Compiling Tools and Resources for Studying of Luxemburgish Language and beyond.

Joshgun Sirajzade
University of Luxembourg
joshgun.sirajzade@uni.lu

After collaborating and working in different projects, which directly applied computational methods, at the Institute of Luxembourgish Language and Literature under Prof. Dr. Peter Gilles, University of Luxembourg[1] I would like not only to summarize the tools and resources we built in the last years but also to give a theoretical analysis of how computational methods can actually affect or even change our research and vice versa, enabling us to see new correlations.[2] For a long time the responsibility of the available technologies for the Digital Humanities remained in the information technology sector, and they were merely considered as routines or tools (Ramsay and Rockwell 2012). As the technology is crucial for our research today, I would like to highlight the necessity to question and develop the technologies we use in the Digital Humanities ourselves. However, looking back I see that some decades ago every project used – just to name one example – some different sort of data encoding and was doing a tedious work in order to query and exchange the data, whereas today we have interoperable strategies like XML and lots of ready to use tools available, which make our work faster and easier. XML has become pretty famous in the Digitals Humanities for some time now, and the same applies to databases, programming languages, web technologies etc., too. Digital Humanities needs its own tools and I think there should also be a theoretical part about it inside of Digital Humanities, while it is also a good thing to leave the low-level part of it like memory control etc. to computer science. What Digital Humanities is interested in is high-level programming for its own tasks. Thus, the question has two tails; a) what tools are needed in Digital Humanities today and b) how can the hermeneutics of Digital Humanities be used to improve these tools? I think if we combine these two strategies, we might one day receive better answers to our questions, which conventional humanities were not able to give. But before providing an insight into theoretical issues, let me introduce some empirical work, on which my theoretical thoughts are based.

The Idea behind the platform Phraseolux[3] was about gathering Luxembourgish phraseologies and storing them in a relational database. The gathered phraseologies were annotated by their lexical constitutes and semantical fields of their meaning. By storing them in a database it was not only possible to get precise frequencies about the material very quickly (and that in many possible ways, e.g. number of phraseologies as a token and type or number of word tokens and types etc.) and do some advanced statistics but also to achieve a better semantical understanding of the phraseologies

[1] See (InfoLux 2016).

[2] See (Fitzpatrick 2012) for further discussion of interaction of „scholars use computing technologies to investigate the kinds of questions that are traditional to humanities“ vs. "ask traditional kind of humanities-oriented questions about computing technologies".

[3] The platform is a result of the project DOLPH - Dynamics of Luxembourgish Phraseology (2011-2014), project leader: Prof. Dr. Peter Gilles, Dr. Anne Kleine, funding: The Luxembourg National Research Fund (PhraseoLux 2016).

through their connection by the semantical field. By doing so, it was possible to see which semantical fields have more phraseological units, thus about what people make the most metaphorical thinking. Another interesting point was visualizing these data and seeing how recipients interpret them (See figure 1). Having the data organized digitally in a database opened so many different opportunities that not all possibilities were able to be drawn upon and the institute is still making use of the data again and again, even after the project was officially finished in 2014. Technically the project made use of scripting web languages like PHP, database software MySQL (later one of its successors MariaDB) and JavaScript libraries like d3.js for visualizing the data.

