). Coptic language has many more examples in the train dataset than Bengali (1151 for Coptic and 40 for Bengali), however, Coptic has values even lower than Bengali (the upper bounds are identical, but the drops in many layers are about 0.2 lower than the Bengali ones). If you check Hindi, you will notice that this language has high *Person* values in all layers, except for three ones. This may be due to the fact that Bengali and Hindi are among the 46 languages in which BLOOM was trained. However, there is no correlation between the pretraining dataset size and the results: Bengali has more weight then Marathi in the pretraining data, but the Bengali has less accuracy. This correlation does not also work with Hindi and Urdu: the percentage of Hindi is higher, the average value is lower.

4.2. BLOOM-1B7

BLOOM-1b7 is trained on 45 languages and 12 programming languages. It has 24 layers and 16 attention heads.

Figure 19: The distribution of languages in the pretraining dataset from https://huggingface.co/bigscience/bloom-1b7



Here, in comparison with BLOOM, the share of French, Portuguese, Spanish, Chinese in the pretraining data is increasing.

Table 3: Categories that are well and poorly recognized by BLOOM-1B7

	Well recognized categories		*Poorly recognized categories*		
	Degree	*Mood*	*Gender*	*Tense*	*PronType*
average f1-score	0.589	0.494	0.332	0.359	0.377

4.2.1. Degree

Within this category, the languages of the Indo-European and Uralic families were probed. For all languages except Erzya, Italian and Komi Zyrian, the values are permanent on all layers.

It cannot be said that the range of values is wide, but Slavic languages show good results: Russian – 0.969 on each layer, Old East Slavic – 0.948, Bulgarian – 0.942, Czech – 0.906. Result of the probing experiments by the Classical Chinese is also high – 0.869. The Italian branch (Latin, Italian, Spanish) of the Indo-European Romance languages family manifests itself worse than English. The most problematic language is Italian: 0.533 alternates with 0.476. The Italian language can be poorly manifested due to the small volume of the train dataset: the number of examples is only 28, while for Spanish there are 2100 examples, for Latin – 847. As for the Slavic group, the Bulgarian has the largest number of examples, but the meaning in Russian is higher. According to the Pearson coefficient, there is a weakly expressed positive correlation between the size of the train dataset and the result (0.4 on average, 0.49 for an Indo-European family).

The Uralic family is less stable than the Indo–European one: in Erzya, the value is 0.791 on 20 layers out of 24, and 0.714 on 4 out of 24, and in Komi Zyrian, 21 layers are stable (0.643), and 3 layers demonstrate a failure of 0.5. Moreover, Erzya has 52 examples in the train dataset, and Komi Zyrian – 30.

Thus, it can be said that Degree in the Ural family is manifested worse than in the Indo-European family. Moreover, the correlation between the accuracy value and the volume of the train dataset works better in Uralic, however, this should be tested in more languages.

4.2.2. Mood

The values on all layers of all languages except two (Classical Chinese and Xibe) are constant and high: the lowest values are in Scottish Gaelic (0.629), and the rest > 0.8 (here Bulgarian, Catalan, English, Latin, Spanish, and etc. – mostly Indo-European languages, but there are also two languages of the Uralic family).

As for Classical Chinese (Sino-Tibetan language family), this language has 8 layers with values close to zero (0.079), the rest have a value of 0.667. The Xibe language (Tungusic family) has 5 layers with a value of 0.101, and the rest – 0.333. Note that the languages of the Sino-Tibetan family are common in East, Southeast and South Asia, and the languages of the Tungusic family are in Siberia and the Far East, in particular, the Xibe language is spoken in northwest China. The “dark” cells (layers with values close to zero) in the two languages under consideration coincide only on layers 9, 13, 14, and the coincidence of layers is not repeated with the same languages in other categories.

Thus, the Mood category in this model recognizes almost all languages well, which is not associated with genealogical data languages.

4.3. mBERT

BERT (Bidirectional Encoder Representation Transformers) Base Multilingual Cased is trained in 104 languages with the largest Wikipedia and has 12 levels and 12 attention sections (Devlin et al., 2019). The total number of parameters of this model is 172M. One of the language collectives, and there is no Tatar language felt in it, which is not present either in the BLOOM model or in the XLM-RoBERTa (more about this model in the subsequent category). Also, the pretraining dataset contains tokens from all Slavic languages: Belarusian, Bosnian, Bulgarian, Polish, Macedonian, Russian, Serbian, Croatian, Czech, Serbo-Croatian, Slovak, Slovenian, Ukrainian.

Interestingly, in this model, the average values are mostly above 0.3.

Table 4: Categories that are well and poorly recognized by mBERT

	Well recognized categories			*Poorly recognized categories*		
	Number	*Person*	*Mood*	*PronType*	*Tense*	*Case*
average f-1 score	0.666	0.646	0.541	0.167	0.343	0.387

4.3.1. Person

It cannot be established that the languages of a certain language family have bad meanings. In this case, rather, there is a correlation with the number of examples in the train dataset – Afro-Asiatic family. Arabic, as part of this language family, does not follow this trend, however, perhaps its high value is due to the fact that its marked-up data is included in pretraining dataset of the models. The mentioned correlation does not work for Slavic languages (see Table 6, column 1).

Table 5: The first column corresponds to Afro-Asiatic Family, the second one reflects the data on the Indo-Aryan group of the Indo-European family

Language	**Number of examples**	**f1-score on each layer**	**Language**	**Number of examples**	**f1-score on each layer**
Assyrian	41	0.45	*Bengali*	40	0.52//0.229

Coptic	307	0.489	*Sanskrit*	138	0.663
Amharic	848	0.688			
Arabic	779	0.954			

Table 6: The first column corresponds to the Slavic group of the Indo-European family, the second one reflects the data on the Romance group of the Indo-European family

Language	**Number of examples**	**f1-score on each layer**	**Language**	**Number of examples**	**f1-score on each layer**
Croatian	876	0.45	*French*	791	0.94
Serbian	395	0.882	*Portuguese*	791	0.94
Slovak	296	0.96	*Italian*	795	0.94
			Spanish	795	0.94

To check this hypothesis, since the sample of the average values for the *Person* category for all the presented languages and the sample including the number of examples in the train dataset for the corresponding languages have a normal distribution (which was checked using the Shapiro-Wilk test), we can calculate the Pearson coefficient. For all languages, the correlation value is +0.39 and we can calculate it for several language families:

– Afro-Asiatic family: +0.4

– Indo-European: +0.65

– Turkic: -0.02

Thus, in general, the category demonstrates a weak positive correlation of the result with the volume of the train dataset. At the same time, the Pearson coefficient varies from one language family to another, which demonstrates the specific behavior of the category in relation to different languages. The highest value still does not confirm the correlation,

The lowest value is shown in Bengali, which belongs to the Indo-Aryan group of the Indo-European family: 9 layers have a value of 0.52, and 3 – 0.229. The highest values are for the languages of the Romance group of the Indo-European family, these languages have almost the same number of examples for languages, the value is the same for all languages. The listed Romance languages also match in values in the Mood category. In the Number category, the values are also almost identical (French has the highest ones).

4.3.2. Tense

It is noteworthy that three languages of the Romance group (Italian, Portuguese, Spanish) have the same layers that give similar values: layers number 5, 6, 7, 8, 9, 10, 11 have a value above 0.7, the rest of the layers appear below 0.4. This trend is not observed in the French language: on all 24 layers of the value low.

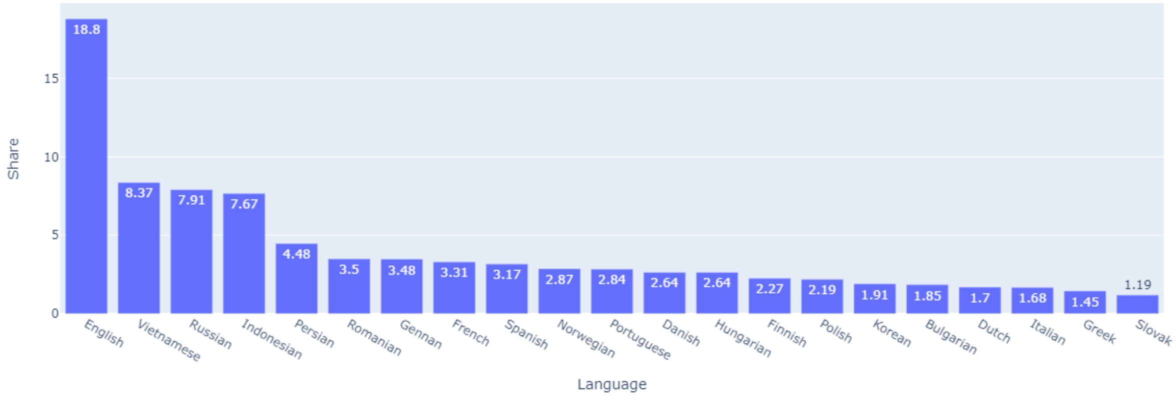
4.3.3. Case

Worst of all, the model perceives the case of the Indo-Iranian group (Amharic, Arabic). Portuguese gets the highest value (0.781 on all layers). Languages with an extensive case system, which include Turkicand Slavic, do not show high scores: Turkish – 0.445, Kazakh – 0.513; Slovak – 0.417, Serbian – 0.592.

