Visualizing Russian kinship term possessive sequences as family trees

Anna Golub[1] and Alyona Belova[1]

Faculty of Infocommunication Technologies, ITMO University, St. Petersburg, Russia
{anna.golub.923 | belova.alyona.itmo}@gmail.com

Abstract. As frequently as they are encountered in texts of various genres, Russian kinship term possessive sequences remain confusing even for the native speakers. The paper presents the authors' original computer science project, whose goal was to suggest a method of extracting such word sequences from a piece of text and visualizing them in an easily comprehensible way. Such an attempt of kinship relations analysis automatization might contribute to future research in history, linguistics and literary studies and be of use to those studying Russian as a foreign language.

Keywords: kinship term · possessive structure · family tree · natural language processing · Russian.

List of Russian Terms

я first-person singular pronoun
мой, моей first-person possessive pronoun forms
мать mother
отец father
сестра sister
муж husband
сын son
бабушка grandmother
тёща mother-in-law, wife's mother

1 Introduction

Russian possessive sequences including several kinship terms often appear confusing in written text as well as in oral speech since it is difficult to quickly calculate the relations between the characters mentioned. Besides, such phrases often include names of relatives by marriage, which are becoming increasingly obsolescent in modern Russian as they only constitute for 1.4 per cent of all kinship term entries in Russian National Corpus [1] in 1990—2020. Consequently, marital kinship terms further complicate human comprehension of the sequences.

See an illustrative example of the sequence in question in (1).

(1) сестра мужа моей тёщи
sister.NOM.SG husband.GEN.SG my.F.GEN.SG wife's_mother.GEN.SG
'the sister of my wife's mother's husband'

Thus, the goal of the authors' original computer science project was to write a computer program that would extract the sequences in question from a given piece of text and visualize them in a way easily comprehensible by humans. Such an attempt of kinship relations analysis automatization might contribute to future research in history, linguistics, and literary studies, e. g. scientific analysis and systematization of fiction and memoirs. Moreover, this technology might be utilized by those studying Russian as a foreign language in order to ease the process of Russian kinship terms' meaning comprehension and memorization.

2 State of the Art

2.1 Kinship Term Possessive Sequences

As to our knowledge, a comprehensive solution to the problem has not yet been suggested. A great deal of research has been done on kinship term systems and the grammar of possession in various languages. However, possessive structures with kinship terms have not been subject to in-depth investigation. A few researchers touched on the topic of kinship terms in their studies of possessive structures (Dahl & Koptjevskaja-Tamm 2001 [2], Paykin & Van Peteghem 2003 [3], Jones 2010 [4]). However, the approach taken was exclusively theoretical, and in addition, the only type of phrases discussed was those consisting of a possessive pronoun and a single kinship term (e.g. *her sister*). Family relationships have also been looked into in studies of social relations extraction (Makazhanov et. al 2014 [5]), but phrases including multiple kinship terms yet have not been profoundly covered. Aside from that, some papers describing data collected through fieldwork mention kinship term possessive sequences, but that kind of research does not seem applicable to building a tool for their computational processing.

2.2 Family Tree Visualization

In turn, a large number of software tools are suitable for family tree visualization. Both specialized packages (e.g. ggenealogy [6]) and libraries aimed at visualization in general (Graphviz [7], Plotly [8], Toytree [9]) provide plotting methods for genealogical data. However, the hierarchical nature of a genealogical structure does not allow horizontal edges and node skipping, which is required for valid representation of relationships in kinship term possessive sequences. Furthermore, the tools mentioned have rather limited customization options while the goal was to display confusing lineages in the most efficient way in regard to human visual comprehension. As to software for intuitive family tree building (e.g. Family Historian [10]), despite representing pedigrees in a visually organized manner, those applications require user interaction, which complicates automatic visualization.

3 Solution

3.1 Text Search

All the source code is written in Python. At first, using NLTK [11], the given piece of text is split into sentences; then each of them is tokenized into words. Next, employing pymorphy2 [12] for part-of-speech tagging and further morphological analysis, continuous word sequences are extracted from sentences. At this point the sequences consist of:

- one or more kinship terms, the first one of them in any case while all the rest in the genitive exclusively;
- certain kinds of modifiers, namely long- or short-formed adjectives and participles, ordinal numerals and adjective pronouns;
- not more than one non-kinship noun in the genitive case. If included, this word is the last one in the sequence (see sequence type 4 below).

Afterwards, each of the sequences is recognized as one of the sequence types listed below, which depict most instances of kinship term possessive sequences in Russian.

