Guidelines on language data visualization

During the development of the language page, we faced a number of difficulties in terms of language data visualization. The guidelines consider five of them: the representation of genealogy, areal distribution, domains of language usage, dynamics of language usage and phonetic data. Each subsection is structured according to the following scheme: theoretical problems, related tasks of visual representation and the technical implementation of this representation.

1. Representation of genealogy

Genealogical relationship is basic information that is usually given in the beginning of a language description in the form of a short note about a related language group and a family; it does not imply further enumeration of related languages belonging to the same group or other groups of the same family. Still, since our website targets a wide range of users, we find it crucial to provide a complete picture of the family tree for each language.

We follow the theoretical paradigm developed in the Institute of Linguistics regarding the list of languages and genealogical relationships. For example, we consider that separate groups of the Altaic languages are genetically related and not just form a Sprachbund. Moreover, we keep track of the latest studies in this field and present a new division within the Uralic language family on our site.

It is important for us to achieve the following goals: easy usage; adequate data display; visualization of the full genealogical picture (from dialect to family) with a possibility to narrow it down to specific segments.

As a solution, we have chosen two interactive schemes that differ in the way of data visualization. The first scheme is a common representation of a language family as a genetic language tree (see Figure 1). Most trees have a lot of branches, so we provided an opportunity to collapse tree nodes to easily focus on a specific segment of the scheme. The collapsed nodes are coloured grey. To make the diagram clearer we coloured the nodes of the languages, groups of dialects and dialects in contrasting colours that distinguish them from the higher level nodes. The values of the colours are given in the diagram legend.

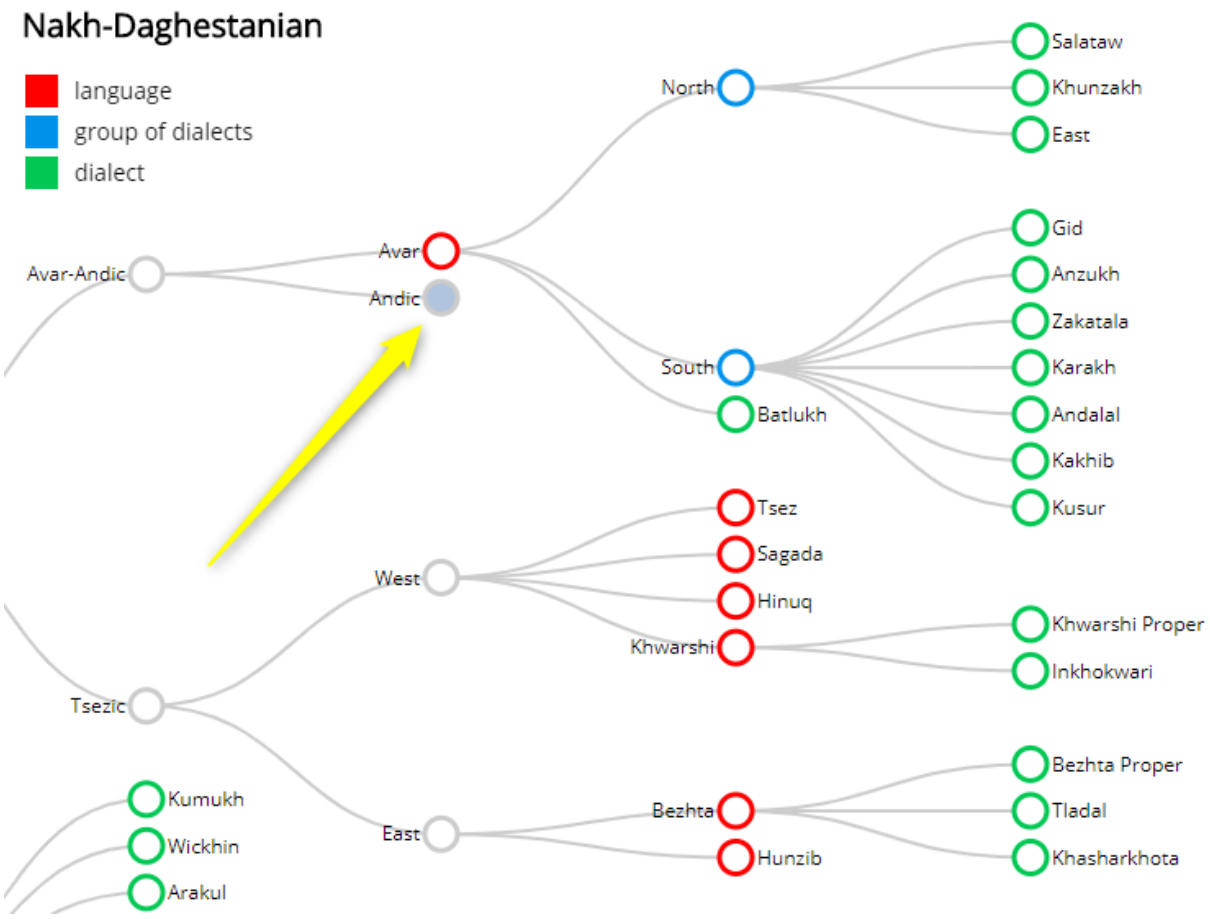

Figure 1: A fragment of the Nakh-Daghestanian genealogical tree. The arrow points at the collapsed node that can be unfolded.

The visualization is based on the D3 tree diagram. The data presented in the tree are pre-packaged in a json file.