4.4. XLM-R

XLM-RoBERTa base is trained on 100 languages and has 12 layers with a total number of parameters of 270M.

Figure 20: The first 20 languages in the distribution of languages in the pretraining dataset based on the quantity of the tokens for each language (the values are given as a percentage) (the data was taken from (Conneau et al., 2020))



Quite notably, like the previous one – the mBERT database is multilingual in the corpus, was adapted into a variety of Slavic languages (Russian, Slovak, Czech, Slovenian, Ukrainian), and also used low-resource languages such as Armenian, Kazakh and Tatar. It is also obvious that the Vietnamese (Austroasiatic language family) and the Indonesian (Austronesian language family) receive large shares. Thus, for XLM-R, shares English and Romance languages in the pretraining dataset less than in BLOOM (176B) and BLOOM-1b7.

Table 7: Categories that are well and poorly recognized by XLM-R

	Well recognized categories			**Poorly recognized categories**	
	Person	*Number*	*Mood*	*Tense*	*PronType*
average f-1 score	0.668	0.657	0.621	0.297	0.325

4.4.1. Number

Number is the characteristics of nouns but can be marked not only on the nouns, but also on other parts of speech.

First of all, it is worth noting that the values for each tested language within this category do not fluctuate, they are constant on all 12 layers.

The worst results are given by Western Armenian (0.584), as well as Croatian (0.599). The Armenian language belongs to the Armenian branch of the Indo-European family, so it can only be compared with Armenian belonging to the same group. The value of Armenian on each layer is 0.66. Croatian belongs to the Slavic group, which also includes Slovak (0.76 in each layer), Serbian (0.617).

Turkish (Turkic family) has one of the highest values (0.911). The Turkic family also includes Tatar (0.711), Kazakh (0.684). Note that in this category, the correlation between the result and the number of examples in the train dataset does not work: there are 729 such examples for Kazakh, and 117 for Tatar, at the same time Tatar is shown better. In other categories, this correlation also does not exist: for example, in *VerbForm*, Kazakh has 733 examples, Turkish has 368, Tatar has 107 examples, and the values are distributed as 0.599-0.809-0.377 accordingly. The Pearson coefficient is 0.07 for *Number* and 0.05 for *VerbForm*, which shows the lack of correlation.

4.4.2. Person

In this category, just as in the case of Number, all values are stable for all layers of the model.

The Afro-Asiatic family manifests itself poorly, it includes Amharic (0.688), Assyrian (0.45), Beja (0.643), Coptic (0.489). Arabic also belongs to this language family, but has a very high value (0.954). The same is true with the Indo-Aryan group of the Indo-European family: Hindi has a high value (0.97) in contrast to Sanskrit (0.663) and Bengali (0.229), which is manifested worse than all languages of all language families.

4.4.3. Mood

All layers have a constant value. The Western Armenian has the worst performance (0.461). At the same time, Armenian has a value of 0.909. As for other languages, the scores of which are slightly lower than the rest, Turkish has the value of 0.625, Classical Chinese – 0.703. All languages perform well, so it is impossible to note any correlations of the results with language groups or dataset volumes.

4.5. Comparison of models

After considering the models and some of their categories, we will try to find patterns and intersections in the results.

4.5.1. Recognition of language categories by different models

To begin with, there are the top-10 categories best and worst recognized by each model (see Table 8). We can note that the models have a lot in common. So, **Mood** is in the top-10 best categories in three of the four models (for BLOOM-1B7 Mood is in 14th place). The same is true for Polarity and for Degree. In the BLOOM model, more files have passed through probing, as a result, more categories have been identified, so it is not surprising that unpopular categories are in the first places in the ranking. These unpopular categories (PrepCase, Person, PunctSide, and etc.) are represented by one or several languages, which is why they have such high indicators relative to other categories. They have shifted such basic categories as Polarity and Degree, which manifest themselves well, but were tested in more languages, so there is a high probability of a large spread of values due to linguistic diversity.

It is significant that the Mood category is not only recognized well by all models under consideration, but also gives high values regardless of the specifics of the language. If a language is poorly recognized, then there may be languages with high values within its language family. This is also unrelated to the training dataset volume. **Person** behaves in a similar way.

Table 8: The top-10 of the best and worst categories for each model

	BLOOM	**BLOOM-1B7**	**mBERT**	**XLM-R**
The top-10 of the best recognised categories	1. PrepCase (0.994) 2. Person[abs] (0.907) 3. PunctSide (0.867) 4. NumValue (0.816) 5. NounType (0.808) 6. AdpType (0.788) 7. NumType (0.783) 8. Mutation (0.744) 9. Person[obj] (0.727)	1. AdpType (0.873) 2. PunctSide (0.823) 3. Polarity (0.76) 4. Animacy (0.687) 5. Strength (0.684) 6. Position (0.658) 7. Person[psor] (0.626) 8. Degree (0.589) 9. Number[psor]	1. Person[psor] (0.698) 2. Number (0.666) 3. AdpType (0.647) 4. Person (0.646) 5. Form (0.62) 6. Degree (0.619) 7. Polarity (0.592) 8. Mood (0.541) 9. Animacy (0.54)	1. AdpType (0.79) 2. Person[psor] (0.698) 3. Animacy (0.682) 4. Person (0.668) 5. Number (0.657) 6. Polarity (0.638) 7. Mood (0.621) 8. Degree (0.575)

	10. Mood (0.706)	(0.584) 10. Voice (0.542)	10. Number[psor] (0.521)	9. Number[psor] (0.539) 10. Gender (0.492)
The top-10 worst categories	1. InflClass (0.005) 2. NounClass (0.007) 3. ExtPos (0.085) 4. InflClass[nominal] (0.114) 5. VerbType (0.148) 6. Clas (0.188) 7. FocusType (0.201) 8. NameType (0.231) 9. Clitic (0.236) 10. Possessed (0.256)	1. InflClass (0.041) 2. Style (0.106) 3. NameType (0.117) 4. InflClass[nominal] (0.148) 5. Clusivity (0.18) 6. Clas (0.204) 7. Rel (0.282) 8. Definite (0..297) 9. Number[subj] (0.302) 10. Derivation (0.307)	1. NounBase (0.131) 2. PartType (0.167) 3. NameType (0.179) 4. Deixis (0.212) 5. Clusivity (0.225) 6. PronType (0.272) 7. Style (0.294) 8. VerbStem (0.322) 9. Tense (0.343) 10. VerbClass (0.351)	1. NameType (0.145) 2. Deixis (0.185) 3. Clusivity (0.267) 4. Style (0.294) 5. Tense (0.297) 6. Subcat (0.321) 7. PronType (0.325) 8. VerbClass (0.355) 9. NumForm (0.356) 10. NumType (0.379)

As for poorly recognized categories, there is a **NameType** in all four columns, and in models this category is not represented by exactly the same languages (for example, in BLOOM-1b7 it was revealed in Classical Chinese, Czech, Erzya, Latin, Russian, and in mBert – in Armenian, Classical Chinese and Western Armenian). Therefore, the category is not specific to languages.

Turning to the **Case** category, it cannot be said that the accuracy values are low for all languages. The results in this category depend on the language: the less extensive the case system in the language, the more likely higher values are (see comparison of Romance languages and Turkic languages). This rule also works for **Gender**. The average value for this category among the four models is 0.37. The values for languages are generally similar for four models (the error is less than 0.1). However, BLOOM shows a lot of dark areas indicating “dips” in accuracy (that is, among the values on the other layers, which may, for example, be about 0.5, there are almost zero values). In general, this trend is manifested not only in *Case*, but this category is most likely the richest in “dips”.

4.5.2. Correlation between pretraining dataset and results

Speaking about the influence of the share of language in the pretraining dataset on the accuracy score, most of the data consists of samples from high-resource languages. BLOOM models also include the low-resource languages of the Niger-Congo language family. As mentioned, mBERT and XLM-R should be more sensitive to the languages of the Slavic group of the Indo-European family since they have their in the pretraining dataset, which is not the case for BLOOM and BLOOM-1B7.