1. **possessive adjective** / **possessive pronoun** + **kinship term**

   (2) бабушкиному мужу
   grandmother.POSS.M.DAT.SG husband.DAT.SG
   ‘to (the) grandmother’s husband’

2. **kinship term** + **kinship term** (GEN) (n=0,1,2...) +
   + **possessive adjective** / **possessive pronoun** + **kinship term** (GEN)

   (3) муж бабушки моей сестры
   husband.NOM.SG grandmother.GEN.SG my.F.GEN.SG sister.GEN.SG
   ‘the husband of my sister’s grandmother’

3. **kinship term** + **kinship term** (GEN) (n=0,1,2...) +
   + **possessive adjective** / **possessive pronoun**

   (4) муж бабушки сестры моей
   husband.NOM.SG grandmother.GEN.SG sister.GEN.SG my.F.GEN.SG
   'the husband of my sister's grandmother'

4. **kinship term** + **kinship term** (GEN) (n=0,1,2...) +
   + **noun** (non-kinship) (GEN)

   (5) мужем бабушки подруги
   husband.INS.SG grandmother.GEN.SG friend.GEN.SG
   'by the husband of a friend's grandmother'

5. **kinship term** + **kinship term** (GEN) (n=0,1,2...)

   (6) мужу бабушки сестры
   husband.DAT.SG grandmother.GEN.SG sister.GEN.SG
   'to the husband of (the) sister's grandmother'

Then, the sequence is cast to the so-called normal form:

- kinship terms are put in the nominative singular form;
- non-kinship nouns are put in the nominative case with the number unchanged;
- possessive adjectives are replaced with their stem nouns in the nominative singular form;
- possessive pronouns are replaced with the corresponding personal ones in the nominative case.

Afterwards, the sequence is reshuffled so as to put the words in the direct relation order (see examples in (7)). If the sequence then starts with a kinship term, a first-person singular pronoun is inserted into the beginning.

(7) a. *бабушкиному мужу — я бабушка муж*
бабушкиному мужу
grandmother.POSS.M.DAT.SG husband.DAT.SG
'to (the) grandmother's husband'

я бабушка муж
1SG.NOM grandmother.NOM.SG husband.NOM.SG
'me grandmother husband'

b. *мужу бабушки моей сестры — я сестра бабушка муж*

мужу бабушки моей сестры
husband.DAT.SG grandmother.GEN.SG my.F.GEN.SG sister.GEN.SG
'the husband of my sister's grandmother'

я сестра бабушка муж
1SG.NOM sister.NOM.SG grandmother.NOM.SG husband.NOM.SG
'me sister grandmother husband'

Thus, all the words in the sequence, except for the first one, turn out to be kinship terms in the nominative singular form. Preprocessed this way, the sequence is fit for further analysis.

3.2 Kinship Relations Analysis

For each word in the sequence, except for the first one, the following actions are performed. First, a graph fragment template is uploaded, which presents the relationship between this and the previous character in the sequence. Within the template all the characters are connected directly, either as **parent** — **child** or **wife** — **husband**. See Fig. 1 as an example.

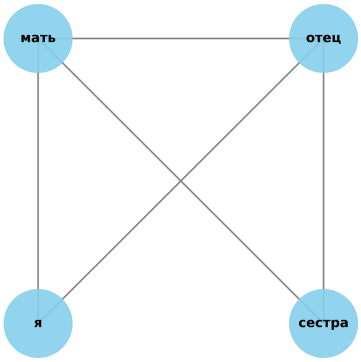


Fig. 1. Сестра (sister) template

Next, the template is incorporated in the graph that has been built so far, namely the root of this template is aligned with the top of the previous one. Here, by the template top we mean the character signified by this template's corresponding word. In turn, the template's root is the character whose relation to the top of the template is being described by the template word. Consequently, by the graph root (respectively, top) we mean the root (respectively, top) of the first (respectively, last) template included in the graph. See Fig. 2: the *бабушка* (*grandmother*) template being aligned with the *сестра* (*sister*) template. (For now, we only discuss the maternal grandmother.)

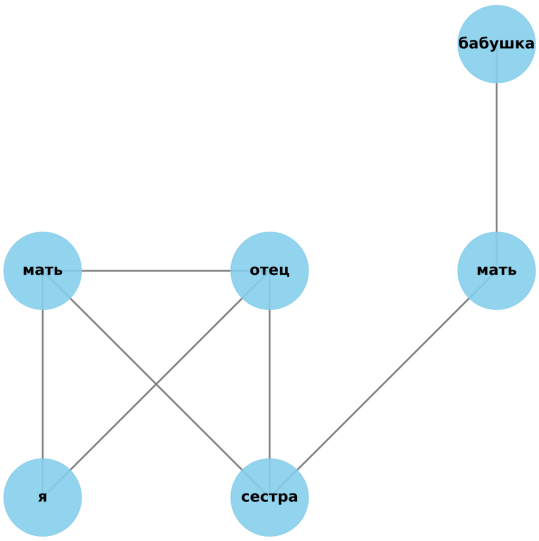


Fig. 2. Template alignment

Finally, if multiple nodes of the graph correspond to the same character of the sequence, they are merged into one. Such cases are unraveled based on the heuristic that each character may only have one mother and one father. The example graph structure is thus transformed into the one in Fig. 3.