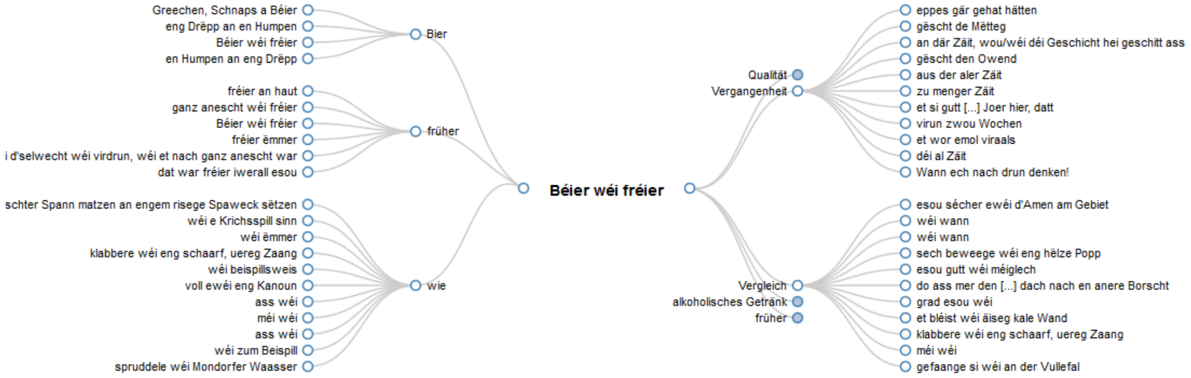

Greechen, Schnaps a Béier
eng Drëpp an en Humpen
Béier wéi fréier
en Humpen an eng Drëpp
Bier
fréier an haut
ganz anescht wéi fréier
Béier wéi fréier
fréier ëmmer
i d'selwecht wéi virdrun, wéi et nach ganz anescht war
dat war fréier iwerall esou
früher
schter Spann matzen an engem risege Spaweck sëtzen
wéi e Krichsspill sinn
wéi ëmmer
klabbere wéi eng schaarf, uereg Zaang
wéi beispillsweis
voll ewéi eng Kanoun
ass wéi
méi wéi
ass wéi
wéi zum Beispill
spruddele wéi Mondorfer Waasser
wie
Béier wéi fréier
Qualität
Vergangenheit
eppes gär gehat hätten
gëscht de Mëtteg
an där Zäit, wou/wéi déi Geschicht hei geschitt ass
gëscht den Owend
aus der aler Zäit
zu menger Zäit
et si gutt [...] Joer hier, datt
virun zwou Wochen
et wor emol viraals
déi al Zäit
Wann ech nach drun denken!
Vergleich
alkoholisches Getränk
früher
esou sécher ewéi d'Amen am Gebiet
wéi wann
wéi wann
sech beweege wéi eng hëlze Popp
esou gutt wéi méiglech
do ass mer den [...] dach nach en anere Borscht
grad esou wéi
et bléist wéi äiseg kale Wand
klabbere wéi eng schaarf, uereg Zaang
méi wéi
gefaange si wéi an der Vullefal

Figure 1 Lexical constituents and semantical fields of 'Béier wéi fréier'(beer as before). By clicking on them (Vergleich (comparing) or Vergangeheit (the past)) one can see other phraseologies, which have same annotations.

Currently I am working on the project WBLUX2 — Luxembourgish word formation. Historical preconditions and contrastive analysis.[4] The project aims to study word formation in the Luxembourgish language. This time we decided to use a different database, namely the so called XML-Database eXist, because our empirical text sources were already available in XML. As a programming language we took Java this time. Script languages are often simple to use and pretty fast to write. But the produced code is often not easily readable, and errors are difficult to find. In my opinion if you work in a bigger project, you need to use a language, in which code can be maintained easily.[5] Java is one of the most common programming languages, has lots of tools and so called libraries with high level abstraction – which I am going to discuss later – that makes it a good choice for long living software. For the study of word formation, we used some tools and strategies, which come both from digital humanities and corpus linguistics. First of all our texts were coded in TEI, so everyone, who is familiar with TEI can take look at our data and knows immediately what it is about. Secondly, we used strategies like Tokenization, Standardization and POS-Tagging, which derive from corpus linguistics (see figure 2 and 3). And again it is about linking the data, organizing it smartly, in order to get best results. For example, if you are searching for an ending *-er* like in *writer*, you also get *better*, because a language sign is multifunctional. But if you use information from POS-Tagging and search only in nouns for this ending, you will get better results. By doing so, it is possible to build annotations on one another, which can be used both in GUI for users as a search tools and for automatic tasks as well.