The default structure of the json file from D3 tree did not meet our goals, so we needed to expand the set of properties and values. Our json file specifies the following properties for the elements: child / parent elements and colors indicating that the elements belong to a language, a group of dialects, or a dialect. A snippet of such a file is given in Figure 2.

```
var treeData = [
  {
    "name": "Nakh-Daghestanian",
    "children": [
      {
        "name": "Nakh",
        "children": [
          {"name": "Chechen", "status": "#ff0000",
          "children": [
            {"name": "Lowland", "status": "#00c853"},
            {"name": "Akkin", "status": "#00c853"},
            {"name": "Galanchozh", "status": "#00c853"},
            {"name": "Itum-kali", "status": "#00c853"},
            {"name": "Kisti", "status": "#00c853"},
            {"name": "Cheberloj", "status": "#00c853"},
            {"name": "Sharoj", "status": "#00c853"}
          ]},
          {"name": "Ingush", "status": "#ff0000"},
          {"name": "Batsbi", "status": "#ff0000"}
        ]
      },
```

Figure 2: A snippet of the json-file

Figure 2 demonstrates the code of the json-file. In line 9, the node with the name "nakh" (in technical terminology, in this case, the attribute "name" with the value "nakh") has the parent node "nakh-daghestanian" in line 6 and child nodes with the names "Ingush" and "Batsbi" in lines 21 and 22, respectively. The "status" attributes contain RGB HTML colour codes in their values. For example, "name":"Chechen","status":"#ff0000" means that the node "Chechen" has the color #ff0000, which means red.

The code for parsing the json file also needed to be changed: the mechanism for parsing the json file and displaying it in the diagram was rewritten (see Figure 3).

```
93    nodeEnter.append("circle")
94        .attr("r", 1e-6)
95        .style("stroke", function(d) { return d.status; });
96
97    nodeEnter.append("circle")
98        .attr("r", 1e-6)
99        .style("fill", function(d) { return d._children ? "lightsteelblue" : "#fff"; });
100
101   nodeEnter.append("text")
102       .attr("x", function(d) { return d.children || d._children ? -13 : 13; })
103       .attr("dy", ".35em")
104       .attr("text-anchor", function(d) { return d.children || d._children ? "end" : "start"; })
105       .text(function(d) { return d.name; })
106       .style("fill-opacity", 1e-6)
107       .append("tspan")
108         .attr("x", function(d) { return d.children || d._children ? -13 : 13; })
109         .attr("dy", "1.5em")
110         .attr("text-anchor", function(d) { return d.children || d._children ? "end" : "start"; })
111         .text(function(d) { return d.description; })
112         .style("fill-opacity", "1");
113
```

Figure 3: A sample code for parsing the json-file

Figure 3 shows a fragment of parsing the json-file and building the diagram. For example, **return d.status** means getting the value of the status attribute (which stores the node colour), and return **d.name** - getting the value of **name** attribute (which stores the node name).

The second scheme represents a language family as circles packed in one another (see Figure 4). The nested circles correspond to the levels of genealogical classification, and the intensity of the colour depends on the nesting depth. The lowest levels are marked in white.

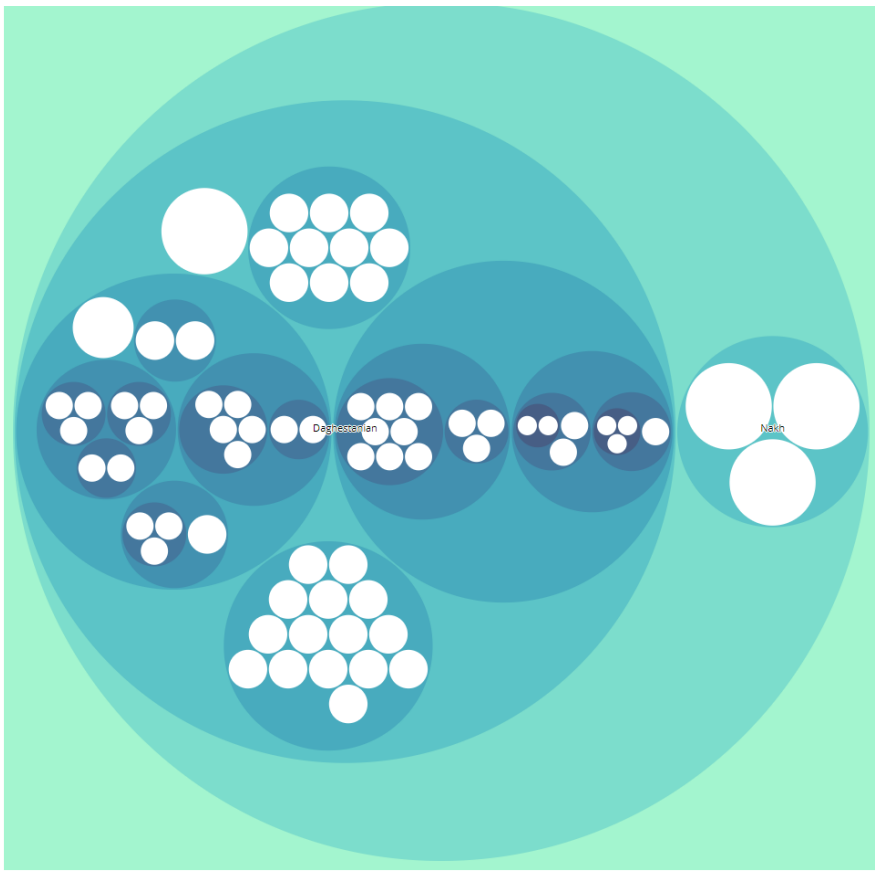

Figure 4: The Nakh-Daghestanian language family

Technically, this scheme is implemented by using the js-library D3 Zoomable Circle Packing. The data are packed in a json file with the appropriate structure. Then, js parses the json file and builds a diagram.

The advantage of this scheme is that it allows to see the whole family in one picture, to get the idea of its size and complexity. Nevertheless, for the moment the choice was made in favour of the tree model, because the default characteristics of the second scheme do not meet some of our goals. In particular, it is not possible to see the names of all languages at once. And what is even more important we cannot mark dialects, groups of dialects and languages with different colours (colours in this scheme are assigned automatically according to the nesting depth of a circle), and a suitable solution for this problem has not been found yet. The modification of this diagram in accordance with our tasks is the subject of further work.

