Universität Tübingen
Seminar für Sprachwissenschaften
General Linguistics
Supervisor: Dr. Johannes Dellert

Exploring the viability of polysemy networks for automated cognate detection under semantic shift

Thesis submitted for the degree of Bachelor of Arts

Arne Rubehn
arne.rubehn@student.uni-tuebingen.de
Matriculation Number: 4017455
Date of Submission: August 16, 2019

Abstract

Over the last decade, a number of promising computational methods have been introduced into historical linguistics. While those methods cleary bear great potential, they mostly are still in a state of development and therefore have flaws and weaknesses. This thesis presents a preliminary approach to enhancing automated cross-linguistic cognate detection by comparing not only translations of the same word to each other, but also taking other semantically close concepts into account. This semantic relatedness could be quantified with the help of polysemy networks. The aim was to find out how generous one can be in terms of semantics when searching for potential cognates between languages, until the probability of similarity by pure chance is too high. This was done by setting up a small sample of six unrelated and geographically separated languages from all across Eurasia, where there should be hardly any cognates at all. However, it became evident that the cognate detection tool that was used was strongly biased towards inputs that contain actual cognates, leading to very unreasonable results that could not satisfyingly answer the initial question. Nevertheless a brief manual analysis of this small sample made the problem and the danger of chance similarities quite clear yet again.

Contents

1 Introduction

Questions about human language and its ways are almost as old as mankind itself and therefore, it is not surprising at all that the first philosophical frameworks with linguistic aspects can be dated to the ancient era. Plato's dialogue *Cratylus* is often considered to be the starting point of language philosophy. This dialogue describes a discussion between Cratylus, who is convinced that all words have their natural righteousness, and Hermogenes, who believes in an arbitrary relation between form and meaning.

Modern linguists would definitely agree with Hermogenes – the arbitrariness between form and meaning is a prominent feature of human language. This arbitrariness allows us to use phonetic similarity between words of different languages with similar meanings as an indicator for relationships. This however leads to highly controversial debates about what lexical resemblances can actually prove a linguistic relationship and which ones are just spurious. I will briefly discuss this debate in Chapter 1.2.

Based on this conflict, in this thesis I will pick up two relatively young approaches on historical linguistics respectively semantic similarity. In Chapter 1.3, I will introduce polysemy networks as a possible solution to quantify semantic similarity, which currently still is a highly vague and subjective value.

It is important to gain a quantifiable value for our next step which is the second aforementioned approach, namely automated cognacy judgement. In this thesis, I am working with the Python library LingPy which I will introduce in Chapter 2.4. Tools like LingPy rely on mathematical approaches that compare phonetic sequences in order to judge how similar they are to each other. Methods like ASJP[1] or weighted sequence alignment[2] have shown promising results in recent studies; this approach is supposed to have great potential for enhancing historical linguistics.

However, in their current state, tools like LingPy completely ignore semantic shifts. They only compare words to each other that have the same meaning in their respective language. My aim in this thesis is to manipulate the input data with the help of polysemy networks as an approach to explore this issue. In Chapter 2, I will explain how this is done in detail. I will then present the results in Chapter 3 and discuss them – and hence also whether the approach is worthwhile in the current state of the tools and databases – in Chapter 4.

1.1 Cognacy

First of all, it is important to define the concept of cognacy, or what we consider to be cognates or not. Usually, a pair or a group of words that are derived from the same word in a common ancestor language are referred to as cognates. However we must keep in mind that there are mainly two definitions of cognacy, a stricter one and a broader one. While classical historical linguists usually refer to such words as cognates that are related to each other via direct descend (vertical transfer), many computational linguists also use the term cognacy for words that have been

[1] Brown et al. (2008)

[2] Jäger (2015)

borrowed from another language (horizontal transfer).[3] To illustrate what that means, consider the English word *queue*, the Spanish *cola* and the French *queue*. All those three words can be traced back to the Latin *cauda* without any doubt; however while it is true that Spanish and French have directly inherited the word from Latin, this isn't the case for English where the word has been borrowed at some point. Therefore, only the Spanish word and the French word are cognates in the stricter sense as they have been vertically transferred, whereas the word has been vertically transferred to the English language.

With this separation in mind, I consider it important to define how the term *cognacy* will be used in a particular framework. In this thesis, I will use the term in the broader sense, i.e. I will consider every pair of words that can be (potentially) traced back to the same ancestral word to be cognates. This definition is more viable when working with automated tools that only rely on sequence comparison without any knowledge of the background of the particular languages – and in this case, we would hope that all the three words in the example mentioned above would be recognized as cognates. For the sake of keeping the terminology simple, words with this kind of relationship shall be called cognates as well.

1.2 Lexical comparison as an index for language relatedness

The most obvious and therefore the most frequently used method to establish relationships between languages or even language families is the successful comparison of mostly basic vocabulary between the languages in question. However, most of the relationships that are sought to be proven are disregarded by the majority of historical linguists as they 'rarely [...] exceed the threshold of chance in both quality and quantity.'[4] Ringe (1992) shows that many similarities that are used to demonstrate alleged relationships can be explained with simple mathematical probability calculations, making them very implausible. While it is true that systematic similarities are the key to proving relationships between languages, many linguists who seek to find deeper relationships than those that are commonly accepted by now seem to disregard the role of chance resemblances. Ringe claims that 'resemblances between languages do not demonstrate a linguistic relationship of any kind unless it can be shown that they are probably not due to chance.' Due to that, the classification of modern languages that most linguists agree on has been pretty stable for the last decades, partitioning the world's languages in approximately 400 families, of which about 200 families consist of only one language. Dellert (2017) claims that the maximum time depth for somewhat safe language reconstruction lies between 6,000 and 8,000 years. If one wants to exceed this threshold and go even further back in time, the density of shared cognates vanishes to an amount that can't be used as a serious proof in order to establish a relationship. As the main reasons for that, Dellert adduces semantic change, lexical replacement and borrowing.