However, the average values of Slavic languages (for example, Croatian, Slovak, Serbian) are identical in four models. In some categories, BLOOM is worse, but a lower average value is associated with “dips” on some layers, while the values of the remaining layers are about the same as in both BERT and XLM-R.

The discrepancy in the ranking of some languages (see Figure 21 and Figure 22) may occur due to the fact that different categories were sometimes considered in the same languages. So, the average value of Hindi in mBERT is about 0.97, while in other models it is about 0.5-0.6 because in this situation, more categories were included. If we do not take into account such cases, languages in general give similar values, but BLOOM and BLOOM-1B7 differ from the values of other models by about 0.1 in each category.

It is possible to allocate some of the languages best recognized by all models: Portuguese, Chinese, Spanish. Among the languages with low scores are Bengali and Western Armenian.

Figure 21: The first graph reflects the languages with the highest average values, and the second graph shows the languages that performed the worst (results of the BLOOM model)

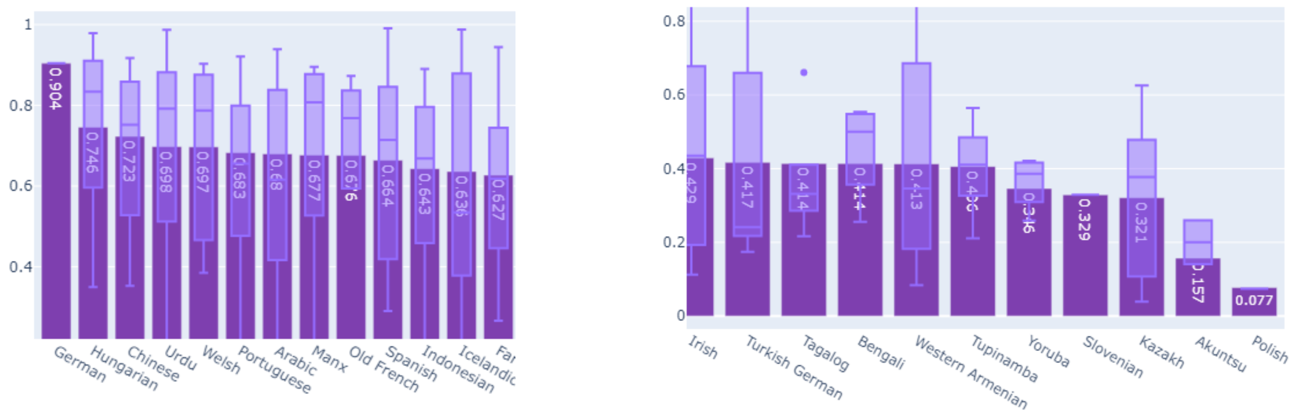
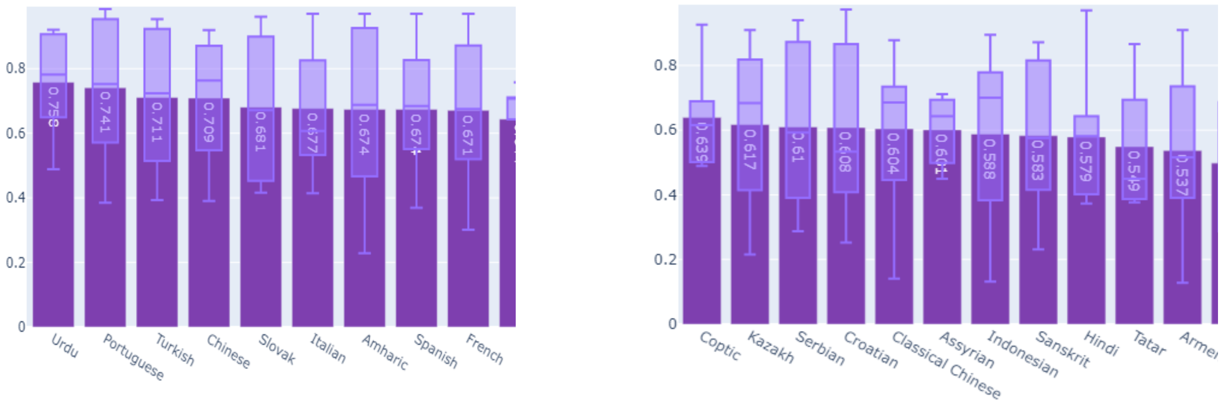


Figure 22: The first graph reflects the languages with the highest average values, and the second graph shows the languages that performed the worst (results of the XLM-R model)



There are also some deviant cases. So, French is recognized much worse in BLOOM than in other models (see Table 9), although the percentage of this language in the pretraining dataset is almost the same as in BLOOM-1B7 (13%) (since the number of tokens in the BLOOM model is greater than in BLOOM-1B7 (366B > 350B), then 13% in BLOOM constitutes more tokens for French, than in BLOOM-1B7, but the value is still lower).

Table 9: Average values of some languages in different models

	BLOOM	*BLOOM-1B7*	*mBERT*	*XLM-R*
French	0.487	0.652	0.689	0.671
Italian	0.601	0.589	0.72	0.677
Kazakh	0.321	has not been tested	0.636	0.617
Turkish	0.509	has not been tested	0.72	0.711
Tatar	0.474	has not been tested	0.571	0.549

The corresponding languages of the Slavic group give similar average values, but in BLOOM and BLOOM-1B7 these values are about 0.1 < than in mBERT and XLM-R (for example, Slovak in mBERT = 0.689, and in BLOOM = 0.53, Serbian in BLOOM = 0.552, and in XLM-R = 0.61). According to Table 9, Kazakh and Turkish have lower values in BLOOM than in other models. However, Tatar has comparable values in most models, although it is also contained in the pretraining dataset only in mBERT and XLM-R models.

Accordingly, in some cases, the share of the language in the pretraining dataset significantly increases accuracy (as with Kazakh and Turkish), sometimes it has a weak effect (as with Slovak and Serbian), and in most cases it does not affect accuracy (Tatar). However, this correlation also requires accurate calculations using token sizes for each language.

4.5.3. Language families and results

Although according to the ANOVA test conducted in the articles (le Scao et al., 2023; Serikov et al., 2022), the results of BLOOM and BLOOM-1B7 (p-value < 0.01), mBERT and XLM-R (p-value = 0.0005) are highly correlated with language family, we have not managed find direct evidence of this connection in any of the models. In many language families, even a small size, F1-score was different for most languages.

For instance, the languages of the Slavic group can be grouped according to the similarity of trends in categories, for example, here are two of them:

– Croatian, Slovak, Bulgarian

– Old Church Slavonic and Old East Slavic

However, even such a small division is not completely correct, because Slovak has much more values close to zero. It is also unclear where to include Serbian, because it is generally similar to the first group, but has more stable values.

Difficulties in dividing languages into groups based on the proximity of their values also arose when trying to understand whether the languages of the Romance group have the similar scores. Most often, Spanish is close to Portuguese and Italian, demonstrating relative stability of values, and French represents an outlier (values are often accompanied by failures, speaking about BLOOM), for instance, in the categories Person, Definite. For the PronType category, French values are much lower than the Spanish, Portuguese and Italian ones, but Portuguese is recognized better than Spanish and Italian (by about 0.5). In some categories, all 4 reviewed languages performance in the same way, as, for example, in the case of NumType. It also happens that French, on the contrary, is better manifested (*Gender* category).

There are cases when languages of the same language group (a narrower language classification) have the same weak layers (for instance, three languages of the Romance group have similar results in seven layers – see section 4.3), but the described pattern does not extend to the entire language family.

Thus, such attempts at classification demonstrate that even within the same language group there are no uniform trends, and similar patterns in behavior can be identified only in pairs or triples of languages.

4.5.4. Model layers

As already mentioned, BLOOM has the most unstable performance on all categories. However, the comparison cannot be called completely fair, because there are ≈ 2.9 times more layers in the BLOOM model (70 layers) than in mBERT (24 layers) and ≈5.8 times more than in BLOOM–1B7 and XLM-R (12 layers). Nevertheless, the values on all layers of the BLOOM model fluctuate greatly. Since mBERT and XLM-R basically have the same value on all layers, this value corresponds to the maximum value of the BLOOM model, and the values on the other layers of BLOOM are lower and undergo drastic changes. The values of the BLOOM-1B7 layers, on the contrary, are not subject to frequent fluctuations.