Thus, in the resulting graph all the characters are connected directly to each other. It is worth mentioning that due to linguistic polysemy, the relationship reflected in a kinship term might be described by a few different templates (e.g. in the above example the grandmother might be both maternal and paternal). Consequently, several graph structures are built considering all the possible template combinations for the kinship terms in the sequence.

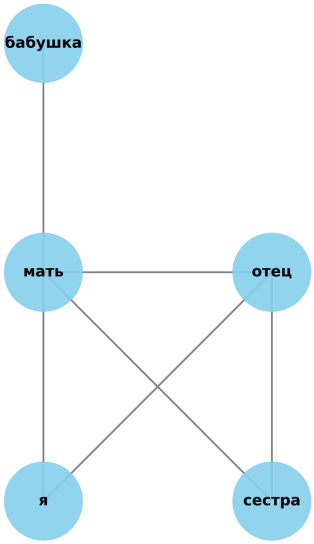


Fig. 3. The final graph structure

3.3 Graph Visualization

Family trees are drawn using NetworkX [13] and Matplotlib [14]. All the edges of the graph are added to the list in a loop that goes through the characters and their connections. The root node gets zero coordinates; for other nodes, the following rules apply:

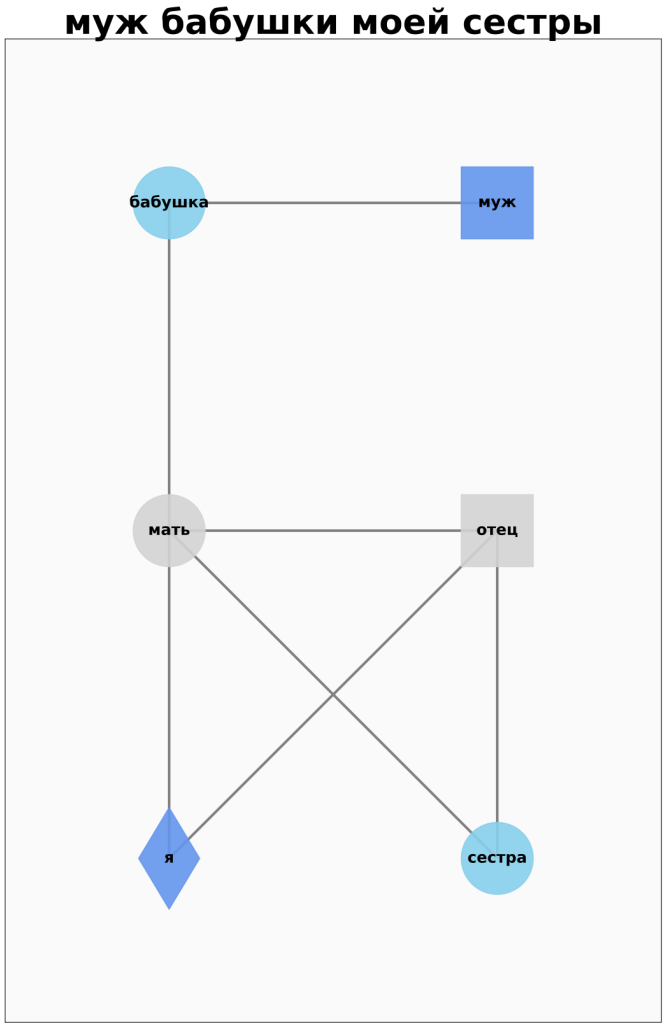
- a parent is positioned one point higher than their child;
- a child is positioned one point lower than their parent;
- a wife/husband is drawn one point to the right from their spouse;
- if the determined spot is already taken, the node is shifted one point to the right.

As to colours, the following rules apply:

- if the character is directly mentioned in the kinship term sequence, the node has a light blue colour;
- if there is no direct mention of the character, the node is painted light gray;
- darker blue is used for the graph top and root nodes.

Gender is displayed through the shape of the nodes: circles for females, squares for males; a rhombus is used for the root node. As a result, a PNG file presents the graph with a gray background, the kinship term sequence and the original sentence where it was included. As an example, see Fig. 4 which shows the final graph visualization of the sequence *муж бабушки моей сестры* from the sentence in (8). Fig. 4 is the program output.