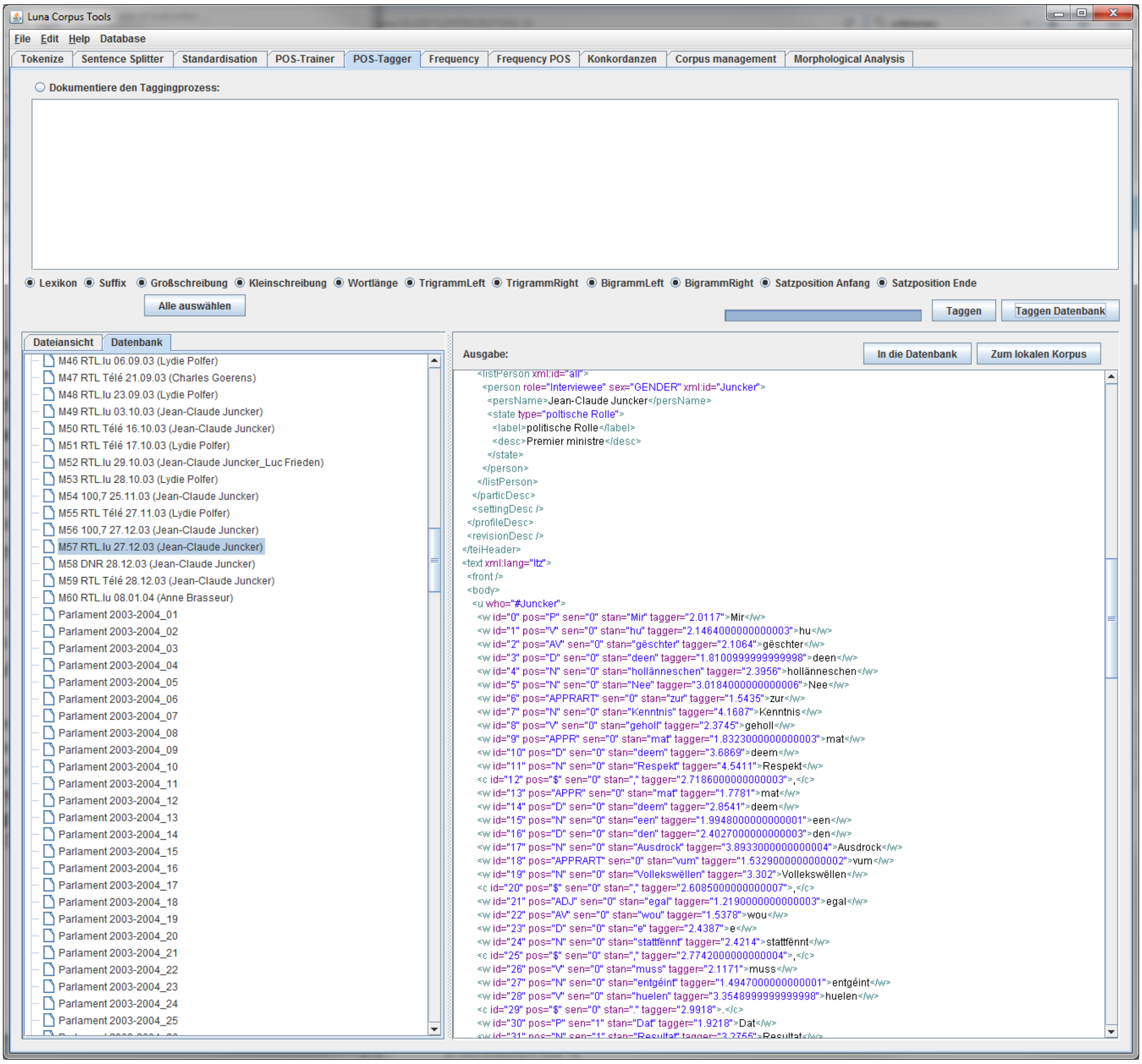
[4] Project Leader: Prof. Dr. Peter Gilles, Prof. Dr. Heinz Sieburg, Funding: University of Luxembourg.

[5] Sustainability is a big issue in building tools, which can also be 'thorny' for any project (Terras 2012).