The described pattern, manifested in the BLOOM model, is demonstrated by the values of the Bengali language (see Table 10):

Table 10: Some trends of models for four categories (Aspect, Tense, Person, VerbForm) of Bengali language

	Aspect	*Tense*	*Person*	*VerbForm*
BLOOM	fluctuations from 0 to *0.333*	fluctuations from 0 to *0.643*	fluctuations from 0 to *0.533*	fluctuations from 0.1 to *0.643*
BERT	*0.333* on all layers	*0.643* on all layers	fluctuations from 0.229 to *0.533*	*0.643* on all layers
XLM-RoBERTa	*0.333* on all layers	*0.643* on all layers	0.229 on all layers	*0.643* on all layers

This pattern is also common among other languages (Sanskrit, Armenian, etc.). So, it seems that BLOOM generally recognizes the Afro-Asiatic family worse. However, Figure 23 and Figure 24 show that the problems in the BLOOM model are associated with certain layers, the maximum values are identical to the values of the XLM-R and BERT models.

Figure 23: *Number* category in the Afro-Asiatic family (BLOOM)

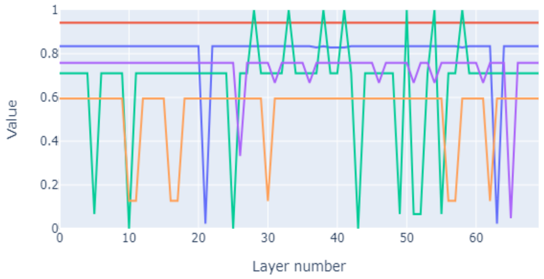
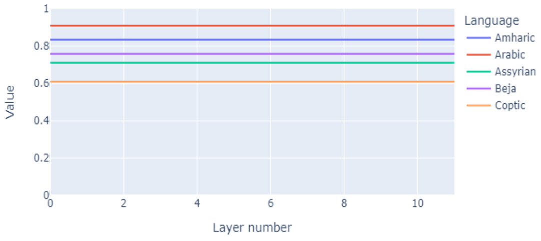


Figure 24: *Number* category in the Afro-Asiatic family (mBERT)

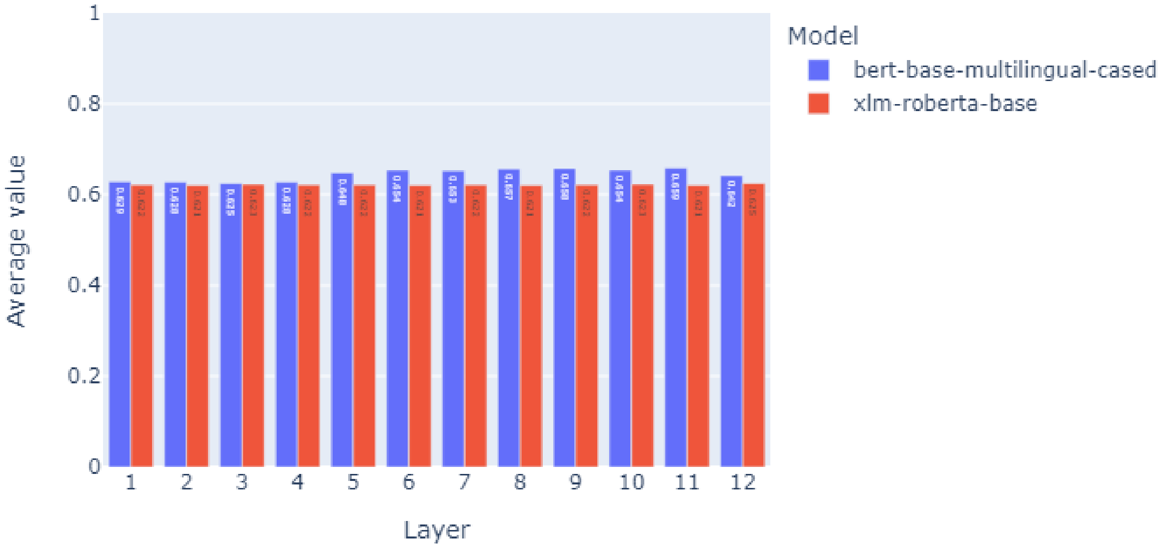


Graphs with the same parameters for BLOOM-1B7 and XLM-R are not given, since the BLOOM graph is identical to BLOOM-1b7, and the mBERT graph is XLM-R.

Due to this trend, the average values for the layers for BLOOM and BLOOM-1B7 are lower than for mBERT and XLM-R. The largest average value is on the 70th layer of the BLOOM model (0.566). As for BLOOM-1B7, the 23rd layer has the largest value (0.592), the average values for mBert and XLM-R exceed 0.6.

The models can be arranged in the following order, depending on the **delta of values** by layers: BLOOM > BLOOM-1B7 > mBERT > XLM-R, where the sample ranges (the difference between the largest and smallest values) are 0.052 > 0.039 > 0.034 > 0.004, respectively: that is, the most stable values in the XLM-R model, and BLOOM has the least. However, although more stable than the XLM-R value, mBERT outputs values higher (see Figure 25).

Figure 25: Average values for each layer for mBERT and XLM-R models



4.5.5. Nominal and verbal categories

There is no strong correlation between which categories, nominal or verbal, are recognized worse on average. In each model, a pair of best and worst categories can be distinguished, and each of them includes both a nominal category and a verbal one (for example, Mood and Person are some of the best categories, where the first category is verbal, and the second is nominal).

4.5.6. Correlation between training dataset size and results

Relying on the Dashboard, we could not find categories whose accuracy is linked with the volume of the training dataset. To confirm the absence of such patterns, **Pearson correlation coefficients** were calculated for each model and the corresponding categories, where the result can range from -1 to 1, where a negative value means a complete absence of correlation, and a positive one means its presence.

After the first attempt of running this test, many categories were revealed, represented by only a few languages (for example, Person[obj] in BLOOM is only in Erzya, Kiche and

Yupik), so such coefficients may be random. Therefore, we got rid of such categories when calculating this indicator.

The absence of correlation or its weak manifestation is the main trend for all models (see Table 10). The results showed that this correlation is least evident in the BLOOM model (see Table 11). At the same time, the results of BLOOM-1B7 are more correlated with the volume of the train dataset than the scores of BLOOM. Then the models can be arranged in descending order of the correlation score: XLM-R > mBERT > BLOOM-1B7 > BLOOM. It is worth noting that mBERT and XLM-R have very similar values.

Table 10: Percentage of cases with different types of Pearson correlation in different models

	BLOOM	*BLOOM-1B7*	*mBERT*	*XLM-R*
No correlation (r < 0.25)	75%	69%	38,5%	46%
Weak correlation (0.25 < r < 0.5)	20%	18,5%	38,5%	27%
Moderate correlation (0.5 < r < 0.75)	0%	12,5%	15%	27%
Strong correlation (r > 0.75)	4%	0%	8%	0%

Table 11: Pearson correlation coefficients for *Voice, Degree* and *Person* category, calculated for different models

	BLOOM	*BLOOM-1B7*	*mBERT*	*XLM-R*
Voice	0.44	0.63	0.5	0.54
Degree	0.34	0.4	0.49	0.56
Person	0.25	0.59	0.39	0.61

5. Conclusion

Interpreting the results of multi-lingual probing is not an easy task. If a large number of languages and their corresponding grammatical categories are passed through probing, the amount of data becomes so large that there is a need for visualization. In this paper, a solution to this problem is proposed in the form of a dashboard, which includes more than 10 tools that help turn raw numbers into decorated graphs, which then make it possible to interpret the data. The data submitted to the input is analyzed by the dashboard, namely: the number of models among the json files is fixed, the classification of languages is carried out based on their belonging to certain language families. As the probing results are processed, new data is generated, which become the basis for future graphs. The mechanism calculates such indicators as the arithmetic mean, the distance between lines, and also creates other datasets, such as datasets for each language family. All this then enters the chart patterns that are already visible to the user.

These plots, built on the basis of Plotly, provide opportunities for interaction with them. For example, a heatmap that reflects all languages and all categories contains many variables, so the user can zoom in on the areas he needs, cut them out and save them in png format. Line graphs that reflect similar and dissimilar trends within the same language family show as many trends as the user specifies. Even if there are too many of them, but you want to see everything, you can hide the selected trends.

Moreover, many charts are accompanied by dropdowns. This allows the user to select the desired language families and compare them on the same graph. Double dropdowns are especially interesting because the selected option of the first one affects the content of the second one.

In this work, an attempt was made to derive universals based on the consideration of languages as parts of language families. The dashboard meets this task, so there is a special series of graphs in it in order to see which languages each language family is represented by, as well as to study its results from different angles.

As for the results obtained, the dashboard allowed us to check possible correlations, compare models and deduce general patterns and specific behaviors of models:

1. There are categories that, apparently, are easy (Mood, Person) and complex (Name Type) for recognition by all models, that is, such categories are not specific to certain languages. There are also categories whose meaning depends on the language (Case).
2. Generalizations to language families failed because the behavior of related languages was sometimes very heterogeneous.
3. We have established a scale based on the stability of average values by layers: XLM-R (the most stable) > mBERT > BLOOM-1B7 > BLOOM.
4. No layers with permanently high or low values were detected. The similarity of the layer scores can be revealed in the languages of language groups or families, but this is not a common case.
5. There is no strong correlation between which categories, nominal or verbal, are recognized worse on average.

