CHAPTER X

BUILDING FAMILY TREES WITH NOOJ

KRISTINA KOCIJAN AND MARKO POŽEGA

Abstract

This paper proposes NooJ as a pre-processing tool for building an ontology based project i.e. building family trees from Croatian obituaries. The names of an ego (deceased) and his family members, extracted from the list of grieving family members, are mapped into specific family relationships and extracted in an XML-like format which is then sent to Python for hierarchical presentation.

Family relationships in Croatia

The words used to describe family relationships in Croatian language make quite a rich addition to Croatian kinship terminology since separate terms are used for family members (relationships created by birth) and in-laws (relationships created by marriage). The terminology also differs among different geographical areas of Croatia.

The main idea behind this project is to find which kinship terms are in use in Croatia today by using on-line obituaries, and to make a graphical representation of those terms for each deceased person.

Tanocki (1986) describes 1 699 distinct relationship terms. However, we found only 99 in the corpus we used. This is important information which implies that the "old" kinship terminology is being replaced with the less detailed one i.e. few terms are being replaced with one term. When comparing list of terms used in contemporary obituaries to that of Tanocki's (1986) kinship terms, we conclude that this change in terminology is only occurring at more distant levels of relationship, while

closer relationships (mother, father, son, daughter, sister, and brother) use the same terminology regardless of time or geographical region.

However, it is not the purpose of this paper either to explain the reason for this terminological shift or when and where it started. Jozić et al. (2010), Tikvica (2009) and Šokota (1998) have more to say on that subject. In this paper, the authors will only try define a list of kinship terms currently used in on-line obituaries in order to produce their hierarchical representation. NooJ is used for finding and annotating family relationships and Python for drawing family trees.

Both Pleše (1998) and Tanocki (1986) suggest that there is a difference between the normative and used terminology. This is surely the case of modern Croatian language as well and it is very likely that the terms used in writing obituaries are not the ones used in every day's communication by those who write obituaries. This is the reason why we do not hold our list of kinship terms as the final list of family relationships in contemporary Croatian language.

In a nutshell, our task is to extract information (personal names) from obituaries and fill the prepared slots (relation types) in order to present a system (family tree) that has an ontological nature (Biemann, 2005).

This is not the first use of NooJ as a preprocessing tool for an ontology related project (Salza, 2014). Although our topics differ, we both recognized NooJ (Silberztein, 2003) as a powerful tool for NLP.

The process for building our ontological framework can be described with the following four steps: preparing a corpus from Croatian obituaries; applying NooJ's syntactic analysis to recognize the kinship terms; annotating the text with the xml notation within NooJ; drawing the tree from an xml file with Python. We proceed with the more detailed description of each step.

Preparing the corpus

For our preliminary research, we build a small corpus of 44 obituaries published on-line on the obituaries site (http://www.osmrtnice.hr/) during the month of November 2013. This corpus was used as our training corpus for building of the initial syntactic grammars.

After we obtained 100% of both precision and recall on training corpus, we applied the grammar to two test corpora constructed out of obituaries from the two separate web sites. The first test corpus (Text A) has 4 498 obituaries posted to osmrtnice.hr web site during June and December of 2013 and January, February, March and April of 2014 and the 2nd (Text B) has 312 obituaries posted to the web site

http://www.osmrtnica.net/ during May, June, July, August, November and December of 2013 as well as February, March and April of 2014. The obituaries that only had the name of the deceased, without any family members, were removed from the test corpora.

Before we started to work on the grammars, it was important to recognize the structure of an obituary which can be split into 5 sections. The first section is (1) a notice of death, followed by (2) the list of relationships of the deceased to the other members of the family describing how that person related to those who are left behind. This is a list of comma separated, gender dependant relationships like mother, sister, grandmother, great grandmother, mother-in-law, aunt. What follows is (3) the name of the person (Figure 3) and (4) the information about the funeral. At the end (5) the list of grieving family members, with additional information of a relationship of that person to the deceased, is provided. The list is populated with two types of constructions. One is the nominative construction that holds a relationship followed by the first and none or all of the following: last name, nick name, "with a spouse", "and children" like in the following examples[1]:

- *kći Ruža; (en. daughter Ruža)*
- *kći Ruža Matić, (en. daughter Ruža Matić)*
- *kći Ruža sa suprugom Markom, (en. daughter Ruža with husband Marko)*
- *kći Ruža sa suprugom Markom i djecom, (en. daughter Ruža with husband Marko and children)*
- *kći Ruža sa suprugom Markom i djecom Anom i Perom. (en. daughter Ruža with husband Marko and children Ana and Pero).*

The second construction is in genitive and it holds expressions like 'family of (deceased)' followed by the first name and none or all of the following: last name, nick name, 'with a spouse', 'and children' as in the following examples:

- *obitelj sestre Ruže (en. family of sister Ruža)*
- *obitelj pokojne sestre Ruže (en. family of the deceased sister Ruža)*
- *obitelj sestre Ruže sa suprugom Markom i djecom Anom i Perom (en. family of sister Ruža with husband Marko and children Ana and Pero).*

[1] Out of respect for the deceased and his/her families, all the names in examples used in this paper will be scrambled and used only to show the patterns that are being described.

CHAPTER X

Building grammars

After applying the preliminary lexical analysis in NooJ, including only Croatian dictionaries, to both corpora we detected which kinship terms were not present in the main dictionary. This problem was solved with a two-step process. The first step was to add to the main dictionary the missing kinship terms found in NooJ's UNKNOWN list including only terms that describe blood relationships between grandparents and grandchildren. All the kinship terms in the dictionary are further described with the new semantic feature *+obt* as an abbreviation of Croatian word for family (hr *obitelj*).

With the second step we needed to cover all the remaining blood relationships (those hierarchically above the grandparents and below grandchildren) and all the half-relationships (obtained either by remarriage or adoption) where possible. We opted to build a lexical grammar that recognizes a large number of relationships that are rarely found in texts.

The grammar checks if there exists a noun in the dictionary that has *+obt* notation and recognizes any such word that has one prefix "*polu*" or *"*po" as a half-relationship or one or more occurrences of the prefix "*pra"* or "*šukun"* for the distinct kinship relations. Thus, words like ***pra****baba* (en. great grandmather), ***prapra****baka* (en. great great grandma), ***prapraprate****ta* (en. great great great aunt) or ***pra****unuk* (en. great grandson), but also words like ***polu****brat* (en. halfbrother) or ***po****majka* (en. stepmother) are recognized. However, the word *pastorak* (en. stepson) would not be recognized with our lexical grammar and was, among few others that are not derived in the above described manner, added directly to the dictionary. The dictionary now consists of 143 kinship terms with the following distribution: 86 terms of blood relationships, 46 terms of marriage relationships, 5 terms used for blood or marriage relationships and 6 terms used for relationships obtained through adoption or remarriage.

After we defined all of the kinship terms we constructed syntactic grammars for recognizing relationships among the deceased and his/her family members. Four main syntactic grammars were built for this project: a) for disambiguating personal names; b) for annotating the deceased person (section 3 in the obituary); c) for annotating the irrelevant text (sections 1 and 4 in the obituary); d) for annotating the individual relationship (section 5 in the obituary). Grammars for sections a, b and d will be shown, described, explained and evaluated in the following sections.

Disambiguation of personal names

After the preliminary lexical analysis, we were surprised to discover a very high number of unknown first and last names even though the Croatian dictionary of personal nouns has 36 359 last names and 5 414 first names (2 809 female and 2 605 male) (Vučković, 2009; Bekavac, 2005). Since the recognition of personal names was important for the project we needed to find a solution that would work not only for the corpus we had but for any future additions to that corpus as well. We were faced with two types of problems concerning: unknown words (words missing from the dictionary) and ambiguous words. Following further analysis we were able to recognize 5 categories of missing names:

a) foreign names – not characteristic for Croatia: Max, Carmen, Ellie;
b) different spelling variants of the name that exists in our dictionary: Mihael (in the dictionary) – Mikael or Mihail (not in the dictionary);
c) names used for both genders but found only as M or F in our dictionary: Nedjeljko (M) – Nedjeljka (F) (same genitive form that causes problem in construction "*sa suprugom Nedjeljkom*" – en. with spouse Nedjeljkom);
d) name used both as a first and last name: Vukman – last name (in the dictionary) – first name (not in the dictionary);
e) misspelled names.