In order to make theoretical assumptions the situation in computer science has to be reviewed first. In the history of software development it has always been the similar case to using existing information in order to produce new ones. There it is called abstraction[6] and has two perspectives: control abstraction and data abstraction. Although the term 'abstraction' can also mean something different in the humanities, here it refers to the combination of commands e.g. subroutines, modules or software components, in order to generate new commands for prompt and frequent usage. As example one can imagine, if I want to open and see a picture, a computer needs to perform many steps starting with reading the data from the hard disc to the memory, and then sending it to the monitor to turn on certain pixels in certain colors etc. Basically, if you need to do it frequently, you abstract these steps to one command, instead of describing these small steps every time again. In fact, you can break down every action on a computer until it goes to ones and zeros. However, as a digital humanist you are often not even really interested in these tiny steps, but rather in using existing abstract tools (like XSLT or certain Java and Python libraries) for certain tasks to do the job.

There are some well-known advantages of humanities going digital, like accessibility over internet, or the fact that everything can easily be copied and multiplied. However, one particular advantage has not yet been discussed extensively. This is the ability of abstraction, thus using software parts and information in order to build new ones. And this is not only control and data abstraction, but also concerns other fields like linking or visualizing the data. A good example here would be hyperlinks, which are almost predated from today's perspective. But in the history of the internet, these were developed as a (new) concept and technical realization. So where are we going now – popups, widget switches, drag and drops to name some other newer methods of acting on internet. Which technologies are best for certain parts of digital humanities? Knowing the history of how digital media evolved so far, we can take a look to the future, and not only predict things which are going to come, but actively take part in their development.

[6] See about „Abstract. Simplify. Generalize" In: (Mcpherson 2012).



○ Dokumentiere den Taggingprozess:

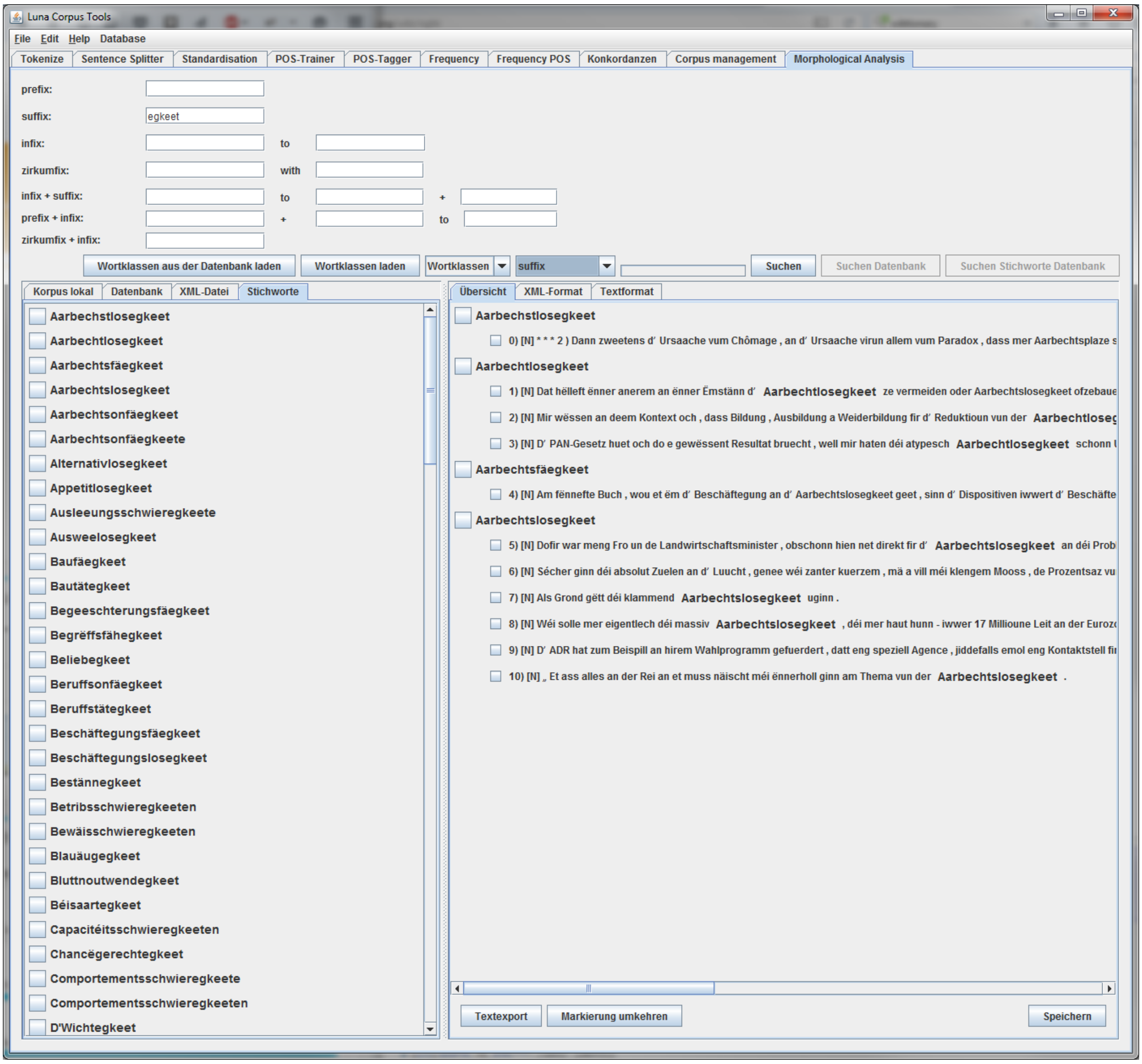
● Lexikon ● Suffix ● Großschreibung ● Kleinschreibung ● Wortlänge ● TrigrammLeft ● TrigrammRight ● BigrammLeft ● BigrammRight ● Satzposition Anfang ● Satzposition Ende