6. There is no strong correlation between the pretraining dataset size in the models. It can be revealed only for a part of the languages of the language family. However, it was not possible to determine the principle by which languages are chosen, so there are no strong arguments that could prove the uniqueness of this correlation for the model. The two pairs of models under consideration, which have different languages in the pretraining dataset, in many cases show similar results, however BLOOM and BLOOM-1B7 consistently show scores worse than other models, regardless of the language. However, to fully test this correlation, it still requires accurate calculations with reference to the size of tokens for each language.
7. Correlation between training dataset size and accuracy scores is absent or weak in all the presented models, but we can still build some ranking: XLM-R > mBERT > BLOOM-1B7 > BLOOM.

Moreover, during the analysis using the dashboard, cases that were not amenable to any regularities were found, which were later identified as shortcomings of treebanks.

This dashboard is based on the fact that the user selects a model and a lot of graphs are provided for this model. Further improvements could be the addition of tools that would allow us to compare different models. In Dash Plotly, there is no possibility to select parameters in pairs using dropdown, you can only select parameters in the second dropdown, which appear depending on the value of the first one. Accordingly, such graphs are interesting that could provide a choice of “model-category” and that this pair should be selected independently of the others. Thus, future dashboard updates can be aimed at creating instruments that make it possible to choose any models, categories and languages for comparative analysis. Now this shortcoming can be solved by multi-window mode.

The Dashboard is available on Github: https://github.com/alinaavanesyan/probing-dashboard.

References

Bhattacharya, S., Zouhar, V., Bojar, O. (2022). Sentence Ambiguity, Grammaticality and Complexity Probes. *arXiv preprint arXiv:2210.06928v2*.

Conneau, A., Kruszewski, G., Lample, G., Barrault, L., & Baroni, M. (2018). What you can cram into a single vector: Probing sentence embeddings for linguistic properties. *arXiv preprint arXiv*:*1805.01070*.

Conneau, A., Kiela, D. (2018). Senteval: An evaluation toolkit for universal sentence representations. *ArXiv*, abs/1803.05449.

Conneau, A., Khandelwal, K., Goayl, N., Chaudhary, V., Wenzek, G., Guzman, F., Grave, E., Ott, M., Zettlemoyer, L., Stoyanov, V. (2020). Unsupervised Cross-lingual Representation Learning at Scale. *arXiv preprint arXiv: arXiv:1911.02116v2.*

de Marneffe, M.-C., Manning, D.C., Nivre, J., & Zeman, D. (2021). Universal Dependencies. *Computational Linguistics*, 47(2):255–308.

Devlin, J., Chang, M., Lee, K., Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv: arXiv:1810.04805v2.*

Dotton, Z., Doyle Wagner, J. (2018). A Grammar of Kazakh. *Center for Slavic, Eurasian and East European Studies.*

Durrani, N., Sajjad, H., Dalvi, F., Belinkov, Y. Analyzing individual neurons in pre-trained language models, CoRR abs/2010.02695 (2020). *arXiv:2010.02695*.

Ferreira, D., Rozanova, J., Thayaparan, M., Valentino, M. (2021). Does My Representation Capture X? Probe-Ably. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations,* pp. 194–201.

Hewitt, J., & Liang, P. (2019). Designing and interpreting probes with control tasks. *arXiv preprint arXiv:1909.03368*.

Le Scao, T., Fan, A., Akiki, C., Pavlick, E., Ilic, S., et al. (2023). BLOOM: A 176B-Parameter Open-Access Multilingual Language Model. *arXiv preprint arXiv: 2211.05100v3*.

Rogers, A., Kovaleva, O., Rumshisky, A. (2020). A primer in BERTology: What we know about how BERT works. *Transactions of the Association for computational Linguistics 8*, pp. 842-866.

Serikov, O., Protasov, V., Voloshina, E., Knyazkova, V., Shavrina, T. (2022). Universal and Independent: Multilingual Probing Framework for Exhaustive Model Interpretation and Evaluation. *arXiv preprint arXiv:2210.13236.*

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Appendix A. The Dashboard content

A.1 Graphs

Here are the graphs included in the Dashboard (the data on the graphs correspond to the mBERT model).

Figure 6: Average value for each category

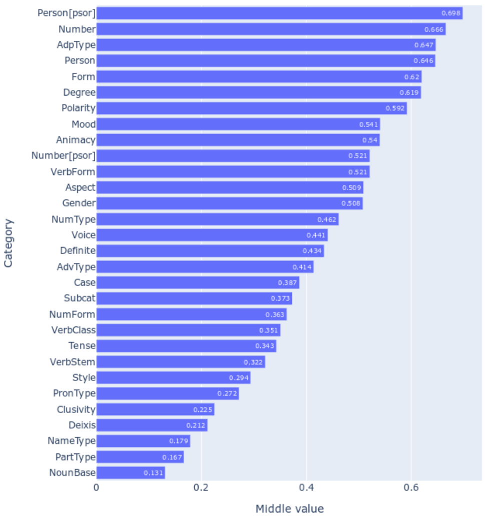


Figure 7: Average values for each layer

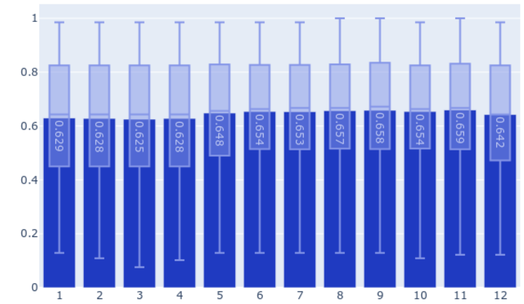


Figure 8: Average values of the languages

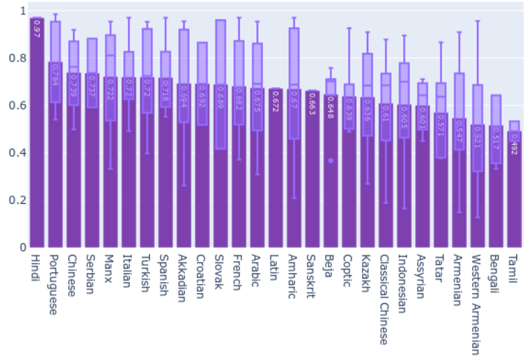


Figure 9: Results grouped by languages and average values (for each “category-language” pair)