The second type of problem is related to ambiguity. We recognize two types of ambiguity: inner and outer homographs. It is very common for Croatian nouns to have the same forms in different cases (e.g. singular Nominative, Accusative and Vocative, or plural Dative, Locative and Instrumental). This type of ambiguity is marked as inner ambiguity and is disambiguated with the grammar in Figure 1. Consequently all the nouns marked as proper nouns in Genitive or Nominative case that are found immediately after the common noun in the same case with the semantic tag **+obt** (+family) are disambiguated as proper nouns <N+vl>.

The list of grieving family members is mainly populated with the Nominative phrases like: brat Nikola (en. *brother Nikola)*, sestra Nada *(en. sister Nada)*, but in the construction where the whole family is listed in the form *family of (deceased) brother Nikola*, the expression following the Nominative noun *family* is in the Genitive. If the personal name in this expression is male with both first and last names given, both names are in the Genitive case (*obitelj brata Nikole Jurića -> family of Nikola Jurić's brother*). However, if that person is female, than her first name is in the Genitive case, but the last name remains in the Nominative case (*obitelj sestre Nade Jurić -> family of Nada Jurić's sister*).

CHAPTER X

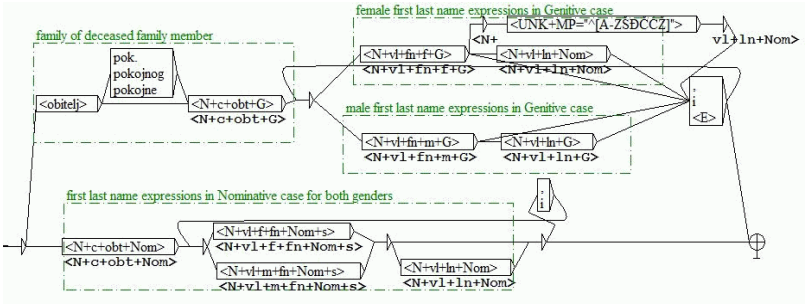

Figure 1 - Disambiguation of personal names

The next subset of proper names that are ambiguous in Croatian language appear as common nouns (Nada -> nada <N> *engl. hope*), as well as adjectives (Mila -> mila <A> *engl. dear*) or verbs (Mare -> mare <V> *engl. to care*). New grammar (Figure 2) was built to solve the problem of unknown names and names marked as another word category. The logic behind this grammar is to annotate the unknown word depending on the kinship term that precedes it while taking into account the gender, number and case of that term. If a kinship term is given in the singular form, than the name that follows it inherits the gender in the following fashion: a female relationship in singular is followed by a female first name, and male relationship in singular is followed by a male first name. The same is true for the plural female relationship that is followed by a list of female comma separated names in the singular form with the last two names connected with "and".

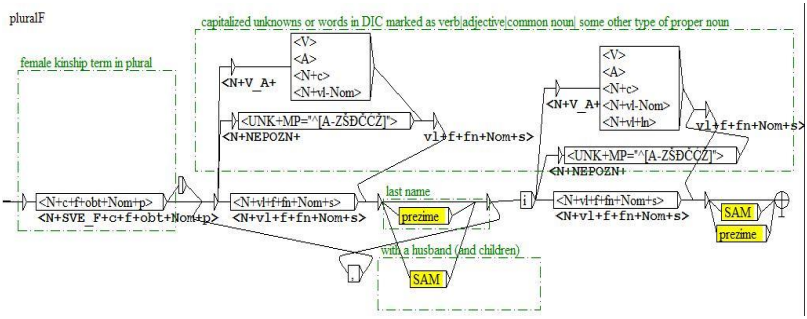

Figure 2 - Annotating unknown and wrong type names after the plural female kinship term

However, when some plural male or neuter term is used followed by a list of names, the problem arises. The list can hold both male and female

names, known and unknown names or even words that belong to other word categories. Luckily, this list of kinship terms is short and it includes the following: *djeca* (en. children), *nećaci* (en. nephews), *pastorci* (en. stepchildren), *praunučad* or *praunuci* (en. great grandchildren), *rođaci* (en. cousins), *unučad* or *unuci* (en. grandchildren). We used some logical operators that helped us solve the following two possibilities:

a) IF only two names follow the plural male term AND known name is female THEN the one unknown name must be male
b) IF more than two names follow the plural male term AND only one is unknown AND all known are of female gender THEN the unknown must be male.