M46 RTL.lu 06.09.03 (Lydie Polfer)
M47 RTL Télé 21.09.03 (Charles Goerens)
M48 RTL.lu 23.09.03 (Lydie Polfer)
M49 RTL.lu 03.10.03 (Jean-Claude Juncker)
M50 RTL Télé 16.10.03 (Jean-Claude Juncker)
M51 RTL Télé 17.10.03 (Lydie Polfer)
M52 RTL.lu 29.10.03 (Jean-Claude Juncker_Luc Frieden)
M53 RTL.lu 28.10.03 (Lydie Polfer)
M54 100,7 25.11.03 (Jean-Claude Juncker)
M55 RTL Télé 27.11.03 (Lydie Polfer)
M56 100,7 27.12.03 (Jean-Claude Juncker)
M57 RTL.lu 27.12.03 (Jean-Claude Juncker)
M58 DNR 28.12.03 (Jean-Claude Juncker)
M59 RTL Télé 28.12.03 (Jean-Claude Juncker)
M60 RTL.lu 08.01.04 (Anne Brasseur)
Parlament 2003-2004_01
Parlament 2003-2004_02
Parlament 2003-2004_03
Parlament 2003-2004_04
Parlament 2003-2004_05
Parlament 2003-2004_06
Parlament 2003-2004_07
Parlament 2003-2004_08
Parlament 2003-2004_09
Parlament 2003-2004_10
Parlament 2003-2004_11
Parlament 2003-2004_12
Parlament 2003-2004_13
Parlament 2003-2004_14
Parlament 2003-2004_15
Parlament 2003-2004_16
Parlament 2003-2004_17
Parlament 2003-2004_18
Parlament 2003-2004_19
Parlament 2003-2004_20
Parlament 2003-2004_21
Parlament 2003-2004_22
Parlament 2003-2004_23
Parlament 2003-2004_24
Parlament 2003-2004_25