2. Representation of areal distribution

An important part of language description is the representation of the areal distribution. There are two ways of displaying this area on the map: with polygons (we paint over the territory where the language is spoken) or with markers (we mark the settlements where the language is spoken). In the future, we intend to use both options: markers - for languages with a small number of local variants, polygons - for languages with a large number of local variants. Since we started with languages with a small number of local variants, at the moment the second option with some modifications is represented on the language page.

We saw our task here in the following: visualization of distribution of the local variants in the settlements; easy usage; adequate, but at the same time compact display of the data.

To achieve these goals we developed an interactive map where the settlements are marked according to the local variants of the language that are spoken there (see Figure 5). Each dialect gets its colour (see two blue markers at the top of the map for Bezhta Proper dialect in Figure 5). The values of the colours are given in the legend under the map. For settlements where more than one dialect is spoken a special multicoloured marker was introduced (see the marker on the right side of Figure 5). When a marker is clicked, a list of dialects appears in a pop-up window (see Figure 6). When zoomed out, several points on the map are combined into one cluster marked with a number that indicates the amount of settlements within the radius of this cluster (see the green marker in the center of Figure 5). This makes the presentation of the data more compact.

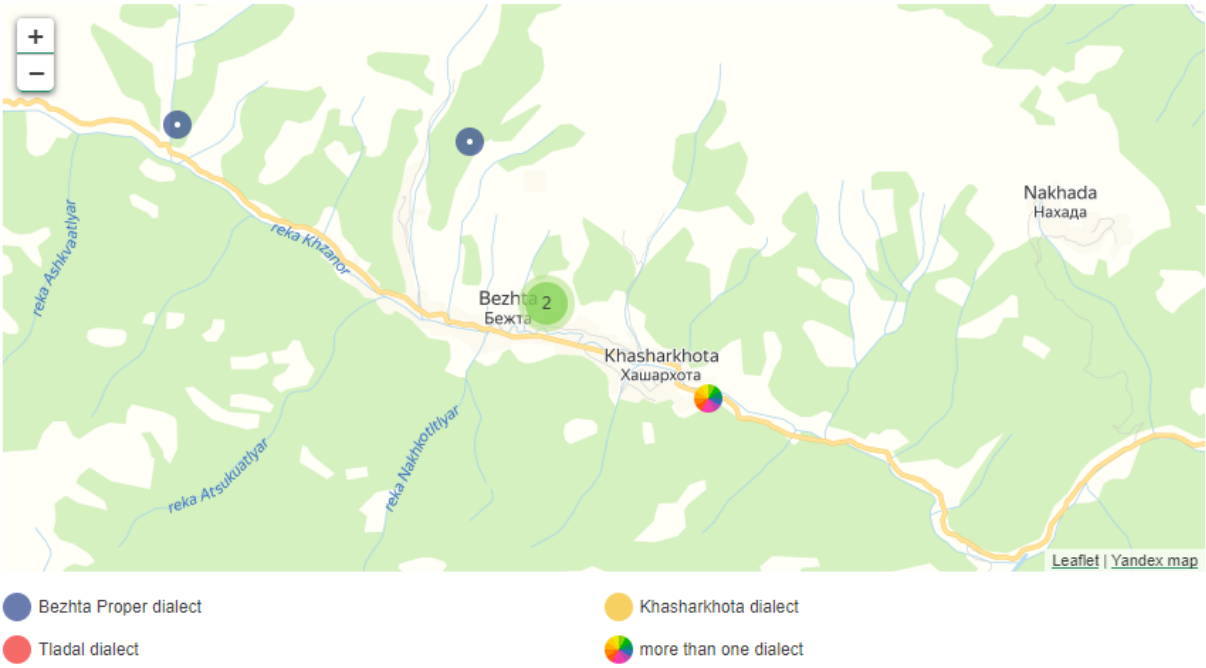

Figure 5: A fragment of the territory where Bezhta is spoken

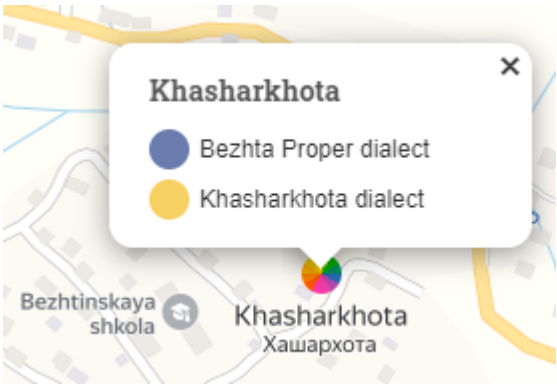

Figure 6: A window with a list of dialects that pops up when you click on a special multicoloured marker of a settlement

The data are represented on the map using the leaflet JavaScript library. Leaflet is a library that has proved its capability in a large number of resources. It provides great opportunities for working with geodata. Leaflet enables one to operate with any type of data (markers, polygons, lines), to use any substrates (OpenStreetMap, Yandex, GMap, etc.), and to customize the appearance of the markers and polygons displayed on the map according to one's needs. Leaflet can be extended with plugins, for example, for clustering data or for attaching json files with a database. Unlike Mapbox (a paid platform for working with GIS), there are no traffic restrictions.