However, there are some scholars that believe in larger-scale relationships between languages and frequently try to reconstruct macro-families. There is even a faction that assumes that all human language has derived from one common proto-

[3] Steiner et al. (2011)

[4] Nichols (2010)

language, most commonly called “Proto-World”. Scholars that believe in macro-families usually rely on alleged reconstructed vocabulary, or to put it in other words, most of these theories only rely on lexical evidence rather than structural similarities in syntax or morphology. There are several reconstructed etymological dictionaries for several macro-families, well-known frameworks for example are the dictionaries for Amerind (Greenberg and Ruhlen; 2007) or Nostratic (Dolgopolsky; 2008). Campbell and Poser (2008) criticize that those theories rely on inaccurate or weak methodologies. The multilateral or mass comparison (i.e. not comparing separate language pairs, but many languages at once) that is commonly used to find evidence for macro-families bears a high risk of chance similarities and is therefore widely regarded as an insufficient method. On top of that, these etymological dictionaries usually are very inaccurate when it comes to semantics, while they also allow for a rather large phonetic variance. As an example for that, Campbell and Poser (2008) adduce and criticize the supposed Amerind root TUNA ‘girl’, where the phonetic representations range over *tun, tana, -tsan, šan, tsini, tu:ne, tele, suri-s, teŋ, tunna, t’ an’ a*, etc.; while the glosses include ‘son, daughter, diminuitive, small, child, be small, mother.’ Campbell and Poser explain that this hypothesis has so weak constraints, that it is very easy to find instances from languages all over the world that fit in the TUNA pattern. To show how broad this etymology is, Campbell and Poser list a bunch of words from non-Amerind langauges that would fit the pattern as well: Finnish *tenava* ‘kid, child’ ; German *Tante* ‘aunt’ ; Japanese *tyoonan* ‘eldest son’ ; Malay *dayang* ‘damsel’ ; Maori *teina* ‘younger sister’ ; ‘younger brother’ ; Tongan *ta’ ahine* ‘girl’ ; Proto-Austronesian **tina* ‘mother’ ; Somali *dállàan* ‘child’ ; Kannada *cina* ‘small’ ; Tamil *tankai/tankacci* ‘younger sister, female parallel cousin’ ; Telugu *cinnadi* ‘girl’ ; Kurux (Dravidian) *tainā* ‘to carry newly married girl out of village’ – they claim that even the English *son* would fit in the pattern.

On the other hand, historical linguistics have to take semantic and phonetic change into account. It would not have been possible to reconstruct the language families that are safely established nowadays without allowing for any change. Ringe (1992) states that the comparison of non-synonyms are generally advantageous for classical methods like the comparative method, despite the higher probability of likeliness due to chance. So what is the right amount of semantic variance we can accept? What is the right trade-off between the both extremes, either only comparing synonyms to each other or comparing anything to each other that shows remote semantic similarity? In the next chapter, I will introduce polysemy networks as an approach to quantifying this semantic similarity and therefore introducing some measurability in this very subjective field.

1.3 Polysemy networks

In recent research, polysemy networks have been proposed as a possible solution to include semantic change in historical linguistics. Croft et al. (2009) stated that semantic change must not be ignored in historical linguistic surveys, yet many computational tools for historical linguistics do not take semantic change into account. That is mainly because semantic change bears very little regularity: Hock (1986) claimed that there are no natural constraints on semantic change. For almost any pair of words one can possibly establish a semantic relationship. According to Traugott and Dasher (2002), the main reason for that is pragmatic: semantic norms are hypothetical norms that underlie the pragmatic aspect of communication. Many semantic changes are therefore also very closely related to the socio-cultural aspect, which again makes it very difficult to state cross-linguistic regularities. McMahon (1994) claims that 'to understand a change in meaning we may also require a good grasp of the socio-cultural situation within a speech community'. For example, early Latin *proclivis* meant 'downhill', but later beared both meanings 'easy' and 'difficult'. While the shift towards the former meaning is quite easy to grasp, Anttila (1972) explained the latter one with a specific technological circumstance: Goods used to be transportated in large ox-carts without efficient brakes, which made it in fact quite difficult to go downhill in such a vehicle.

So instead of observing the mere possibility of a semantic change, it is more worthwhile to shift the focus towards the probability of certain changes, which requires quantitative methods. While it may be true that any change can happen, there are some changes that are more likely to happen than others. Closely related concepts are naturally also more likely to be semantically related to each other, be it that one concept is derived from the other one or that both concepts are represented by the same word. The latter example is what we call a polysemy: Two different concepts that share the same notion in a language. A prominent example for polysemies and different conceptualizations across languages, introduced by Hjelmslev (1961), refers to the conceptual partition between the English notions *tree* and *wood* and between Danish *træ* and *skov*. Both notion pairs span over three concepts that are conceptualized by the German notions *Baum*, *Holz* and *Wald*. But whereas the English *tree* only includes the concept of *Baum*, the Danish *træ* is used for both *Baum* and *Holz*. Parallely, English *wood* is used for *Holz* and *Wald*, in contrast to Danish *skov* which only means *Wald* (see Table 1).

Danish	English	German
træ	*tree*	*Baum*
	wood	*Holz*
skov		*Wald*

Table 1: Different conceptualizations in Danish and English

But why are polysemies so valuable for historical linguistics? Simply speaking, because semantic shift can only happen via colexification. If a word in a language used to have the meaning of A and now means B, there necessarily had to be an 'intermediate' stage where this word meant both A and B.[5] Knowing that, cross-linguistic colexifications become a very valuable resource for reconstructing potential previous semantic shifts and can therefore be very useful for detecting cognates that have undergone semantic changes.

Polysemy networks are an approach to modeling cross-linguistic colexifications that was introduced by Croft et al. (2009) and Perrin (2010). Each concept in a polysemy network is covered by a gloss, a link between two glosses represents that

[5] cf. Perrin (2010), List et al. (2013)

there is at least one language that has a word covering both concepts.[6] Polysemy networks can be extracted automatically from large-scale input data that covers both many languages and many concepts. They can also be enhanced by using weighted edges, i.e. non-binary links that also bear the information of how often (in how many languages or families) the two respective concepts are colexified in the input data. List et al. (2013) use this to extract community structures from a network by cutting off weak and spurious edges.