Ausgabe:

```
      <listPerson xml:id="all">
       <person role="Interviewee" sex="GENDER" xml:id="Juncker">
        <persName>Jean-Claude Juncker</persName>
        <state type="poltische Rolle">
         <label>politische Rolle</label>
         <desc>Premier ministre</desc>
        </state>
       </person>
      </listPerson>
     </particDesc>
     <settingDesc />
    </profileDesc>
    <revisionDesc />
   </teiHeader>
   <text xml:lang="ltz">
    <front />
    <body>
     <u who="#Juncker">
      <w id="0" pos="P" sen="0" stan="Mir" tagger="2.0117">Mir</w>
      <w id="1" pos="V" sen="0" stan="hu" tagger="2.1464000000000003">hu</w>
      <w id="2" pos="AV" sen="0" stan="gëschter" tagger="2.1064">gëschter</w>
      <w id="3" pos="D" sen="0" stan="deen" tagger="1.8100999999999998">deen</w>
      <w id="4" pos="N" sen="0" stan="hollänneschen" tagger="2.3956">hollänneschen</w>
      <w id="5" pos="N" sen="0" stan="Nee" tagger="3.0184000000000006">Nee</w>
      <w id="6" pos="APPRART" sen="0" stan="zur" tagger="1.5435">zur</w>
      <w id="7" pos="N" sen="0" stan="Kenntnis" tagger="4.1687">Kenntnis</w>
      <w id="8" pos="V" sen="0" stan="geholl" tagger="2.3745">geholl</w>
      <w id="9" pos="APPR" sen="0" stan="mat" tagger="1.8323000000000003">mat</w>
      <w id="10" pos="D" sen="0" stan="deem" tagger="3.6869">deem</w>
      <w id="11" pos="N" sen="0" stan="Respekt" tagger="4.5411">Respekt</w>
      <c id="12" pos="$" sen="0" stan="," tagger="2.7186000000000003">,</c>
      <w id="13" pos="APPR" sen="0" stan="mat" tagger="1.7781">mat</w>
      <w id="14" pos="D" sen="0" stan="deem" tagger="2.8541">deem</w>
      <w id="15" pos="N" sen="0" stan="een" tagger="1.9948000000000001">een</w>
      <w id="16" pos="D" sen="0" stan="den" tagger="2.4027000000000003">den</w>
      <w id="17" pos="N" sen="0" stan="Ausdrock" tagger="3.8933000000000004">Ausdrock</w>
      <w id="18" pos="APPRART" sen="0" stan="vum" tagger="1.5329000000000002">vum</w>
      <w id="19" pos="N" sen="0" stan="Vollekswëllen" tagger="3.302">Vollekswëllen</w>
      <c id="20" pos="$" sen="0" stan="," tagger="2.6085000000000007">,</c>
      <w id="21" pos="ADJ" sen="0" stan="egal" tagger="1.2190000000000003">egal</w>
      <w id="22" pos="AV" sen="0" stan="wou" tagger="1.5378">wou</w>
      <w id="23" pos="D" sen="0" stan="e" tagger="2.4387">e</w>
      <w id="24" pos="N" sen="0" stan="stattfënnt" tagger="2.4214">stattfënnt</w>
      <c id="25" pos="$" sen="0" stan="," tagger="2.7742000000000004">,</c>
      <w id="26" pos="V" sen="0" stan="muss" tagger="2.1171">muss</w>
      <w id="27" pos="N" sen="0" stan="entgéint" tagger="1.4947000000000001">entgéint</w>
      <w id="28" pos="V" sen="0" stan="huelen" tagger="3.3548999999999998">huelen</w>
      <c id="29" pos="$" sen="0" stan="." tagger="2.9918">.</c>
      <w id="30" pos="P" sen="1" stan="Dat" tagger="1.9218">Dat</w>
      <w id="31" pos="N" sen="1" stan="Resultat" tagger="3.2755">Resultat</w>
```

Figure 2 The Java Swing GUI to the bundle of tools, like Tokeniser, POS-Tagger etc. The Tool has eXist-Database at the backend and supports XML



suffix: egkeet

suffix

☐ Aarbechstlosegkeet
☐ Aarbechtlosegkeet
☐ Aarbechtsfäegkeet
☐ Aarbechtslosegkeet
☐ Aarbechtsonfäegkeet
☐ Aarbechtsonfäegkeete
☐ Alternativlosegkeet
☐ Appetitlosegkeet
☐ Ausleeungsschwieregkeete
☐ Ausweelosegkeet
☐ Baufäegkeet
☐ Bautätegkeet
☐ Begeeschterungsfäegkeet
☐ Begrëffsfähegkeet
☐ Beliebegkeet
☐ Beruffsonfäegkeet
☐ Beruffstätegkeet
☐ Beschäftegungsfäegkeet
☐ Beschäftegungslosegkeet
☐ Bestännegkeet
☐ Betribsschwieregkeeten
☐ Bewäisschwieregkeeten
☐ Blauäugegkeet
☐ Bluttnoutwendegkeet
☐ Béisaartegkeet
☐ Capacitéitsschwieregkeeten
☐ Chancëgerechtegkeet
☐ Comportementsschwieregkeete
☐ Comportementsschwieregkeeten
☐ D'Wichtegkeet