The geographical coordinates of the settlements are pre-packed in an array with the following properties: settlement name, dialect, and dialect code (see Figure 7). Then the array is output to the map with marker clustering.

```
var langPoints2 = [
[42.1322,46.1361, '<h3>Bezhta</h3> <p>Bezhta Proper</p>', '1' ],
[43.7065,46.7132, '<h3>Kachalaj</h3> <p class="c1">Bezhta Proper</p><p class="c2">Khasharkhota</p>', 'm'],
[41.910286,45.915684, '<h3>Chantliskuri</h3> <p>Bezhta Proper</p>', '1'],
[41.901944,45.998056, '<h3>Shorokhi</h3> <p>Bezhta Proper</p>', '1'],
[41.7521,46.4502, '<h3>Kabakhchol</h3> <p>Bezhta Proper</p>', '1'],
[42.127889,46.149028, '<h3>Khasharkhota</h3> <p class="c1">Bezhta Proper</p><p class="c2">Khasharkhota</p>', 'm'],
[42.12267,46.23664, '<h3>Tladal</h3> <p class="c3">Tladal</p>', '3'],
[43.899722,46.71, '<h3>Vpered</h3> <p>Bezhta Proper</p>', '1'],
[43.777117,46.654361, '<h3>Rybalko</h3> <p>Bezhta Proper</p>', '1'],
[43.6875, 46.662778, '<h3>Karauzek</h3> <p class="c1">Bezhta Proper</p><p class="c3">Tladal</p>', 'm'],
[43.922778,46.561111, '<h3>Aleksandro-Nevskoe</h3> <p class="c1">Bezhta Proper</p><p class="c3">Tladal</p>', 'm'],
[44.017189,46.823258, '<h3>Malaja Areshevka</h3> <p class="c1">Bezhta Proper</p><p class="c3">Tladal</p>', 'm'],
[43.762561,46.586383, '<h3>Zarechnoe</h3> <p class="c1">Bezhta Proper</p><p class="c3">Tladal</p>', 'm'],
[43.816408,46.724056, '<h3>Yuzhnoe</h3> <p class="c2">Khasharkhota</p><p class="c1">Bezhta Proper</p>', 'm'],
[42.1526,46.0849, '<h3>Zhamod</h3> <p>Bezhta Proper</p>', '1' ],
[42.1388,46.134, '<h3>Isoo</h3> <p>Bezhta Proper</p>', '1' ],
[42.1511,46.1203, '<h3>Balakuri</h3> <p>Bezhta Proper</p>', '1' ],
[43.192,46.9847, '<h3>Shushanovka</h3> <p>Khasharkhota</p>', '2' ],
[43.9687,47.3765, '<h3>Krajnovka</h3> <p>Tladal</p>', '3' ],
[43.1778,46.9683, '<h3>Stalskoe</h3> <p>Khasharkhota</p>', '2' ]
];
```

Figure 7: A code snippet with the data array

The array elements correspond to the following model: [{latitude Coordinate}, {longitude Coordinate}, {pop-up window content}, {dialect number (more than one dialect is indicated by the letter m)}].

Figure 8 shows a fragment of the data output to the map with comments on some functions.

```
81      coords2.push(new L.LatLng(a[0], a[1])); // add coordinates (markers) to array
82      langPoint2.bindPopup(a[2]); // content for popup dialog
83      markerscluster2.addLayer(langPoint2); // markers clustering
84    }
85
86    map2.fitBounds(coords2, {
87        padding: [5, 5] // add padding
88    });
89
90    map2.addLayer(markerscluster2); // output data to map
```

Figure 8: A code snippet of the data output to the map

3. Representation of the domains of language usage

An important component of the sociolinguistic description of a language is a list of domains in which it functions. Thus, the volume and nature of the language use in these domains allow one to judge the vitality of the language. A list of 16 core domains was compiled, starting with family communication and education and up to the language use on the Internet.

When choosing a method for representing the domains, the following tasks were primarily solved: visualization of a complete picture of the domains within a single user screen; convenience and speed of obtaining the information.

This has been achieved by drawing up an interactive table of sixteen cells (see Figure 9), corresponding to the domains of the language usage. For ease of viewing and perception, the table is structured as follows: the cells of the table contain the titles of the domains, and it is noted whether the language is used in this domain or not (see the “empty or filled circle” marker in the lower right corner of each cell in Figure 9). When clicking on a cell, a pop-up window opens with expanded information about the peculiarities of the language use in this domain (see Figure 10).