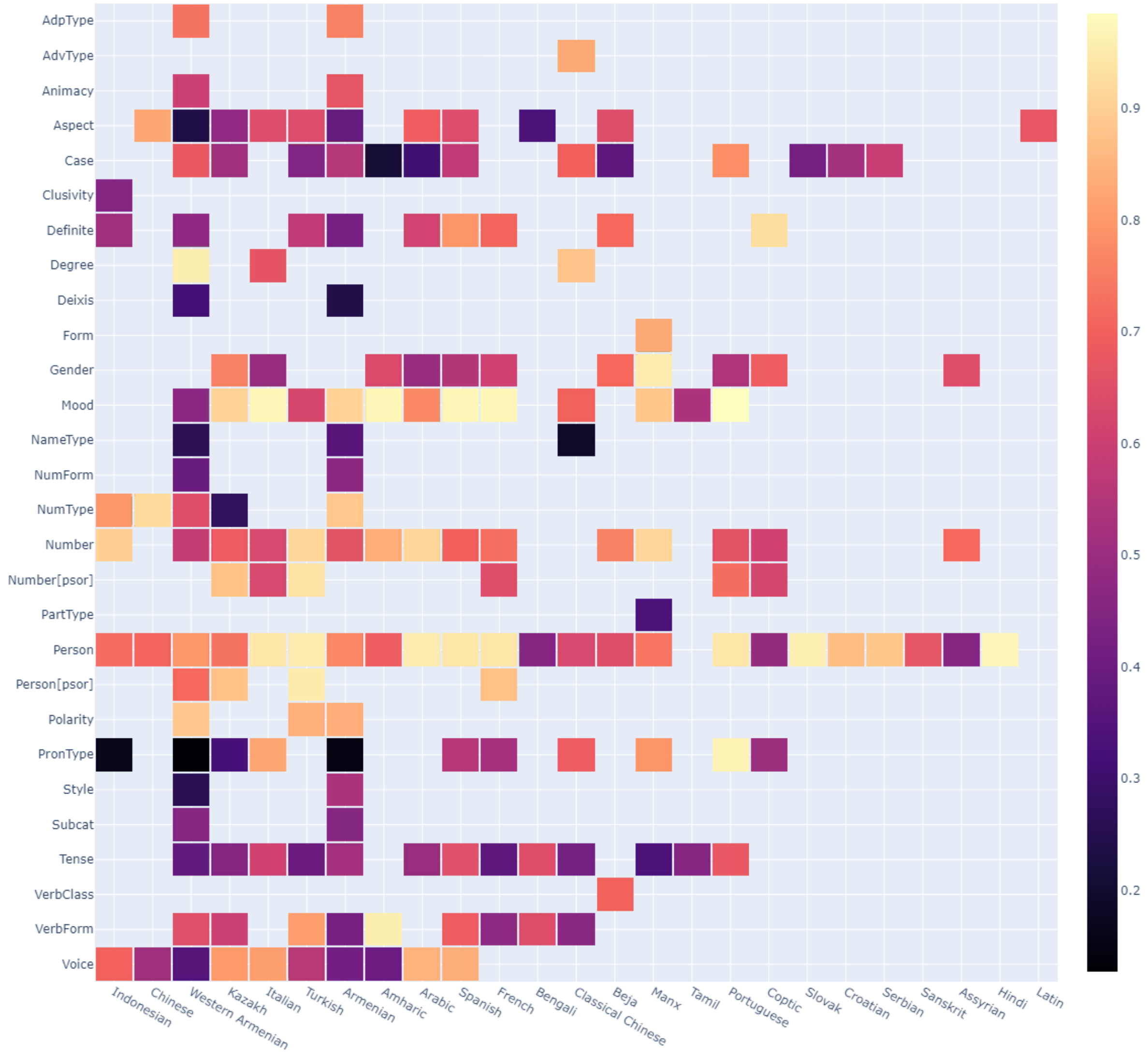


Figure 10: Average values for all families

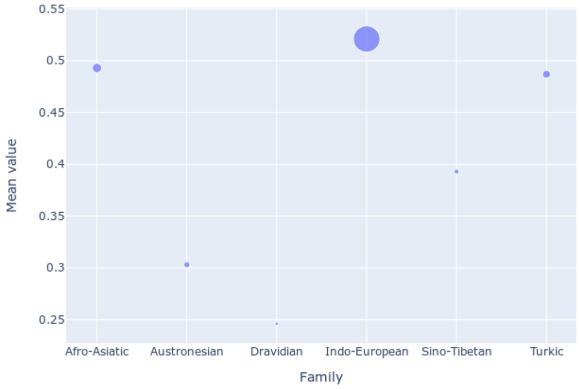


Figure 11: Structure of the language family

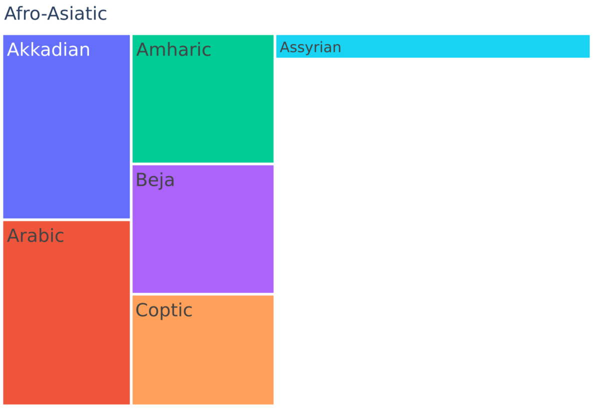


Figure 12: The distribution of the average values for each category. The second and third images reflect the data that appears when you hover over the scatter plot and boxplot, respectively

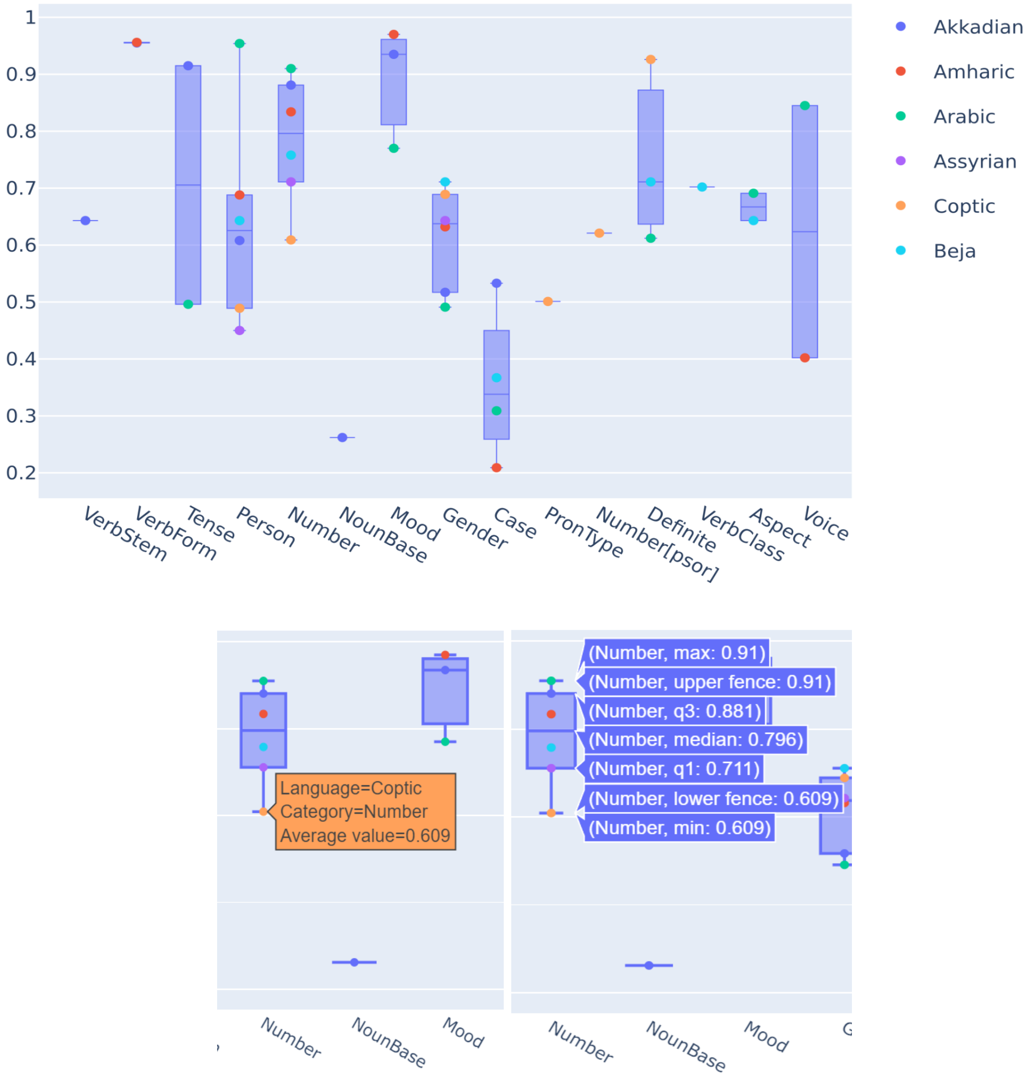


Figure 13: Trends that can be grouped by their similarity/dissimilarity and those that are non-comparable