For all other combinations, a separate, more detailed study is needed.

Annotating the deceased person

The deceased person is written with capitalized (one or two) first and (one or two) last names in the Nominative case. Very often, the male names have their nicknames written immediately following their names with or without brackets. In some cases, the name is preceeded with a title attached to the name by virtue of office, rank or as a mark of respect (see node <N+titula> in Figure 3).

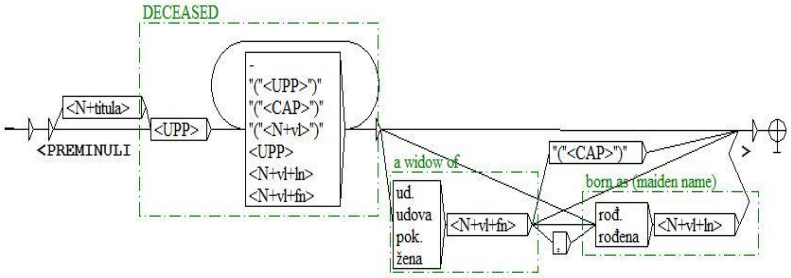

Figure 3 - Recognizing the deceased person

After the male names, constructions that mark whose son he was, may be found. After the female names two additional constructions are found, either separate or in combination, marking whose wife or a widdow she was, and/or her maiden name (Figure 3) as in the following examples:

- *gđa. RUŽA MATIĆ rođ. Ružić (en. Mrs. Ruža Matić born Ružić)*
- *gđa. RUŽA MATIĆ žena Markova rođ. Ružić (en. Mrs. Ruža Matić wife of Marko born Ružić)*

The grammar in Figure 3 is a subgraph of a larger graph where its exact context is defined. The grammar has the following performance: Text A (P: 0.95; R: 0.82; f: 0.88) and Text B (P: 1; R: 0.99; f: 0.99). Some city names that are MWUs, like *Kaštel Lukšić* or *New Jersey*, are falsely recognized with this grammar, as well as other capitalized words found after the name of the person (due to the missing full stop after the name). These occurrences are the main cause of lower precision in Text A.

Recognizing the individual relationships

Croatian language offers kinship terms for all gender dependant relationships like, e.g.:

- *pašanac –wife's sister's husband*
- *šurjak –wife's brother*
- *djever –husband's brother*
- *svak –sister's husband*
- *svastika –wife's sister*
- *šurjakinja –wife's brother's wife*
- *zaova –husband's sister*
- *jetrva –husband's brother's wife*

However, not all parts of Croatia use the same terminology. Sometimes, *šogor* or *kunjad* replace four different male relationships (*pašanac, šurjak, djever* and *svak*) and *šogorica* or *kunjada* four different female relationships (*svastika, šurjakinja, zaova, jetrva*). To get a more unified list of relationships, only one term was used where multiple possibilities exist, e.g. we used term <MATI> for occurrences *mama, majka, mati* (en. mom, mother).