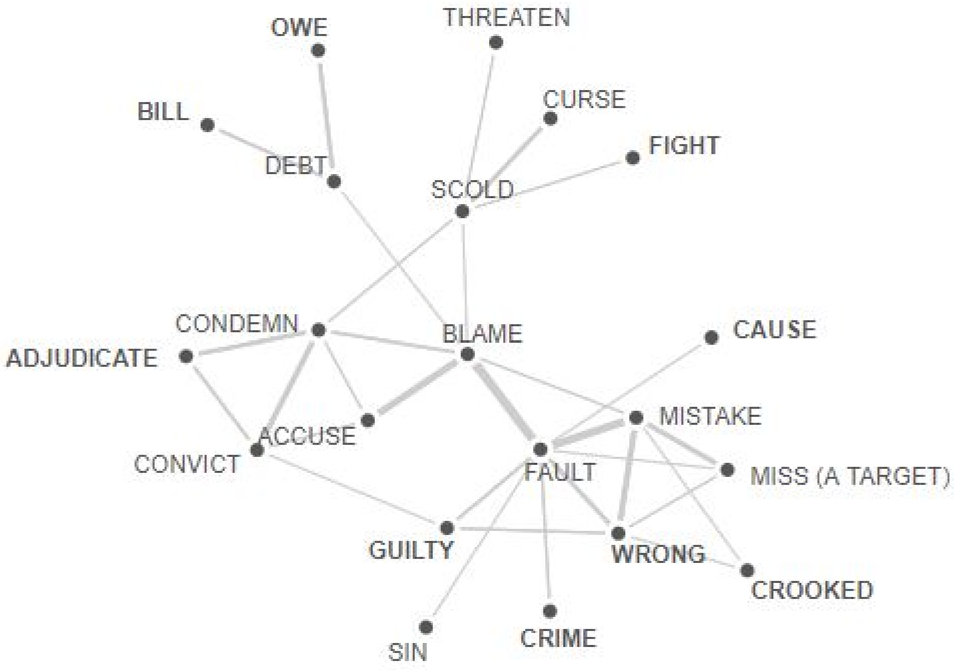

Figure 1: Subgraph for the concept *blame* from CLICS. Screenshot taken from http://clics.clld.org

Figure 1 shows the subgraph for the concept *blame* in the polysemy network CLICS, the older version of the network CLICS$_2$[7] that I am working with in this thesis. We can see that CLICS also uses weighted edges; the link between *blame* and *fault* for instance is thicker than the link between *scold* and *threaten*, which means that the former pair of concepts gets colexified more often than the latter one. The subgraph shows all concepts that are connected to the gloss *blame* with a path length of 2 or less, i.e. every gloss that can be reached from *blame* within a maximum of two links. I will henceforth use the variable l for the path length as different path lengths will play a central role in this thesis.

[6] Dellert (2014)

[7] List, Greenhill, Anderson, Mayer, Tresoldi and Forkel (2018) (To avoid confusion with footnotes, I replace the actual notion 'CLICS^2' with 'CLICS$_2$')

In conclusion, we can state that cross-linguistic colexifications and hence polysemy networks can be used as a valuable resource for universally plausible semantic shifts, which again can be very useful for enhancing historical linguistic research. In the next section I will describe how I try to intertwine polysemy networks with computational tools for historical linguistics.

1.4 Motivation / Application of Polysemy Networks

As already mentioned in the last section, a possible field where polysemy networks might be applied is computational historical lingustics, or to be more precise, automated cognate detection. The problem that all current cognate detection tools have is that semantic similarities aren't taken into account at all because only the words for the same concept are compared to each other. Therefore, any cognate pair where the respective word in language A has a different meaning than its counterpart in language B cannot be detected. Due to that, many quite obvious cognate pairs are not found by automated cognate detection tools. Dellert (2014) found that only one third of expert-judged cognates between Finnish and Hungarian haven't undergone any semantic changes. Typical cognate detection tools therefore would miss around two thirds of the cognate pairs only because the words in question are not compared to each other in the first place. Finnish *ääni* "voice; sound" and Hungarian *ének* "singing; song" are undisputed cognates, but nevertheless, the classical cognate detection method would not be able to find this particular pair.

This is where polysemy networks come into play: The concepts *song* and *voice* are connected by only one edge in CLICS$_2$. Dellert (2014) finds that a network with $l = 2$ already covers 46% more cognate pairs in his data. Münch and Dellert (2015) conducted a similar, yet larger-scale study with similar results. They compared a large network of cross-linguistic polysemies to the catalogue of semantic shifts by the Russian Academy of Sciences.[8] They showed that over a third of the attested semantic changes happens between concepts that are connected to each other over a path length of 1 or 2. This again can be interpreted as a strong evidence that semantic shifts are a lot more likely to happen between semantically related concepts. This leads us back to point made by Croft et al. (2009) that we should not investigate the possibility, but the probability of semantic changes. Both aforementioned studies proposed a path length of 2 as a worthwhile threshold for neighbors that should be taken into account within a polysemy network. Semantic changes that happen over a longer path length are considered to be too vague and erratic due to the sheer amount of concepts that would be considered. Therefore, it is not worthwhile to model them by the approach of semantic similarity that lies within polysemy networks.

In this thesis, I try to include closely related concepts in the respective wordlists. The method I used is quite simple and far from an elaborated approach to systematically enhance automated cognate detection using a model of semantic neighborhood, but it still could give us an impression of how cognate detection tools in their status quo can handle this unusual type of input. As far as I know, there are no studies yet that actually try to use polysemy networks in order to access cognate pairs that could not be automatically detected due to semantic shifts. The closest effort in

[8] Zalizniak (2008)

this direction was made by Steiner et al. (2011) whose approach on cognacy was the other way round. They proposed a pipeline inspired by molecular phylogenetics that would automatize the comparative method. Their approach was to compare words that looked similar in the first place, and then they would use averaged Levenshtein distances of the meanings in order to judge whether a cognacy is likely. With respect to this background in research, this thesis should be understood as a pilot project in this direction and not as a general solution for the problems I have already mentioned.

2 Aims and Methods

The general aim of this thesis is to test to what extent polysemy networks can be intertwined with computational tools for cognate detection. For this purpose, I decided to take a small sample of six languages from the NorthEuraLex database[9] that should not share any true cognates, manipulating the individual wordlists based on the CLICS database[10] and finally running the manipulated input through the cognate detection tool LingPy.[11] I will describe each step and each database more precisely later in this section. The main goal of this thesis will be to inspect if the amount of false positives that is generated by this method is low enough to make it worthwhile to include cross-semantic cognates. To put this survey into a broader context, it could be interpreted as an index for the quite difficult trade-off between allowing semantic vagueness (in order to find more true cognates) and receiving a tremendous amount of false cognates. In order to quantify the results for the previously mentioned sample, I decided to also set up a control sample consisting of six Indo-European languages; the results of those two samples can then be compared with regard to different parameters.

2.1 Language samples

The aforementioned main sample of unrelated languages (which I will be calling S_1 in what follows) consists of Adyghe (Abkhaz-Adyghe, Circassian), Basque (Basque), Chukchi (Chukotko-Kamchatkan, Chukotian), Korean (Koreanic), Lithuanian (Indo-European, Balto-Slavic) and Tamil (Dravidian, South Dravidian).[12] Whilst these languages are relatively safely unrelated to each other, they are also geographically clearly separated (see Figure 2), which at least minimizes the chance that there are shared words due to borrowing.