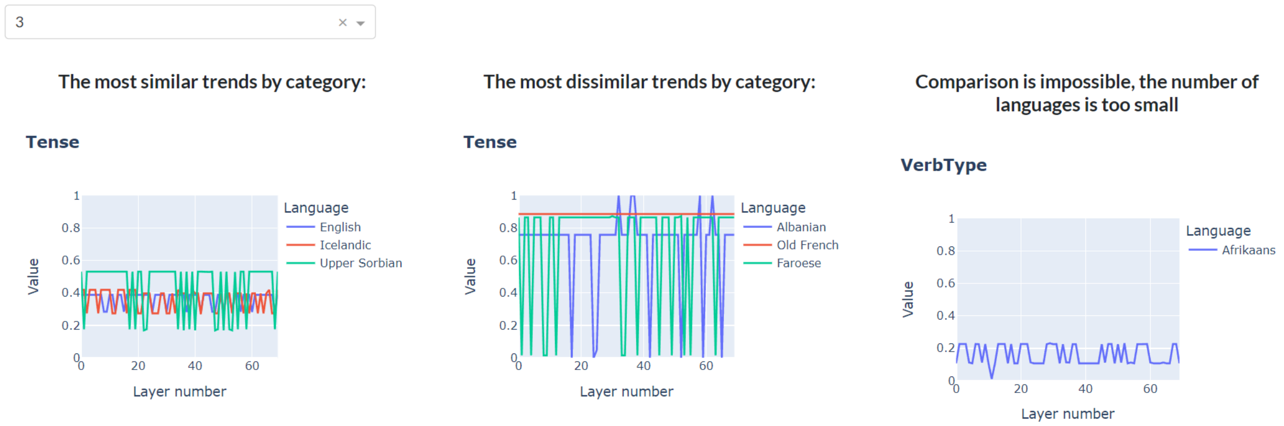


Figure 14: Average values of language families

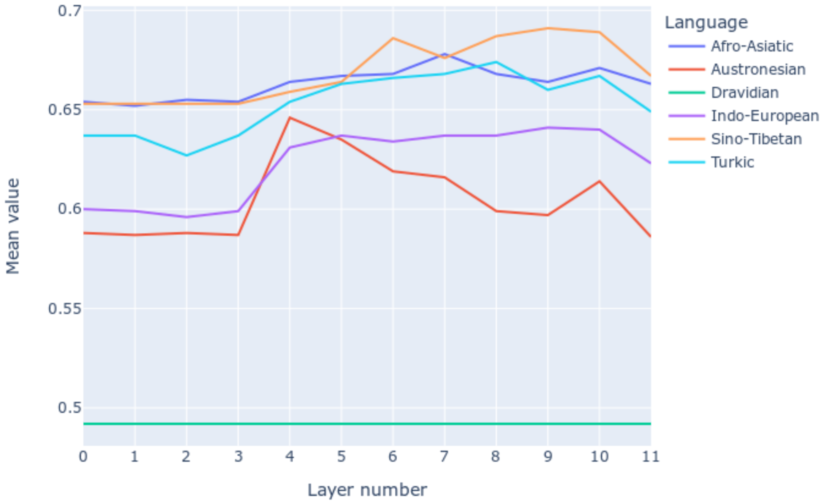


Figure 15: Values of all languages in which the selected category is represented

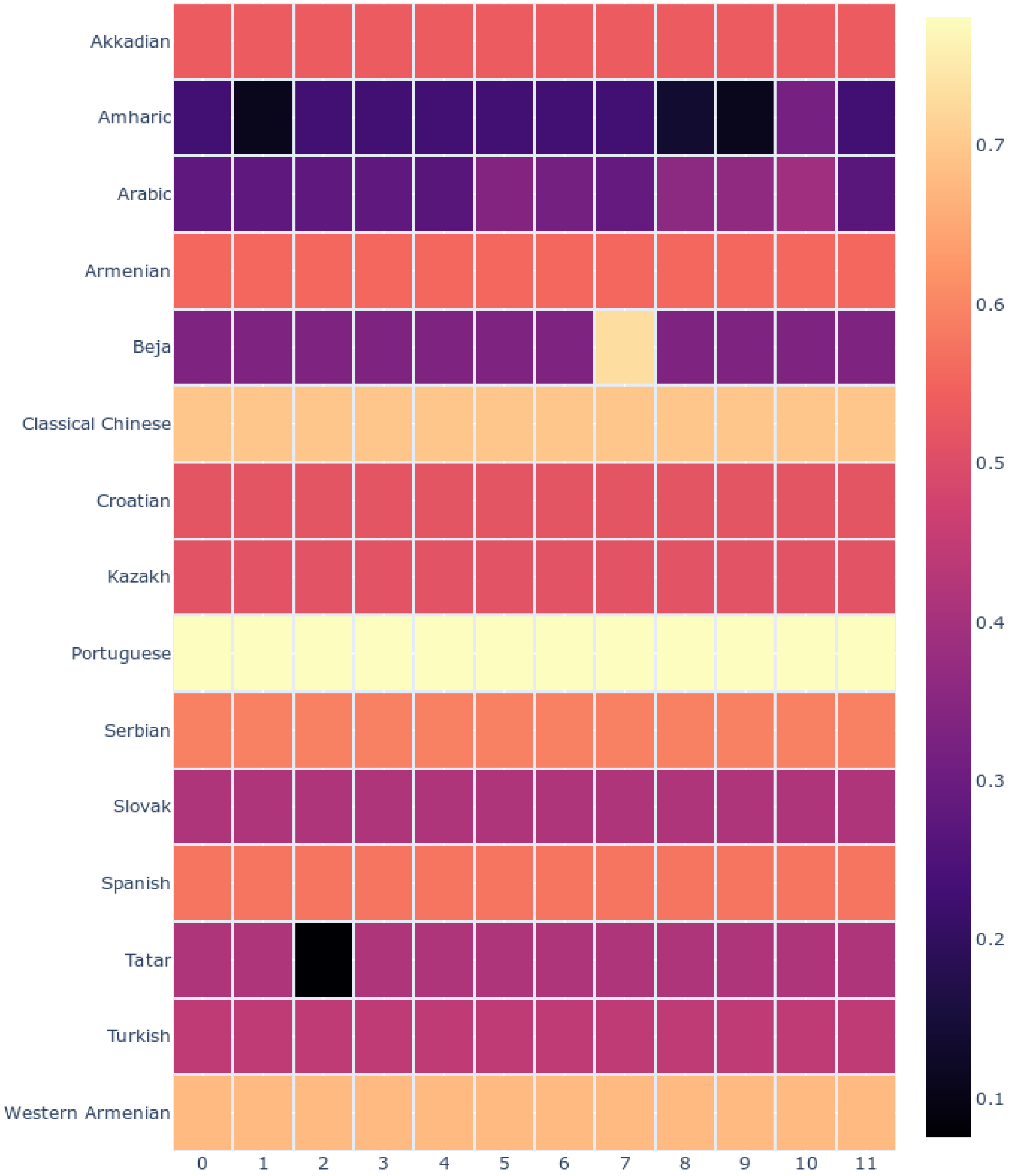

Akkadian
Amharic
Arabic
Armenian
Beja
Classical Chinese
Croatian
Kazakh
Portuguese
Serbian
Slovak
Spanish
Tatar
Turkish
Western Armenian
0
1
2
3
4
5
6
7
8
9
10
11
0.7
0.6
0.5
0.4
0.3
0.2
0.1

Figure 16: Values of the languages represented in the selected category

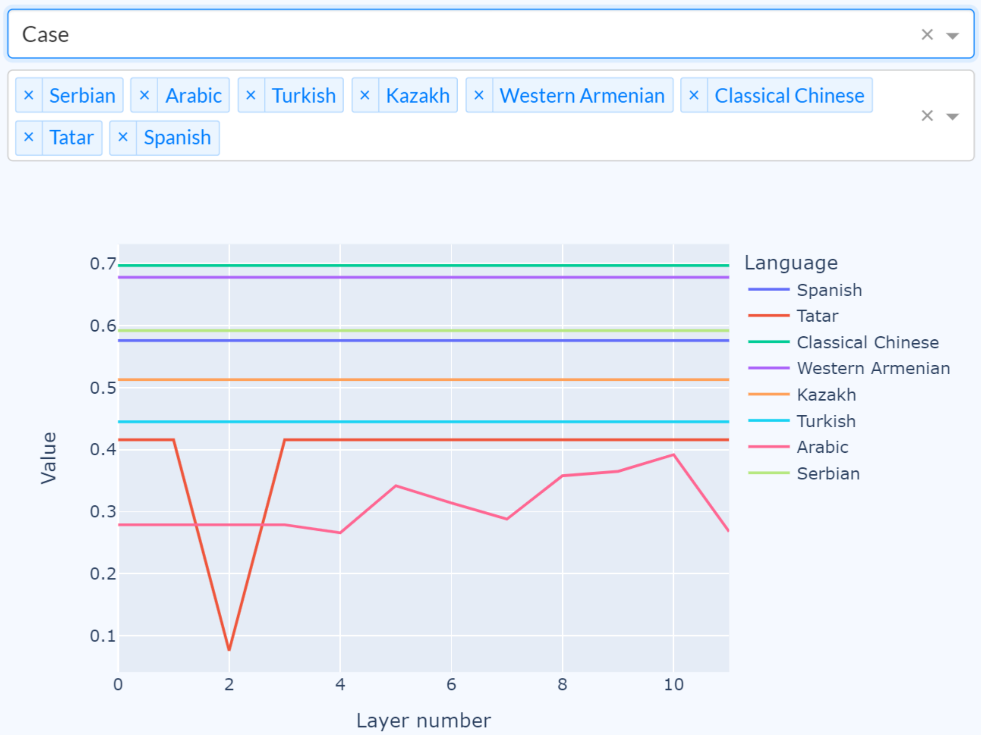
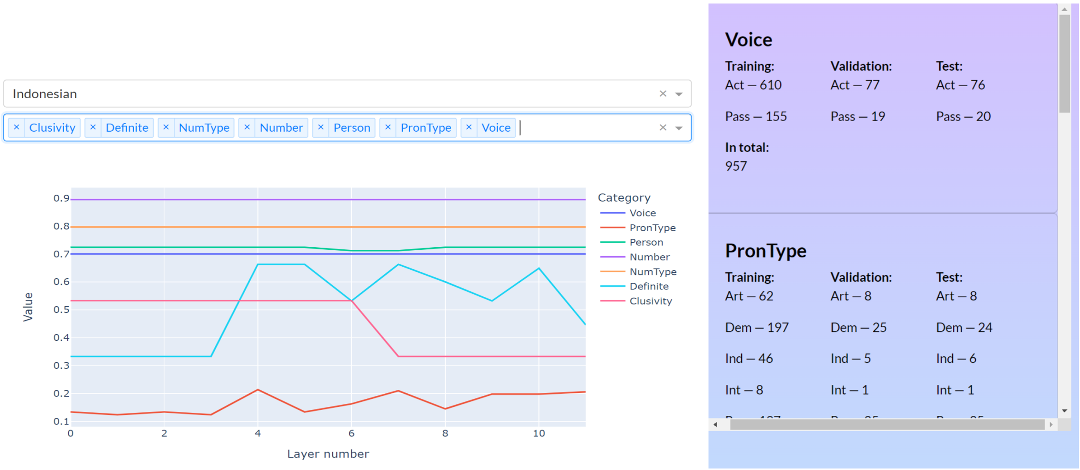