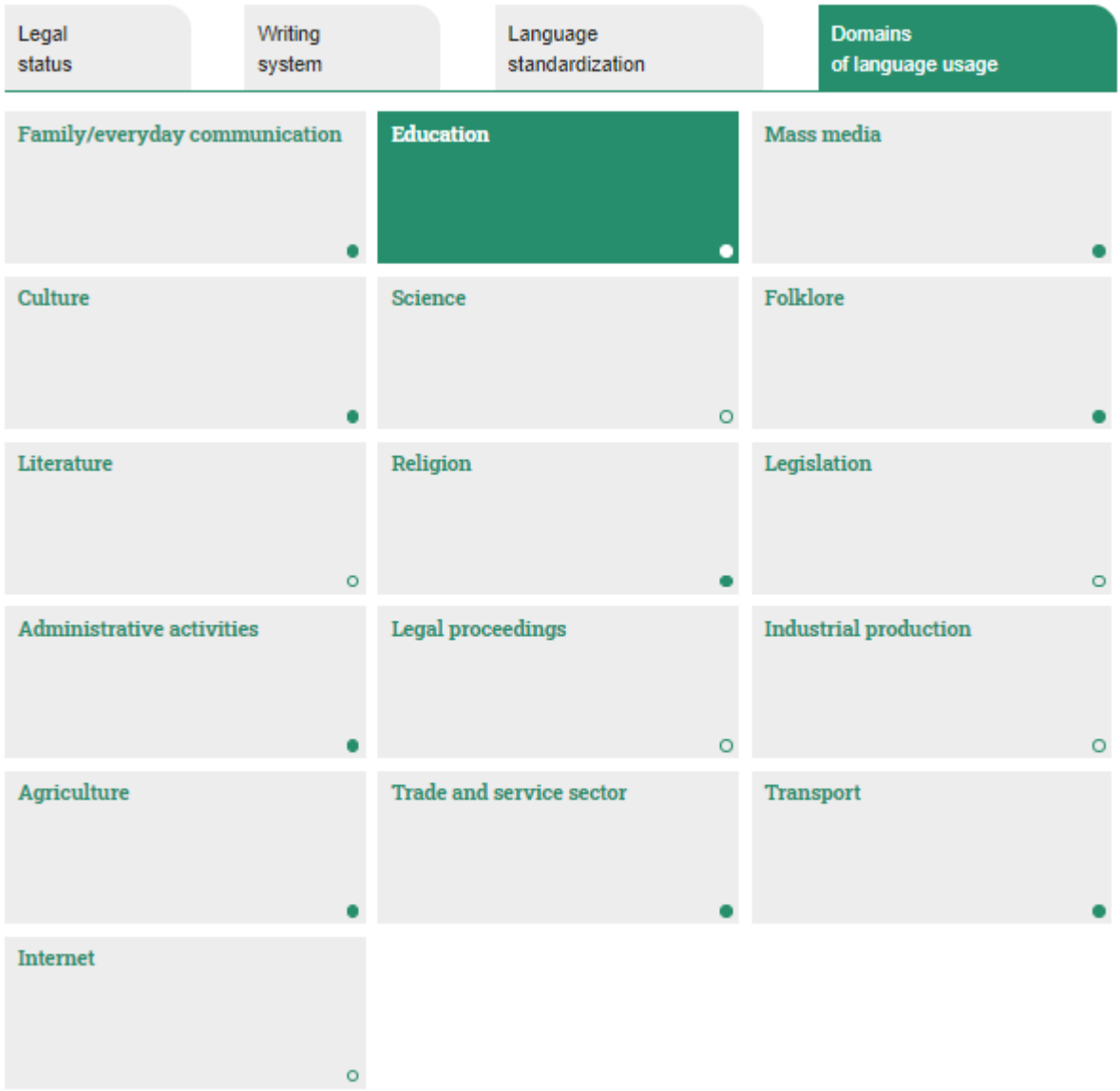

Figure 9: Domains of language usage in Bezhta

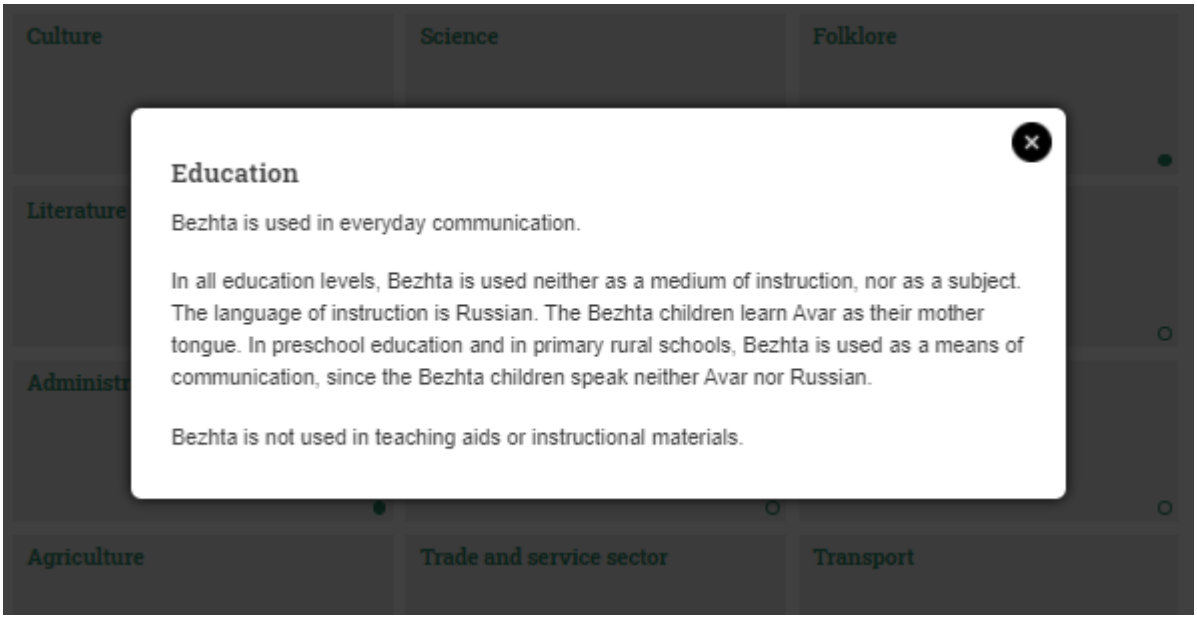

Figure 10: A pop-up window describes the use of Bezhta in religion

The "Domains of language usage" section is marked with HTML tags in accordance with the intended structure (see Figure 11). The CSS style is responsible for the graphic design (see Figure 12). JavaScript provides a mechanism for opening and collapsing a pop-up window.

```
<div id="tab-4" class="tab-content current">
  <h4 class="readmore-header is-info" id="link-ex2">Family/everyday communication</h4>
  <div class="modal" id="ex2">
    <h4>Family/everyday communication</h4>
    <p>Bezhta is used in everyday communication.</p>
  </div>
  <h4 class="readmore-header is-info" id="link-bez01">Education</h4>
  <div class="modal" id="bez01">
    <h4>Education</h4>
    <p>Bezhta is used in everyday communication.</p>
    <p>In all  education levels, Bezhta is used neither as a medium of instruction,
    nor as a subject. The language of instruction is Russian. The Bezhta children
    learn Avar as their mother tongue. In preschool education and in primary rural
    schools, Bezhta is used as a means of communication, since the Bezhta children
    speak neither Avar nor Russian.</p>
    <p>Bezhta is not used in teaching aids or instructional materials.</p>
  </div>
  <...>
</div>
```