(8) Я знал его с детства — это был *муж бабушки моей сестры*.
‘I had known him since my early years. He was *the husband of my sister’s grandmother*.’



Я знал его с детства — это был муж бабушки моей сестры.

Fig. 4. The program output

4 Evaluation

4.1 Evaluation Process

In order to evaluate the tool's performance, it was run on a purposefully combined corpus of texts, selected manually from the Russian National Corpus [1]. The corpus consists of 3067 words and includes at least five entries of each of the kinship terms while keeping a rough balance between the sequence types. For text search evaluation, each kinship term possessive sequence in the corpus was manually classified as follows:

- True Positive — the sequence was found by the program, and its boundaries were identified correctly;
- False Positive — the sequence was found, but extra words were included;
- True Negative — there are no sequences in the sentence, and none were found;
- False Negative — the sequence was found, but some necessary words were excluded.

The precision and recall scores were then calculated, turning out 0.96 and 0.93 respectively.

Afterwards, for each of the sequences found by the program, the graphs were drawn to evaluate kinship relations analysis and visualization. For each of the sequences the number of expected and present correct visualizations was estimated manually, with the resulting accuracy score being 0.95.

4.2 Discussion

The evaluation results are high enough to say that the tool serves its purpose adequately. Nevertheless, a few areas for future work may be suggested.

- Processing proper names. As for now, the tool cannot correctly process input sequences including first name + patronym collocations or their abbreviated forms.
- Broadening the range of sequence types. For example, currently the sequence in (9) cannot be processed correctly because it does not fit any of the sequence type schemas.

(9) бабушки моей муж
grandmother.GEN.SG my.F.GEN.SG husband.NOM.SG
'my grandmother's husband'

- Adding context analysis features in order to better differentiate between the sequence types.

- Updating the template approach. At the relations analysis stage, complex kinship terms can be replaced with their simpler explanations, e. g. turning *тёща* (*mother-in-law*) into *мать жены* (*wife's mother*), allowing to only store templates for the basic kinship terms, namely parents, children, siblings and spouses.
- Identifying coreference. At this point, the program does not register several words referring to the same character as in *сын моего отца* (*my dad's son*) and is unable to depict that in the graph.
- Adapting the tool to other languages. This would affect the text processing stage only since the relations analysis and visualization algorithms are language-independent.

5 Conclusion

This paper has approached kinship term possessive sequences as a field for natural language processing development, presenting the authors' original tool for the sequences' human-readable visualization.
The project source code is available on github ; the evaluation corpus may also be viewed there.

References

1. Russian National Corpus https://ruscorpora.ru/new/
2. Dahl, Östen & Koptjevskaja-Tamm, Maria. (2001). 11. Kinship in grammar. 10.1075/tsl.47.12dah.
3. Paykin, K., van Peteghem, M. External vs. Internal Possessor Structures and Inalienability in Russian. Russian Linguistics 27, 329–348 (2003).
4. Jones, Doug. (2010). Human kinship, from conceptual structure to grammar. The Behavioral and brain sciences. 33. 367-404
5. Makazhanov, Aibek & Barbosa, Denilson & Kondrak, Grzegorz. (2014). Extracting Family Relationship Networks from Novels.
6. Rutter L, VanderPlas S, Cook D, Graham MA (2019). "ggenealogy: An R Package for Visualizing Genealogical Data." Journal of Statistical Software, 89(13), 1–31.
7. "Graphviz and Dynagraph – Static and Dynamic Graph Drawing Tools", by John Ellson, Emden R. Gansner, Eleftherios Koutsofios, Stephen C. North, and Gordon Woodhull, in Jünger & Mutzel (2004).
8. Plotly Technologies Inc. Collaborative data science. Montréal, QC, 2015. https://plot.ly
9. Toytree package https://toytree.readthedocs.io/en/latest/
10. Best family tree makers 2021 https://www.toptenreviews.com/software/home/best-genealogy-software
11. Bird, Steven, Edward Loper and Ewan Klein (2009), Natural Language Processing with Python. O'Reilly Media Inc.
12. Korobov M.: Morphological Analyzer and Generator for Russian and Ukrainian Languages // Analysis of Images, Social Networks and Texts, pp. 320-332 (2015)

13. Aric A. Hagberg, Daniel A. Schult and Pieter J. Swart, “Exploring network structure, dynamics, and function using NetworkX”, in Proceedings of the 7th Python in Science Conference (SciPy2008), Gäel Varoquaux, Travis Vaught, and Jarrod Millman (Eds), (Pasadena, CA USA), pp. 11–15, Aug 2008
14. J. D. Hunter, "Matplotlib: A 2D Graphics Environment", Computing in Science & Engineering, vol. 9, no. 3, pp. 90-95, 2007.