However, we should keep in mind that all of those languages are spoken in Eurasia, so there is a possibility that there are indeed some true cognates. Many of them can be traced back to *Wanderwörter*, words that had spread over a large geographical area within a short time frame. Typical *Wanderwörter* are inventions in technological fields like agriculture, animal husbandry, mining and other important

[9] Dellert and Jäger (2017)

[10] List, Greenhill, Anderson, Mayer, Tresoldi and Forkel (2018)

[11] List, Greenhill, Tresoldi and Forkel (2018)

[12] Classifications according to Glottolog (Hammarström et al.; 2018)

innovations in human prehistory.[13] There are also well-known instances of younger *Wanderwörter*, such as the (originally) Latin month names and the concept of *tea*, but the majority of them traces back to the Bronze Ages or the early Antiquity.

In the end, this sample is merely one of many possibilities to combine six unrelated and geographically separated languages that are covered in the NorthEuraLex database.

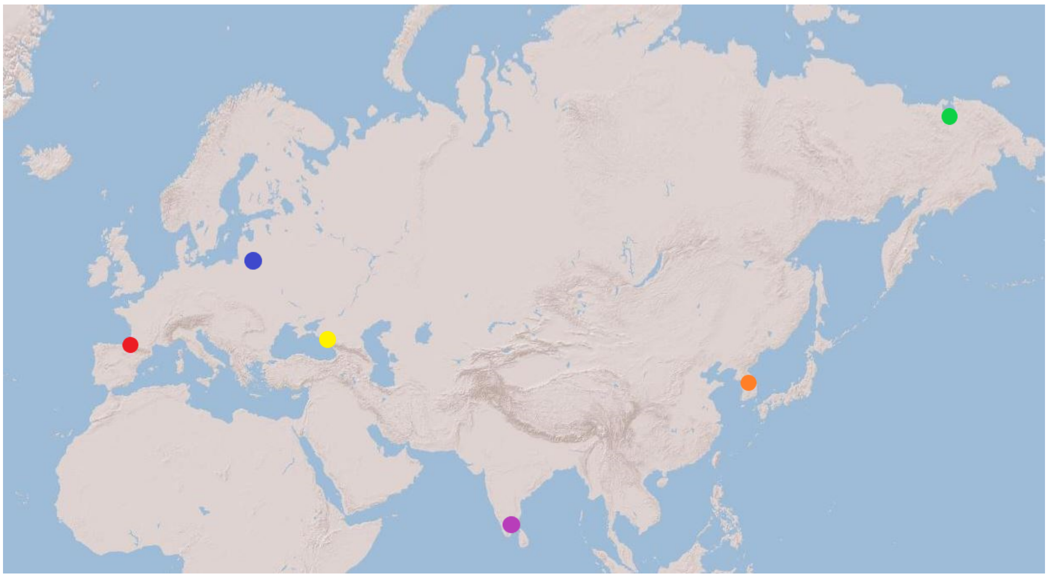

Figure 2: Geographical distribution of S_1

On the other hand, I set up a control sample (henceforth called S_2) of the same size and therefore with the same amount of language pairs. S_2 contains six languages from different branches of Indo-European, namely Albanian (Albanian), Catalan (Italic), Czech (Balto-Slavic), Hindi (Indo-Iranian), Icelandic (Germanic) and Irish (Celtic). The purpose of this sample is to give us a reference value for the results of S_1, so that the results that we get for S_1 can be compared with quantitative methods. Furthermore, we may get an impression whether LingPy works differently on related and non-related samples. We naturally expect that, whatever parameters or path lengths we will be working with, there will be more detected cognates in S_2 than in S_1.

2.2 Databases

Let us now move on to the databases I am working with in this thesis, namely NorthEuraLex and CLICS$_2$. NorthEuraLex is a large-scale lexicostatical database that covers a list of 1,016 concepts in 107 languages, covering more than twenty language families. It has a strong bias towards Indo-European and Uralic languages, which I expect not to matter much for this project as I am only using the two language samples I have described before; however this could have minor influences on the LingPy scorer that has been calculated based on the original NorthEuraLex

[13] Nichols (2003)

list. NorthEuraLex is a good database for large-scale computational investigations due to its wide span of concepts and its uniform IPA encoding for every language in question, thus making it easy to handle and avoiding tiresome data normalization efforts.

In the next step, I established my polysemy network based on the Improved Database of Cross-Linguistic Colexifications (CLICS$_2$). CLICS$_2$ was compiled using 15 wordlist databases – including the NorthEuraLex database – with many different foci, leading to a very wide set of languages from all across the world and from many different language families. Assembling colexifications from such a large-scale and typologically diverse dataset gives us a good idea of which concepts are closely related to each other. As we have already seen, CLICS$_2$ has weighted edges and is hence a weighted network, but I only make use of that in order to exclude false or misleading colexifications – otherwise I am working with an unweighted network whose edges are binary. The network I am using has been calculated with the following parameters:

```
$ clics −t 3 −f families colexification
```

Those parameters secure that only colexifications that are reflected in at least three language families will be included; i.e. if a colexification is only found in less than three families, it will be ignored in this network. This is an important step to exclude erroneous or misleading colexifications that could trace back to errors in the input data or homonymies. It is important to keep in mind that a computationally compiled large-scale network like CLICS$_2$ can't tell true polysemies apart from homonymies that occur due to chance, such as *arm* ("arm" or "poor") in Dutch and Swedish.[14] As we are only interested in those colexifications that occur due to conceptual relations, we need to infer this threshold for the sake of screening our data.

2.3 Manipulation of input data

The next step was to combine both databases and making the data compatible for LingPy, so it could be imported as a LexStat object. LexStat is the particular package within LingPy that is used for the automated cognate detection. After making the data compatible for LexStat, several files were created, extending the originial data by including polysemous concepts. For this purpose, I just acted as if each particular word would have all the meanings that are related to this concept over the desired path length. I did this for $l = 0, 1, 2$. Luckily I could handle the issue that NorthEuraLex is based on German concepts, whereas CLICS uses English concepts, easily because Johannes Dellert provided me with the concept mapping that was used to implement NorthEuraLex into CLICS. Thanks to this concept mapping, I could easily import a dictionary that would translate the English concepts used in CLICS$_2$ to the German glosses from NorthEuraLex and vice versa. Table 2 shows an example of how the manipulated input data that was generated by this script looked like in the end.