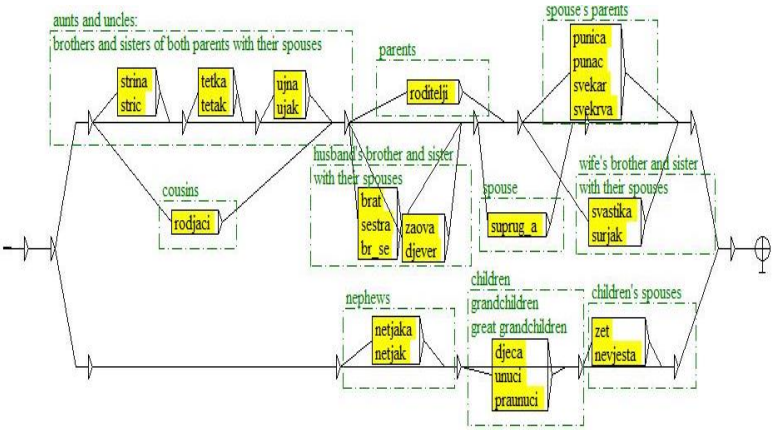

Figure 4 - Recognizing each family member of the deceased

The more detailed subgraph, describing the relationship *sestra* (en. sister) is given in Figure 5. The main graph *sestra* is in the middle while its subgraph <obtPok> describing the constructions like 'the family of the deceased', is positioned above and the subgraph describing female name <nameF> bellow it. The graphs of other relationships are similar and are gender dependent, i.e. <nameF> is replaced with the subgraph describing male names <nameM> for the male directed relationships.

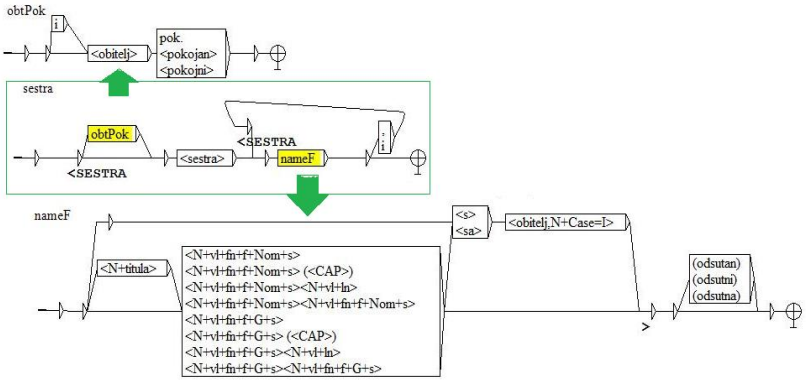

Figure 5 - Detailed description of the node *"sestra"* (en. sister)

Unlike Japanese, Chinese or Korean (Baik et al, 2010), there is no distinction between the older or younger siblings in the Croatian kinship terminology. Figure 4 presents a main grammar with subgraphs describing each relationship found in the corpus. This grammar has somewhat better performance on Text B (P: 0,98; R: 0,93; f: 0,95) than on Text A (P: 0,95; R: 0,85; f: 0,90).

Building the XML File

The last grammar that is applied to an obituary is the grammar that produces an XML notation for each family. The main node, marked <OBITELJ> (en. family), has a set of atribute-value pairs. The first pair is always the deceased and his/her name. In other pairs, an attribute is the name of the relationship and the value is a name. Thus, an example for the deceased person MARA KEVO and a list of the grieving family members *daughter Ruža, son in law Marko and grandchildren Ana and Pero* is rewritten as:

- <OBITELJ PREM='MARA KEVO' KĆER='Ruža' ZET='Marko' UNUKA='Ana' UNUK='Pero'>

Drawing the relationships with Python

We built an algorithm in Python to process the XML files with predefined family relations. The main purpose of this algorithm is to generate a graphic representation of the family relations and the family members (family trees). We used one of Python's modules, Pydot, not to draw graphs or visual representations of data, but to send commands and instructions to software application called „Graphviz".

Following our first theoretical idea for the algorithm, we imagined a simple GUI which takes 1 argument as the input ("Name of XML file") and returns the graphic images of the family trees (family tree relations). After some time, we ran into a problem with big data processing and organizing of outputs, so we determined 2 inputs: "Directory of multiple XML files" and "Directory of output files". This solution helps the algorithm process multiple XML files in one loop and output the images to defined directories and subdirectories named after the XML filename. The name of each output file (.gif, .pdf, .png, .jpg) is defined by the name of the corresponding family it represents thus making the outputs easier to search and organize.