Figure 11: A snippet of the code markup

As you can see in Figure 11, the **h4** element in the **class** attribute has the value **is-info**, which means that the language is used in this area. Otherwise, the **class** attribute is set to **no-info**.

```css
.readmore-header {
    display: block;
    width: 32%;
    min-height: 120px;
    line-height: 25px;
    cursor: pointer;
    background: #ededed;
    float: left;
    margin: 0 10px 10px 0 !important;
    padding: 5px 10px;
    font-size: 16px;
    transition: background-color .3s linear;
    color: rgba(38, 142, 108,1);
    position: relative;

}
.readmore-header:hover {
    color: #fff;
    background: rgba(38, 142, 108,1);
}

.readmore-header.is-info:after {
    display: block;
    content: '';
    position: absolute;
    right: 5px;
    bottom: 5px;
    width: 10px;
    height: 10px;
    background: rgba(38, 142, 108,1);
    border-radius: 50%;
    border: solid 2px rgba(38, 142, 108,1);
}

.readmore-header.is-info:hover:after {
    background: #fff;
    border: solid 2px #fff;
}
```

Figure 12: A sample snippet of the CSS-style for the table

Figure 12 shows some properties for the elements of the **readmore-header** class. For example, the background property is set to **#ededed**, which means that the blocks of the **readmore-header** class have a gray background. The **hover** pseudo-class is responsible for the style that is applied when the mouse pointer hovers over a table cell.

4. Representation of the dynamics of language usage

For our internet resource dedicated to minority languages, it turns out to be critical to track the change in the size of the ethnic group and the number of speakers over time in the case of each particular language. This section of the language page currently accumulates the data from the population censuses of various years, and in the future, we hope to represent the data from local administrations and researchers' estimates.

Thus, the main task was to visualize the general picture of dynamics, which allows tracking the change of various quantitative indicators in time.

To solve this problem, an interactive diagram was developed. It displays the change in the size of the ethnic group, the number of group members who consider the ethnic language of the group to be their mother tongue, and the number of those who reported proficiency in this language (see Figure 13).

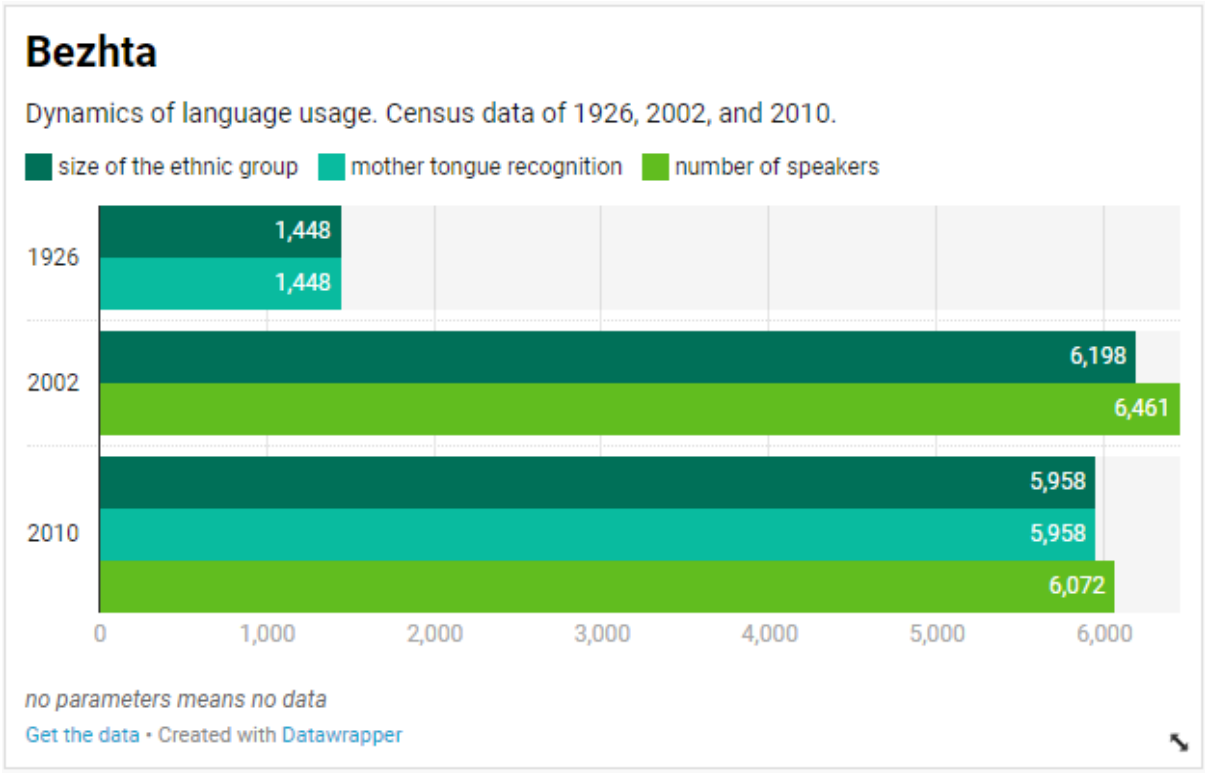

Figure 13: Dynamics of the Bezhta language usage

A simple charting tool - datawrapper.de - was used. To submit the data, one just needs to register on the site datawrapper.de, to select the desired widget, to enter the data in a specific format (the format depends on the selected template, see Figures 14-15). Then one gets the embed code of the widget (see Figure 16) and inserts it on the site.