[14] Dellert (2014)

DOCULECT	CONCEPT	ORTHOGRAPHY	IPA	TOKENS	GLOSSID
lit	Auge::N	akis	ɐkʲîs	ɐ kʲ î s	10000
lit	Auge::N	akis	ɐkʲîs	ɐ kʲ î s	10174
lit	Auge::N	akis	ɐkʲîs	ɐ kʲ î s	10350

Table 2: Entries for the concept AUGE::N in Lithuanian with $l = 1$

2.4 LingPy

After adjusting the input data sufficiently, the actual analyses on LingPy could be conducted. It is important to keep in mind that we are, in many ways, using the tool in an unexpected or unintended way. In his tutorial on LingPy, List (2017) himself warns that the input data should not have any of the following problems:

- extensive number of synonyms in one language
- multiple variant forms for the same word form
- data merged from different sources without adjusting the phonetic transcription
- mutual coverage below 100 words per language pair

While the latter two problems do not really apply to our data, we do find the former ones on purpose. Already with $l = 1$, our input data is almost three times larger than the original data for the six languages of a sample (i.e. with $l = 0$). Whereas the input data for S_1 with $l = 0$ spans a total of 6,800 entries, it already has 19,304 entries with $l = 1$. This means that for every concept in every language in the mentioned manipulated dataset, we find almost three entries on average; and vice versa for every notion in a language we find an average of roughly three concepts related to it. This number grows exponentially for longer paths – the data for $l = 2$ already spans 92,525 entries which is about 13.5 times the size of the original data.

Another quite uncommon use of LingPy that I applied nevertheless was generating the scorer on another dataset than the one it would be applied on later. I calculated the scorer for the analyses on the NorthEuraLex database in its unmanipulated state, i.e. with all the languages and the correct word lists, just to use it to detect cognates in the different manipulated input datasets. The reason why I had to do this is that I needed a good scorer that would generate reliable results. Such a scorer naturally could only be generated on a good database that lacked the aforementioned problems List warned about. I used 10.000 runs for the calculation of the scorer without preprocessing of the input data. These parameters have shown to generate a quite reliable scorer for the NorthEuraLex database. In order to reuse the generated scorer, I saved the objects LexStat.scorer and LexStat.cscorer using the pickle library. The following code was used to generate the scorer with the described parameters.

```
from lingpy import *
import pickle

nelex = LexStat('nelex-lexstat-fin.csv', check=False)
```

```
scorer = open('scorer.pkl', 'wb')
cscorer = open('cscorer.pkl', 'wb')
nelex.get_scorer(preprocessing=False, runs=10000)
pickle.dump(nelex.scorer, scorer)
scorer.close()
pickle.dump(nelex.cscorer, cscorer)
cscorer.close()
```

After obtaining the scorer that I would use for all the data I will inspect in this thesis, I could extract the cognate pairs that were detected by LingPy for each respective data set. To do so, I had to load the already generated scorer again; then I compiled the supposed cognates using the cluster command from LexStat. The clustering method I used is lexstat, which is a well-rounded method that usually works quite well. Regarding the threshold parameter, I ran the clustering with different thresholds, mainly due to some quite unexpected results for S_1 with $l = 0$, as we will see in the results section. The threshold parameter defines how generous the tool will be in clustering cognates; the higher the value is, the more generous the tool will be. I will henceforth use the variable t to indicate the threshold parameter in the clustering method. I chose to run my analyses with three different values for t, namely 0.7, 0.4 and 0.1. The following code was used for the clustering of cognates (in this example, for the data from S_2 with $l = 0$ and $t = 0.7$).

```
from lingpy import *
import pickle

nelex = LexStat('nelex_ie_l0.csv', check=False)
nelex.cscorer = pickle.load(open('cscorer.pkl', 'rb'))
nelex.scorer = pickle.load(open('scorer_2.pkl', 'rb'))
nelex.cluster(method='lexstat', threshold=0.7,
ref='cognates')
nelex.output('tsv', filename='nelex-ie-clusters-0_tr70',
prettify=False, ignore='all')
```

2.5 Quantitative analysis

After repeatedly running the analyses of the two samplesfor the different values for l and t, the results could be analyzed with different quantitative methods. In order to formalize the results and the parameters, I will introduce C as a set-valued function of detected cognate pairs that takes two arguments: The sample and the threshold it is generated on. The path length again is a variable of the sample itself. That gives us a general notion $C(S_x(l), t)$ where every flexible parameter I made use of is represented. To illustrate what the notion means, $C(S_1(l = 2), t = 0.7)$ is a set of cognate pairs that were found for S_1 with $l = 2$ while using a threshold of 0.7. By the term cognate pair I am formally referring to a 6-tuple that contains the respective concept, the language pair and the orthography as well as the phonetic transcription for both languages involved.

Generally speaking, we are interested in the cognate pairs, not only how many pairs there are in each sample, but also how they are distributed amongst the separate language pairs. Both samples include six languages and therefore contain 15 language pairs. For each C we can hence count how many cognate pairs there were found for each respective language pair. One can expect that these 15 values will follow a normal distribution. If we now extract such a set (which from now on I will note as C_{lpairs}) from two Cs that don't differ in the parameters l and t, but only in the underlying sample, we can compare the results for both samples with an unpaired t-test. This test compares the means of two normally distributed samples in order to determine whether there is a significant difference between these two. Our *null hypothesis* (H_0) is that, regardless of l and p, the mean of $C_{lpairs}(S_2)$ will be higher than the mean of the corresponding set based on S_1, or formally spoken:

$H_0 : m_{C_{lpairs}(S_1)} \leq m_{C_{lpairs}(S_2)}$

This gives us the following *alternative hypothesis*, H_a:

$H_a : m_{C_{lpairs}(S_1)} > m_{C_{lpairs}(S_2)}$

While we will always adapt the same null hypothesis – regardless of l or t – we might expect an approximation of the results to each other with a bigger value for l. Such an approximation can be expected due to the fact that bigger path lengths and therefore larger input data will lead to more falsely detected cognate pairs.

Another way to analyze the results for S_1 and quantitatively compare them to the results for S_2 is to check for how many of the concepts covered in NorthEuraLex there is at least one detected cognate pair. This result will merely be a percentage that indicates how many of the 1,016 concepts do have at least one detected cognate pair, or formally speaking, the number of different concepts in the elements of C divided by 1,016. As the total number of found cognate pairs will rise with higher values for l and t, we can safely predict that this will also be the case for this percentage. Finally, I compared the amount of found cognate pairs to the amount of entries in the respective input data. As the larger input data generated much more word pairs that LingPy can compare to each other, I expect that this ratio will rise with a bigger value for l.