There are three main steps in our Python algorithm. In the first part of the algorithm each line of the document is parsed and attribute-value pairs are extracted into lists and diagrams in form (TAG, Name). In the second step the FOR loop iterates through all the diagrams in the family list and checks for values of TAG. From those values the algorithm coordinates and organizes specific predefined clusters according to the Name variable. After organizing all names to matching clusters, the algorithm checks the gender of the person and sends a command to Graphviz to create a cluster and write names within that cluster. The main part of the third step is setting relationships between clusters and adding empty clusters to make the graph look like a family tree. Finally, the graph is closed and image is placed into a predefined directory.

Because of some limitations within Graphviz, we decided on a unique output solution as shown in Figure 6. The deceased person (ego) is highlighted and oriented in the center of the graph and all other relations are set according to him/her. Individuals are grouped into clusters (Children, Parents, Cousins, etc.). The edge of every cluster is color-coded according to the relation (red → blood relatives, green→ spouse, brown → spouse's family) and the gender (blue → male, purple → female).

However, only a spouse, parents and parents in law of an ego are known. Marital connections of other relatives were not marked since they were ambiguously stated in the text.

Unsolved Problems

There are still two main types of unsolved problems that will be hard to find the (unambiguous) solutions to. The first type makes building family tree from all obituaries impossible and the second one makes it a challenge. The first list would include those occurrences where a list of grieving family members is build out of:

a) only names, without the type of relation (HR: njegova Maria, Ana, Ivo i Nikola - EN: his Maria, Ana, Ivo and Nikola);
b) only relations, without the personal names (HR: sinovi i kćeri s obiteljima - EN: sons and daughters with their families);
c) only the last names (HR: obitelj Frankopan - EN: family Frankopan)-

The second type includes:

a) list of (mixed) names given after the plural relation that appear both as male or female names (HR: unuci Matija i Saša - EN: grandchildren Matija and Saša);
b) two or more names after the relation that can be either last name or 2nd part of first name (HR: kći Ana Franka - EN: daughter Ana Franka);
c) ungrammatical constructions, e.g. - there is no comma before *i* (en. and) (HR: unučad Luka, Nina, Ivan, i Andrija. - EN: grandchildren Luka, Nina, Ivan, and Andrija).

Conclusion

In this paper, we described the process of information extraction with NooJ in order to fill an ontological framework dealing with Croatian family relationships.

Due to the problems described in the previous sections, it is not possible to build a family tree from all obituaries. However, those that have a list of grieving family members in the form of relation-personal name(s), easily fit in the prepared framework.

In future work, in order to improve our recall and precision, we intend to further enhance our grammars describing additional contexts and add more frequent personal names to NooJ dictionary.

CHAPTER X

References

Songiy Baik, Hee-Rahk Chae. 2010. An Ontological Analysis of Japanese and Chinese Kinship Terms. Proceedings of the 24th Pacific Asia Conference on Language, Information and Computation. Institute for Digital Enhancement of Cognitive Development, Waseda University: 349-356.

Božo Bekavac. 2005. Strojno prepoznavanje naziva u suvremenim hrvatskim tekstovima (en. Machine Named Entity Recognition in Contemporary Croatian Texts), PhD Thesis, Department of Linguistics, Faculty of Humanities and Social Sciences, University of Zagreb.

Chris Biemann. 2005. Ontology Learning from Text: A Survey of Methods. LDV Forum 20(2): 75-93.

Željko Jozić, Perina Vukša, Dijana Ćurković. 2012. Nazivi za bratova sina u hrvatskome jeziku (en. Brother's son in the Croatian language). Rasprave: Časopis Instituta za hrvatski jezik i jezikoslovlje 37/2: 393-422.

Iva Pleše. 1998. Neki aspekti hrvatske terminologije srodstva (en. Some Aspects of the Croatian Kinship Terminology). Etnološka tribina: godišnjak Hrvatskog etnološkog društva (0351-1944) 28: 59-78.

Eduardo Salza. 2014. Using NooJ as a System for (Shallow) Ontology Population from Italian Texts in Formalising Natural Languages with NooJ 2013: Selected Papers from the NooJ 2013 International Conference. Eds. S. Koeva, S. Mesfar, M. Silberztein. Cambridge Scholars Publishing: Newcastle, Germany.