☐ **Aarbechstlosegkeet**

☐ 0) [N] * * * 2) Dann zweetens d’ Ursaache vum Chômage , an d’ Ursaache virun allem vum Paradox , dass mer Aarbechtsplaze s

☐ **Aarbechtlosegkeet**

☐ 1) [N] Dat hëlleft ënner anerem an ënner Ëmstänn d’ **Aarbechtlosegkeet** ze vermeiden oder Aarbechtslosegkeet ofzebaue

☐ 2) [N] Mir wëssen an deem Kontext och , dass Bildung , Ausbildung a Weiderbildung fir d’ Reduktioun vun der **Aarbechtloseg**

☐ 3) [N] D’ PAN-Gesetz huet och do e gewëssent Resultat bruecht , well mir haten déi atypesch **Aarbechtlosegkeet** schonn l

☐ **Aarbechtsfäegkeet**

☐ 4) [N] Am fënnefte Buch , wou et ëm d’ Beschäftegung an d’ Aarbechtslosegkeet geet , sinn d’ Dispositiven iwwert d’ Beschäfte

☐ **Aarbechtslosegkeet**

☐ 5) [N] Dofir war meng Fro un de Landwirtschaftsminister , obschonn hien net direkt fir d’ **Aarbechtslosegkeet** an déi Probl

☐ 6) [N] Sécher ginn déi absolut Zuelen an d’ Luucht , genee wéi zanter kuerzem , mä a vill méi klengem Mooss , de Prozentsaz vu

☐ 7) [N] Als Grond gëtt déi klammend **Aarbechtslosegkeet** uginn .

☐ 8) [N] Wéi solle mer eigentlech déi massiv **Aarbechtslosegkeet** , déi mer haut hunn - iwwer 17 Millioune Leit an der Euroz

☐ 9) [N] D’ ADR hat zum Beispill an hirem Wahlprogramm gefuerdert , datt eng speziell Agence , jiddefalls emol eng Kontaktstell fi

☐ 10) [N] „ Et ass alles an der Rei an et muss näischt méi ënnerholl ginn am Thema vun der **Aarbechtslosegkeet** .

Figure 3 The Part of the GUI where users can search for affixes with combination of part of speech

References

Fitzpatrick, K. (2012). The Humanities, Done Digitally. In: Matthew K. Gold (Ed.) Debates in the Digital Humanities. University of Minnesota Press. Minneapolis. p 12-13.

InfoLux. (2016) Fuerschungsportal iwwert d'Lëtzebuergescht. Institut fir lëtzebuergesch Sprooch- a Literaturwëssenschaft. Available: infolux.uni.lu. Last accessed 18th April 2016.

Mcpherson, T (2012). Why Are the Digital Humanities So White? or Thinking the Histories of Race and Computation. In: Matthew K. Gold (Ed.) Debates in the Digital Humanities. University of Minnesota Press. Minneapolis. p 146.

PhraseoLux. (2016). Datenbank zur Phraseologie des Luxemburgischen . Available: http://phraseolux.uni.lu. Last accessed 18th April 2016.

Ramsay, S. and Rockwell, G. (2012). Developing Things: Notes toward an Epistemology of Building in the Digital Humanities. In: Gold, Matthew K. (Ed.) Debates in the Digital Humanities. University of Minnesota Press. Minneapolis. p 75.

Terras M. (2012). Present, Not Voting: Digital Humanities in the Panopticon. In David M. Berry (ed.) Understanding Digital Humanities. Palgrave Macmillan, London. p 172.