3 Results

3.1 General Overview

S_1	$l = 0$	$l = 1$	$l = 2$	S_2	$l = 0$	$l = 1$	$l = 2$
$t = 0.7$	570	5,269	56,669	$t = 0.7$	894	7,062	72,300
$t = 0.4$	188	2,240	27,921	$t = 0.4$	306	3,076	37,801
$t = 0.1$	119	1,477	19,624	$t = 0.1$	179	2,017	26,847

Table 3: Total of detected cognate pairs for the different samples, thresholds and path lentghs

I want to begin the results section by giving a general overview of how many cognate pairs were detected by LingPy with respect to all the previously presented parameters. Note that when I speak of cognates or cognate pairs, throughout the whole results section I am referring to what LingPy has recognized as such, regardless of their quality, i.e. whether they are reasonable or not. Table 3 shows how many cognates there were detected for each input, or formally speaking, $|C(S_x(l), t)|$ for every possible combination of the different values for x, l and t I used. Those results are not only unexpected, but also quite disappointing: The amount of cognate pairs that were detected for $S_1(l = 0)$ are way too high to be somewhat reasonable. Knowing that already an unmanipulated input of $S_1(l = 0)$ generates very erroneous and misleading results makes the results for $l = 1$ and $l = 2$ almost completely useless, as we can quite safely assume that those results won't be of better quality. Interestingly enough, the most useful results for both samples are the ones that were generated with $t = 0.7$, the lower thresholds exclude some very obvious true cognates while they keep many weird and spurious ones. The strongest evidence for that can be found in S_2, where the words for Tee::N ('tea') in Albanian (*çaj*) and Czech (*čaj*) share the exact same phonetic transcription [t͡ʃaj], yet this pair isn't included in $C(S_2(l = 0), t = 0.4)$. On the other side, a threshold of 0.1 still keeps some very disturbing pairs like Albanian *ylber* [ylbɛr] and Icelandic *regnbogi* [rɛknpɔjɪ] for the concept Regenbogen::N ('rainbow'). Similar examples can be found for S_1: Again for the concept Tee::N, we can find one of the few true cognate pairs of this sample, namely between Adyghe *щай* [ɕaːj] and Chukchi *чай* [t͡ɕaj]. This pair is correctly recognized with a threshold of 0.7, but again disregarded when lowering the threshold to 0.4. The Chukchi word for *tea* interestingly enough is put together with the Lithuanian word *arbata* [ɐrbɐtâ], this highly dubious pair is still considered with a threshold of 0.1. We can generally observe that the differences between $t = 0.1$ and $t = 0.4$ are not that large compared to the differences between $t = 0.4$ and $t = 0.7$. Only relatively few pairs that are found with a threshold of 0.4 are disregarded when lowering the threshold to 0.1.

S_2/S_1	$l = 0$	$l = 1$	$l = 2$
$t = 0.7$	1.57	1.34	1.28
$t = 0.4$	1.63	1.37	1.35
$t = 0.1$	1.50	1.37	1.37

Table 4: Ratio of the total of detected cognate pairs between S_1 and S_2

Table 4 shows the ratio between the two samples in terms of found cognate pairs. The values represent the respective values of Table 3 where the values of S_2 are divided by the ones of S_1. While this ratio gets lower between $l = 0$ and $l = 1$, meaning that $|C(S_1)|$ approximates $|C(S_2)|$, there is hardly any difference between the path lengths of 1 and 2. Only with $t = 0.7$, there is still a slight approximation between those two. Figure 3 again nicely visualizes how this expected approximation alreaedy stagnates between $l = 1$ and $l = 2$.

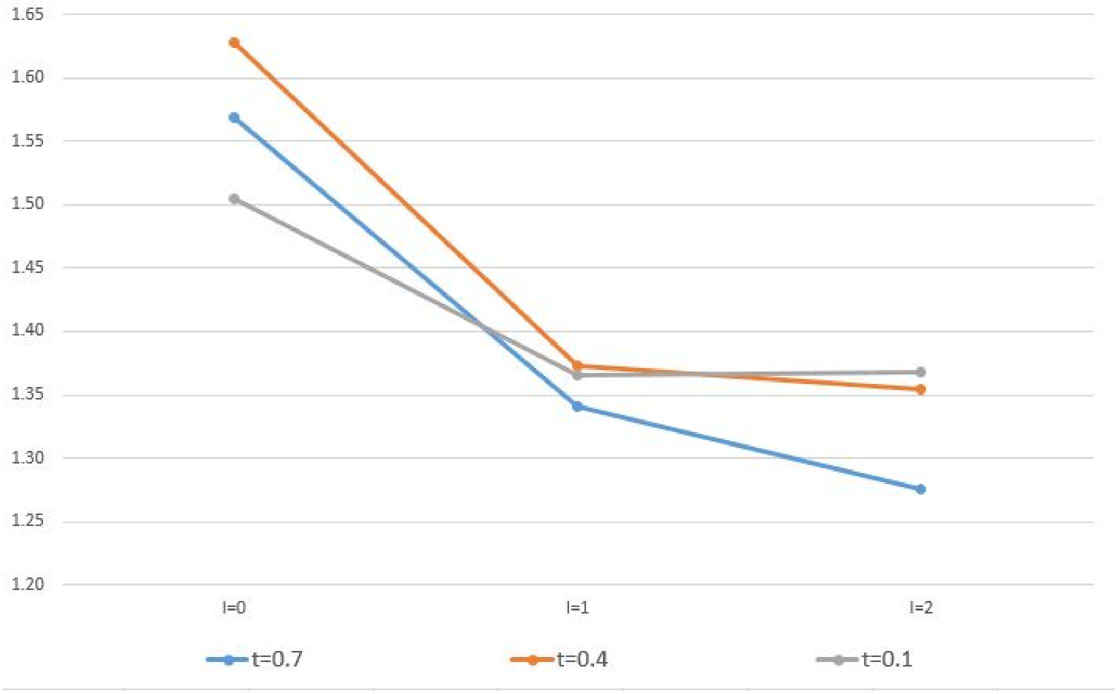

Figure 3: Visualization of Table 4 with respect to different l and t

3.2 Concepts with a detected pair in relation to total concepts

In the next step the results were sorted by concepts, which means that, for each respective C, I inspected for how many concepts there is at least one detected cognate pair, then I set the resultant number in relation to the total of 1,016 concepts that are contained in NorthEuraLex. A general overview of those numbers is given in Table 5 (absolute numbers) and Table 6 (relative to the 1,016 NorthEuraLex concepts).