Figure 17: Values of the categories of the selected language



Voice

Training:	**Validation:**	**Test:**
Act – 610	Act – 77	Act – 76
Pass – 155	Pass – 19	Pass – 20

In total:
957

PronType

Training:	**Validation:**	**Test:**
Art – 62	Art – 8	Art – 8
Dem – 197	Dem – 25	Dem – 24
Ind – 46	Ind – 5	Ind – 6
Int – 8	Int – 1	Int – 1

A.2 Manual

This Dashboard helps to visualize the probing data you have uploaded. Below you will find a description of each chart. Remember that since this dashboard is based on Plotly, you can interact with each chart: use zoom in/out, save graphs as PNG, hide trends on line graphs (by clicking on the legend of the desired language) and much more.

Map – it provides the genealogical data on the languages, so you can choose the language(s) that you need and see its language family, subfamily, genus and the distribution area of chosen language(s).

Graph 1 – "Average values of each category" (horizontal bar chart)
It provides the average values of all categories presented in the selected model.

Graph 2 – "Average values for all families" (*scatter plot*)
It displays the average values of language families, where the size of the points corresponds to the number of languages of this family represented in the files. This graph is accompanied by a sidebar "Number of files", which counts the files in which the languages of this family are represented.

Graph 3 – "Average values of each layer" (bar chart + boxplot)
It shows the average values of each layer of the selected model, and also reflects the spread of values (information appears when hovering over the boxplot).

Graph 4 – "Average values of all languages" *(bar chart + boxplot)*
It gives a rating of languages sorted by average values (when you hover over the boxplot, you can see the spread of values for each language).

Graph 5 – "Average values of all categories (for each language family)" (line graph)
First, you select a language family (one or more), and then lines become available that reflect the average values of the language families in each layer.

Graph 6 – "Values of the languages represented in the selected category" (line graph)
There are two dropdowns where first you can choose the category and then you will see the list of languages presented in the selected category. It shows the average values of languages.

Graph 7 – "Values of the categories of the selected language" (line graph)
This graph shows the average values of the selected categories for a particular language. Statistics are also available for each selected category: the number of examples presented in the train, validation and test dataset.

Graph 8 – "Structure of the language family" (treemap)
It reflects the structure of the selected language family, where the size of cells is linked with the number of files presented in the probing data for each language.

Graph 9 – "Average values by category" (boxplot + scatter plot)
It shows the range of average values for all categories presented in the chosen language

family. If the switch "Show languages" is on, you can see the points corresponding to the language average values.

Graph 10 – "The most similar/dissimilar trends & non-comparable trends by category" (line graphs)
There are three columns. The similarity of trends is calculated by comparing all languages with the pattern line (median trend), so the first and the second columns show the languages that are close to this pattern line to the greatest and least degree (the Frechet distance is used). With the help of the dropdown, you can select the quantity of languages that you want to see on the charts. However, if the number is maximum, then two columns will be identical. The third column reflects the categories represented by a small quantity of languages, so it would be unfair to compare trends.

Graph 11 – "Average values for each "category-language" pair" (heatmap)
The cell of this heatmap is the average values for the certain category and language. The darker the cell, the lower the value.

Graph 12 – "Values of all languages in which the selected category is represented" (heatmap)
It provides the data for each language from the category selected in the dropdown. Here you can see the average values for each language for each layer of the model.

Appendix B. Languages information

Table 1: Languages, their families and codes corresponding to Universal Dependencies data

Codes	Language	Family	Codes	Language	Family
af	Afrikaans	Indo-European	*kpv*	Komi Zyrian	Uralic
akk	Akkadian	Afro-Asiatic	*ko*	Korean	Koreanic
aqz	Akuntsu	Tupian	*kmr*	Kurmanji	Indo-European
sq	Albanian	Indo-European	*la*	Latin	Indo-European
am	Amharic	Afro-Asiatic	*lv*	Latvian	Indo-European
grc	Ancient Greek	Indo-European	*lij*	Ligurian	Indo-European
apu	Apurina	Arawakan	*lt*	Lithuanian	Indo-European
ar	Arabic	Afro-Asiatic	*olo*	Livvi	Uralic
hy	Armenian	Indo-European	*nds*	Low Saxon	Indo-European
aii	Assyrian	Afro-Asiatic	*gv*	Manx	Indo-European
bm	Bambara	Mande	*mr*	Marathi	Indo-European
eu	Basque	-	*gun*	Mbya Guarani	Tupian
bej	Beja	Afro-Asiatic	*mdf*	Moksha	Uralic
be	Belarusian	Indo-European	*myu*	Munduruku	Tupian
bn	Bengali	Indo-European	*pcm*	Naija	Atlantic-Congo
bho	Bhojpuri	Indo-European	*sme*	North Sami	Uralic
br	Breton	Indo-European	*no*	Norwegian	Indo-European

bg	Bulgarian	Indo-European	*cu*	Old Church Slavonic	Indo-European
bxr	Buryat	Mongolic-Khitan	*orv*	Old East Slavic	Indo-European
ca	Catalan	Indo-European	*fro*	Old French	Indo-European
zh	Chinese	Sino-Tibetan	*fa*	Persian	Indo-European
lzh	Classical Chinese	Sino-Tibetan	*pl*	Polish	Indo-European
cop	Coptic	Afro-Asiatic	*pt*	Portuguese	Indo-European
hr	Croatian	Indo-European	*ro*	Romanian	Indo-European
cs	Czech	Indo-European	*ru*	Russian	Indo-European
da	Danish	Indo-European	*sa*	Sanskrit	Indo-European
nl	Dutch	Indo-European	*gd*	Scottish Gaelic	Indo-European
en	English	Indo-European	*sr*	Serbian	Indo-European
myv	Erzya	Uralic	*sms*	Skolt Sami	Uralic
et	Estonian	Uralic	*sk*	Slovak	Indo-European
fo	Faroese	Indo-European	*sl*	Slovenian	Indo-European
fi	Finnish	Uralic	*es*	Spanish	Indo-European
fr	French	Indo-European	*sv*	Swedish	Indo-European
gl	Galician	Indo-European	*tl*	Tagalog	Austronesian
de	German	Indo-European	*ta*	Tamil	Dravidian
got	Gothic	Indo-European	*tt*	Tatar	Turkic
el	Greek	Indo-European	*th*	Thai	Tai-Kadai
gub	Guajajara	Tupian	*tpn*	Tupinamba	Tupian
he	Hebrew	Afro-Asiatic	*tr*	Turkish	Turkic
hi	Hindi	Indo-European	*qtd*	Turkish German	Indo-European
hu	Hungarian	Uralic	*uk*	Ukrainian	Indo-European
is	Icelandic	Indo-European	*hsb*	Upper Sorbian	Indo-European
id	Indonesian	Austronesian	*ur*	Urdu	Indo-European
ga	Irish	Indo-European	*ug*	Uyghur	Turkic
it	Italian	Indo-European	*wbp*	Warlpiri	Pama-Nyungan
jv	Javanese	Austronesian	*cy*	Welsh	Indo-European
urb	Kaapor	Tupian	*hyw*	Western Armenian	Indo-European
krl	Karelian	Uralic	*wo*	Wolof	Atlantic-Congo
arr	Karo	Tupian	*sjo*	Xibe	Tungusic
kk	Kazakh	Turkic	*sah*	Yakut	Turkic
quc	Kiche	Indo-European	*yo*	Yoruba	Atlantic-Congo
koi	Komi Permyak	Uralic	*ess*	Yupik	Eskimo-Aleut