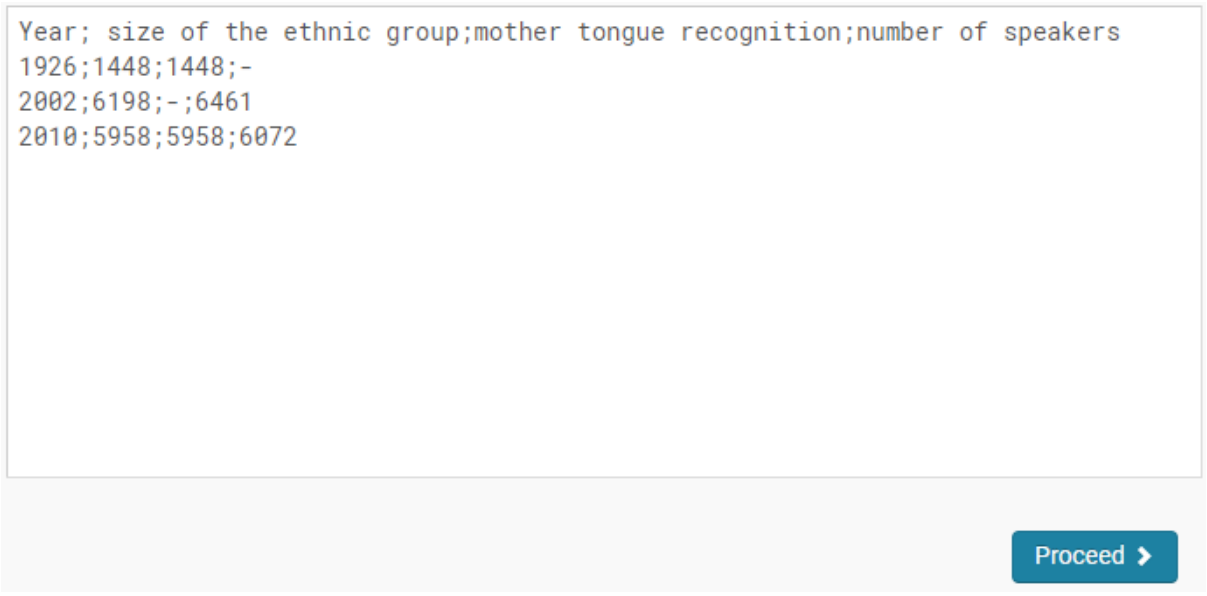

Figure 14: An example of the data entry

	A	B	C	D
1	Year	size of the ethnic group	mother tongue recognition	number of speakers
2	1926	1,448	1,448	-
3	2002	6,198	-	6,461
4	2010	5,958	5,958	6,072

Figure 15: An example of the data entry

```
<iframe title="Bezhta" aria-label="Grouped Bars" id="datawrapper-chart-x2Z5S"
src="https://datawrapper.dwcdn.net/x2Z5S/1/" scrolling="no" frameborder="0"
style="width: 0; min-width: 100% !important; border: none;" height="374"></
iframe><script type="text/javascript">!function(){"use strict";window.
    addEventListener("message",(function(a){if(void 0!==a.data["
    datawrapper-height"])for(var e in a.data["datawrapper-height"]){var t=
    document.getElementById("datawrapper-chart-"+e)||document.querySelector("
    iframe[src*='"+e+"']");t&&(t.style.height=a.data["datawrapper-height"][e]+"
    px")}}))}();
</script>
```

Figure 16: The output embed-code

One can also customize the appearance of the chart using the built-in tools. For example, it is possible to change the colours (see Figure 17), fonts, sizes, and the contents of the text fields.

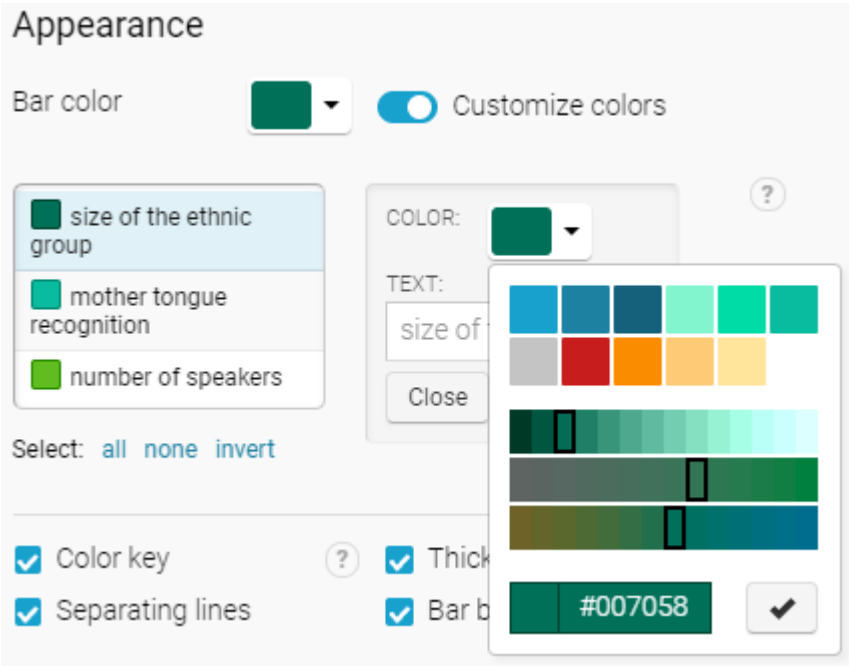

Figure 17: An example of the control colour chart

5. Representation of phonetic data

'Phonetic data' usually refers to an array of systems, describing both acoustic/articulatory phenomena and phonology.

A coordinate plane, which is determined by the first and the second formants, or a trapezoid vowel diagram are often used for representation of a sound system of vowels. However, for phonological vowel systems a table structure is more relevant, as it enables to describe several differential features apart from height and backness. As for consonant sound systems, they are traditionally provided in tables: it is possible to draw a line between different types of features without data loss in this case.

Language description implies a phonology description and a presentation of its sound manifestations. Our goal is to combine the acoustic and the articulatory features with the phonological representation in order to make our description compact and easy to analyze without special work with audio data.

We primarily aimed at a united representation of phonetics and phonology, alongside with user-friendliness, objectivity and descriptive representativeness.

As a solution, we have constructed graphic representations for vocalism (see Figure 18) and consonantism (see Figure 19) that reflect a phonological system with its correct sound manifestations. Each sound is transcribed according to the IPA[1], and the whole classification is synchronized with Russian terms. An optimal representation of the structure within universal consonant and vowel tables has been elaborated, in order to facilitate the comparison of languages.