S_1	$l = 0$	$l = 1$	$l = 2$	S_2	$l = 0$	$l = 1$	$l = 2$
$t = 0.7$	387	667	748	$t = 0.7$	552	789	821
$t = 0.4$	156	487	637	$t = 0.4$	246	588	693
$t = 0.1$	112	423	608	$t = 0.1$	156	509	646

Table 5: Absolute number of concepts for which at least one cognate pair was found

S_1	$l = 0$	$l = 1$	$l = 2$	S_2	$l = 0$	$l = 1$	$l = 2$
$t = 0.7$	38.1%	65.6%	73.6%	$t = 0.7$	54.3%	77.7%	80.8%
$t = 0.4$	15.4%	47.9%	62.7%	$t = 0.4$	24.2%	57.9%	68.2%
$t = 0.1$	11.0%	41.6%	59.8%	$t = 0.1$	15.4%	50.1%	63.6%

Table 6: The numbers of Table 5 in relation to all NorthEuraLex concepts

The biggest cut can be found between $l = 0$ and $l = 1$. Whereas $C(S_1(l = 0), t = 0.7)$ covers 387 concepts (38.1% of all NorthEuraLex concepts), $C(S_1(l = 1), t = 0.7)$

already spans a total of 667 concepts, equaling 65.6% of all given concepts. This is an increase of around 58%; with lower thresholds, this cut becomes even stronger – $C(S_1(l = 1), t = 0.1)$ covers almost four times the concepts compared to its counterpart with $l = 0$. For the results of S_2 this increase of covered concepts is not that drastic, given that there are already many cognate pairs (and therefore many covered concepts) for $l = 0$. Figure 4 illustrates how many concepts both samples cover over different path lengths with $t = 0.7$. It is well visible that there are significant differences for $l = 0$ (38.1% for S_1 opposed to 54.3% for S_2), but with a rising value for l, the values for both samples get closer to each other. $C(S_2(l = 2))$ only covers 9.8% more concepts than $C(S_1(l = 2))$ – for $l = 0$ we find that $C(S_2)$ covers 42.1% more concepts than its S_1 counterpart.

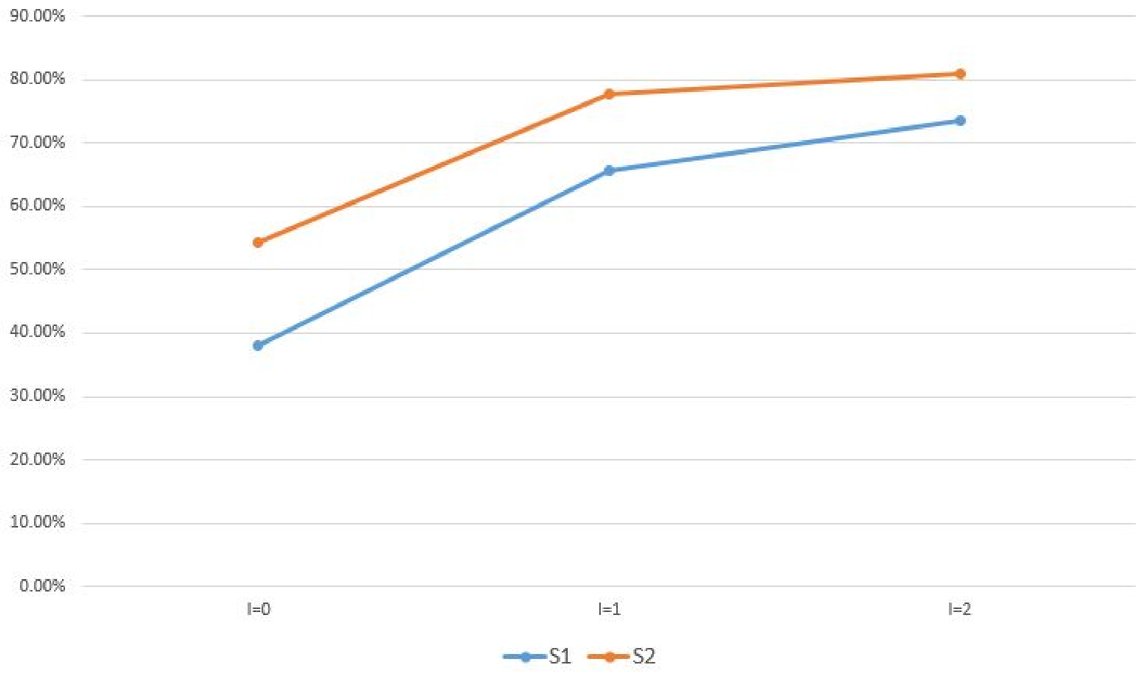

Figure 4: Percentage of the 1,016 NorthEuraLex concepts with at least one detected cognate pair ($t = 0.7$)

In the next step, I split up these results by word class, i.e. how many of the concepts with at least one detected cognate pair were nouns, verbs, adjectives or others. The group "others" contains all the words that aren't classified as nouns, verbs or adjectives. The word classes that are grouped together by this are adverbs, prepositions, pronouns, interrogatives, numbers and conjunctions. This distribution is very similar across both samples and path lengths and generally reflects the distribution of word classes in the NorthEuraLex database quite well. In each case, around half of the covered concepts are nouns, while verbs make up nearly 30% of the concepts. Adjectives and other word classes combine up to around 20% of the concepts, with adjectives making up slightly more than the half of this subgroup. Across all the C samples, a very uniform distribution in terms of word classes can be found. This distribution doesn't significantly differ from the distribution found in the NorthEuraLex database.

Word	S_1		S_2		**North-**
Class	$l = 0$	$l = 2$	$l = 0$	$l = 2$	**EuraLex**
N	207 (53.5%)	386 (51.6%)	286 (51.8%)	404 (49.2%)	480 (47.2%)
V	111 (28.7%)	212 (28.3%)	150 (27.2%)	239 (29.1%)	340 (33.5%)
A	35 (9.0%)	87 (11.6%)	71 (12.9%)	97 (11.8%)	102 (10.0%)
others	34 (8.8%)	63 (8.4%)	45 (8.2%)	81 (9.9%)	94 (9.3%)
total	**387**	**748**	**552**	**821**	**1,016**

Table 7: Distribution of word classes in concepts with at least one detected cognate pair (data for $t = 0.7$)

3.3 Analysis with an unpaired two-sampled t-test

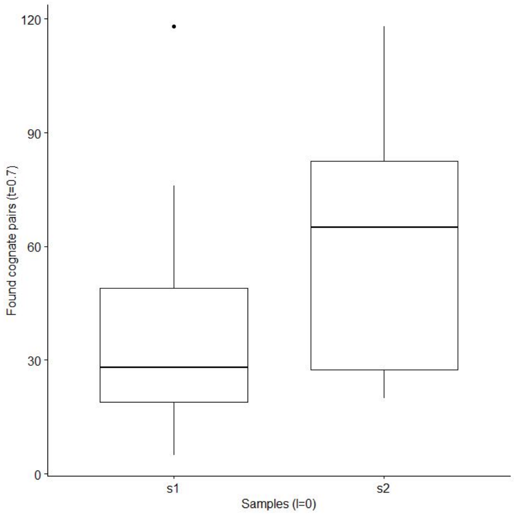

Figure 5: The distributions of $C_{lpairs}(S_{1,2}(l = 0), t = 0.7)$ with their mean values.