Max Silberztein. 2003. NooJ manual. http://www.nooj4nlp.net (223 pages)

Mirjana Šokota. 1998. Rodbinsko-svojbinski i slični nazivi u Ždrelcu (en: Family – In-Laws' and Similar Names in Ždrelac). Čakavska rič XXVI, br. 1-2, Split: 33-44.

Franjo Tanocki. 1986. Rječnik rodbinskih naziva (en: Dictionary of kinship terms) Revija- izdavački centar Radničkog sveučilišta 'Božidar Maslarić', Osijek.

Ljubica Tikvica. 2009. O hrvatskoj terminologiji srodstva – Neki aspekti obradbe u suvremenim rječnicima hrvatskoga jezika (en. About Croatian Kinship Terminology – Some aspects in contemporary Croatian dictionaries). Hum: časopis Filozofskog fakulteta Sveučilišta u Mostaru (1840-233X) 1, 5: 106-124.

Kristina Vučković. 2009. Model parsera za hrvatski jezik (en. Model of a Parser for Croatian Language), PhD Thesis, Department of Information and Communication Sciences, Faculty of Humanities and Social Sciences, University of Zagreb.

Building Family Trees With NooJ

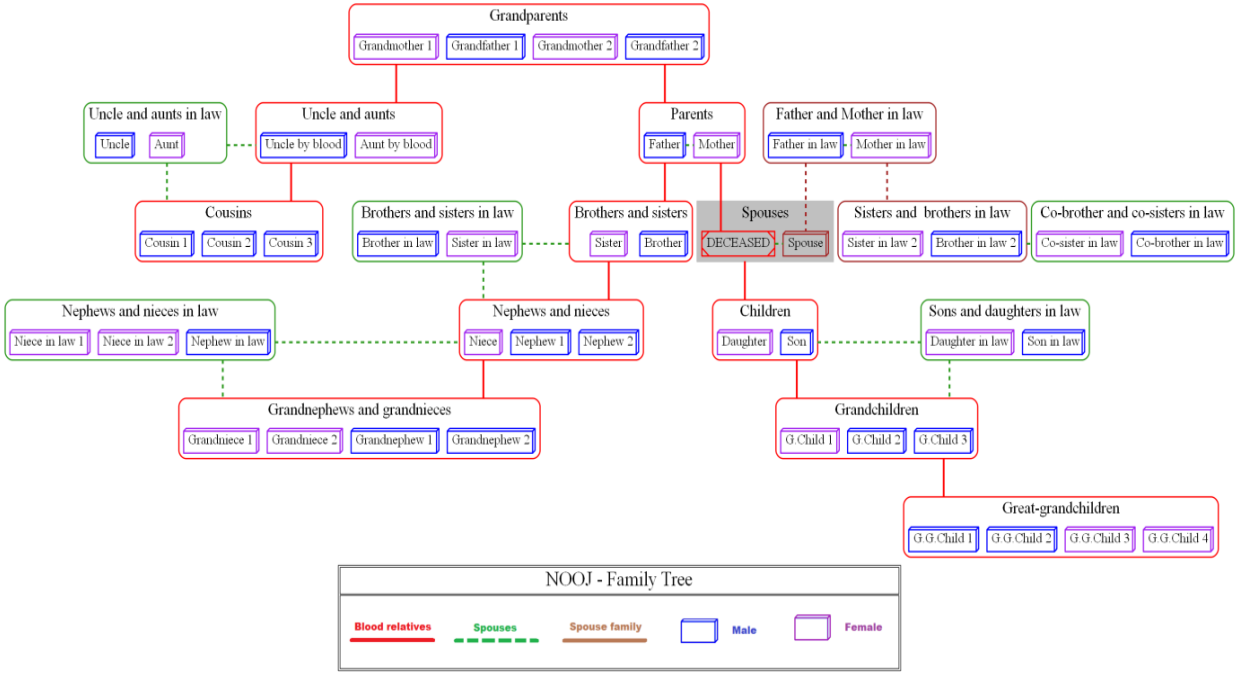

Figure 6 - Representation of a family tree using Python