Markus, Rozhanskij [2017]

Figure 18: Votic vowel system

1This resource is available at: https://www.internationalphoneticassociation.org/content/full-ipa-chart.

Forest Enets consonant system

		place of articulation									
		labial		prepalatal				palatal		velar	
				dental/alveolar		postalveolar					
manner of articulation	plosive	p	b	t	d				ɟ	k	g
	nasal		m		n				ɲ		ŋ
	trill				r						
	affricates					t͡ʃ					
	fricative		β	s	z	ʃ			j		x
	lateral approximants								ʎ		

Colour marking:

voiceless
aspiration
glottalization
voicing

Figure 19: Forest Enets consonant system

The representation of vocalism is based on filling out an initial matrix with height values in strings and backness values in columns. We provided about two or three slots for each traditional element of classification (height/backness), as the same phoneme in the classification may display different spectrum in various languages. This solution is crucial both for language comparison and for the reflection of phonological processes.

In addition to height and backness, there are secondary articulations significant for phonology: vowel length, labialization, nasalization and pharyngealization. For each characteristic there exists an additional string or a column, which tentatively accounts for the articulation manner and its impact on the spectrum. Thus, pharyngealized vowels are placed in columns to the right of the unpharyngealized ones, and nasal vowels are placed in strings below the non-nasal ones. The length of the vowel is marked when it is relevant: long vowels take place closer to the cardinal locus than their short pairs. The recited secondary articulations are marked by diacritics, while labial vowels have their own symbols in the IPA. Traditionally labials are placed to the right of their illabial pairs.

If the language's vowel system lacks nasalization/pharyngealization/length, the corresponding table cells are removed from the initial matrix. Primary strings and columns (based on height and backness) are preserved, empty cells are visualized as an empty space. It is necessary to distinguish the phonologically similar vowels with different acoustic features in different languages. Then a trapezoid is superimposed on the table: phonemes with the same height value adjoin to its horizontal contours, while phonemes with the same backness value adjoin to its vertical contours, so that each line corresponds to phonologically meaningful height/backness.

The visual representation of vocalism is created by means of HTML markup stylized in a certain way, and a graphic figure (trapezoid) superimposed on this markup (see Figures 20-21). Technically, this is arranged as follows: an HTML container is created, and two more containers (layers) are created inside it. The first container includes the marked up data (characters), and the second container includes an image with a graphic figure. Then, using CSS, the characters are given an exact position towards the upper-left edge of the container, so that they reach the desired positions towards the superimposed graphics.

```
.voc-data .v-el-2,
.voc-data .v-el-4,
.voc-data .v-el-6,
.voc-data .v-el-7 { left: 600px; }

.voc-data .v-el-1,
.voc-data .v-el-2 { top: 35px; }
.voc-data .v-el-3,
.voc-data .v-el-4 { top: 130px; }
.voc-data .v-el-5,
.voc-data .v-el-6 { top:332px; }

.voc-data .v-el-1 { top: 35px; left: 50px; }
.voc-data .v-el-3 { left: 114px; }
.voc-data .v-el-5 { left: 139px; }
.voc-data .v-el-7 { top:437px; }
```

Figure 20: CSS style positioning of the elements (vocalism)

```
▼<div class="voc-data">
    <img src="vowelBg6.png" alt width="541" height="440">
    <span class="v-el-1">i</span>
    <span class="v-el-2">u</span>
    <span class="v-el-3">e</span>
    <span class="v-el-4">o</span> == $0
    <span class="v-el-5">ɛ</span>
    <span class="v-el-6">ɔ</span>
    <span class="v-el-7">ɑ</span>
```

Figure 21: A fragment of the HTML-structure (vocalism)

The universal table of consonants is defined by the place and manner of articulation as columns and strings respectively. Secondary articulations refer to the place of articulation. In particular, labialization or pharyngealization splits each column into additional subcolumns. Labialized consonants are placed to the left of the neutral column, while pharingalized ones are placed to the right of it. Palatalization is described as follows: subcolumns are usually added to the right of the initial column in case of labial and coronal consonants, while in case of dorsal consonants they are added to the left. Abruptives are put in substrings below each string for pulmonic consonants. The last splitting of columns is done according to the state of the vocal fold (4 subcolumns: voicelessness, aspiration, glottalization and voicing). This level of organization has colour marking.

After the initial universal table is filled out, strings and columns with empty cells are removed from it. Therefore, we end up with a compact table describing the consonant sound and phonological system of a given language.

The visual representation of consonantism is created via HTML markup and table layout applying the CSS styles (see Figures 22-23). In particular, the CSS is used to detect **voicelessness**, **aspiration**, **glottalization**, and **voicing**, according to the specified classes in the HTML structure.

```
.feature-1 { background: #d9ead3; }
.feature-2  { background: #b6d7a8; }
.feature-3 { background: #93c47d; }
.feature-4 { background: #6aa84f; }
```

Figure 22: A fragment of the CSS style (consonantism)

```
▼<div class="table-container"> == $0
  ▼<table class="konsonantizm table is-bordered scrollable">
    ▼<tbody>
      ▶<tr>…</tr>
      ▶<tr>…</tr>
      ▼<tr>
        ▶<td rowspan="5" class="left-label">…</td>
          <td class="left-sublabel">взрывные</td>
          <td class="is feature-1">p</td>
          <td class="is feature-4">b</td>
          <td class="is feature-1">t</td>
          <td class="is feature-4">d</td>
          <td class="is feature-1">c</td>
          <td class="is feature-4">ɟ</td>
          <td class="is feature-1">k</td>
          <td class="is feature-4">g</td>
          <td class="is feature-1">(q)</td>
          <td class="is feature-4">(G)</td>
        </tr>
      ▶<tr>…</tr>
      ▶<tr>…</tr>
      ▶<tr>…</tr>
      ▶<tr>…</tr>
    </tbody>
  </table>
</div>
```

Figure 23: A fragment of the HTML-structure (consonantism)