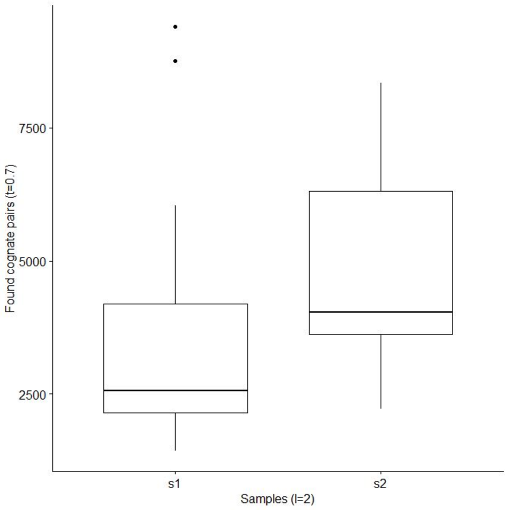

Figure 6: The distributions of $C_{lpairs}(S_{1,2}(l = 2), t = 0.7)$ with their mean values.

In this section, we inspect each $C(S_x(l), p)$ not in terms of the total of found cognate pairs, but in terms of the language pairs for both samples. As both samples contain six languages, for every C there are 15 language pairs. As described previously, I will use each language pair as a datapoint in order to set up new C_{lpairs} samples that can be compared to each other.

In order to check whether those samples are normally distributed I used the Shapiro-Wilk normality test. Note that I only take the results for $t = 0.7$ into account, because they still contain the best results in relation to the other thresholds. The Shapiro-Wilk normality test puts out a normalized *p-value* indicating whether the sample in question differs significantly from a normal distribution ($p \leq 0.05$) or not ($p > 0.05$). Against our expectation, the $C_{lpairs}(S_1)$ samples are not normally distributed, only the $C_{lpairs}(S_2)$ show a distribution that isn't significantly different from a normal distribution. That might indicate that there are artifacts in the results and that they do not actually capture a pluricausal phenomenon. Therefore, we should interpret the results of the *t-tests* at least with a grain of salt, as the mean values of the $C_{lpairs}(S_1)$ samples won't be as meaningful as expected.

Figures 5 and 6 show the distribution of the C_{lpairs}, in each case with $C_{lpairs}(S_1)$ to the left and $C_{lpairs}(S_2)$ to the right. In Figure 5 we see the results for $l = 0$, Figure 6 illustrates the results for $l = 2$. I didn't include a Figure for $l = 1$ because it looked very similar to Figure 6, which again confirms the results I have already presented in the last section, regarding the almost completely stagnant approximation between $l = 1$ and $l = 2$. Seeing these Figures, we might already assume that the $C_{lpairs}(S_1)$ samples are not quite normally distributed. For $l = 0$ we find one language pair that has way more detected cognate pairs than the other language pairs, for $l = 1$ and $l = 2$ there are even two of them. In each case, this extraordinary language pair is Basque/Lithuanian; the follow-up being Korean/Lithuanian.

I conducted an unpaired, two-sampled t-test for each l, only using the results from $t = 0.7$. This test has shown that $C_{lpairs}(S_1(l = 0))$ is significantly different from its S_2 counterpart ($p = 0.03$), this however doesn't hold for the path lengths of 1 and 2, where only the non-significant p-values of 0.10 ($l = 1$) and 0.11 ($l = 2$) were found. This is not that surprising given that $C_{lpairs}(S_1(l = 1, 2))$ contain those aforementioned language pairs with this abundant amount of supposed cognate pairs. Nevertheless it shows how for the data with manipulated path lengths the results for S_1 get closer to the results of S_2 in terms of detected cognate pairs.

3.4 Ratio between detected cognate pairs and size of input

As the final quantitative analysis of my results, I want to compare the amount of found cognate pairs for each C to the size of the input data the respective C was generated on. For example, the input data for $S_1(l = 0)$ spans a total of 6,799 entries (1,016 concepts $*$ 6 languages = 6,096; the remaining 703 entries are made up by synonyms). Table 8 shows this ratio (number of found cognate pairs divided by the number of entries in the input data) for every sample C.

S_1	$l = 0$	$l = 1$	$l = 2$	S_2	$l = 0$	$l = 1$	$l = 2$
$t = 0.7$	0.08	0.27	0.84	$t = 0.7$	0.12	0.33	0.99
$t = 0.4$	0.03	0.12	0.41	$t = 0.4$	0.04	0.15	0.52
$t = 0.1$	0.02	0.08	0.27	$t = 0.1$	0.02	0.10	0.36

Table 8: Ratio of found cognate pairs to input size

A slightly abstract way to understand those numbers is that each value represents how many cognate pairs were found for each entry in the input data. While this number is quite low for the $l = 0$ datasets, it exponentially grows with higher path lengths. This exponential growth can be easily explained by the large growth of word pairs that are compared to each other. To illustrate this, consider the concept AUGE::N ('eye') in S_1: The "clean" input with $l = 0$ has six entries for this concept – exactly one per language – and therefore has 15 cross-linguistic pairs to be judged. Including the closest neighbors ($l = 1$), there are now 30 entries for the concept AUGE::N – four Chukchi entries, six Adyghe entries, five of each other language. This results in 374 cross-linguistic word pairs. This increase of the number of compared pairs can very much explain the growth of the ratio shown in Table 8.

4 Discussion

As I've briefly mentioned previously, the results in general are as surprising as disappointing. $C(S_1(l = 0), t = 0.7)$ containing already 570 spurious cognate pairs could not be expected by any means – those results are generated on a clean sample containing six unrelated languages, the scorer that was used had been well trained on a large database in which many true cognates are included and the clustering method had previously shown to generate quite reliable results and has been recommended by List (2017). While there certainly are clustering methods that work slightly better, as shown by List et al. (2017), the LexStat method I used did not achieve clearly worse results than for example Infomap or UPGMA. Therefore, it is not really understandable how those spurious results were obtained. Although I ignored the recommended threshold of 0.6 (List et al.; 2017), I ran the analysis with one higher threshold and two lower ones, but none of those different thresholds provided satisfying results – the fairly high threshold of 0.7 included very much noise, however the clustering with the lower thresholds lead to even weirder and more implausible results. There were a few clear cognates in S_1 that got disregarded with lower thresholds – and also for S_2, lowering the threshold made the results even worse instead of producing less, but more plausible pairs. I already mentioned the concept Tee::N as an example of how this was the case in both samples.

I manually worked through my results of S_1 with $l = 0$ and $t = 0.7$ to screen them for reasonable cognate pairs, be it due to actual cognacy (which of course includes borrowed words) or to phonetic similarity that can occur by pure chance or because of onomatopoetic etymologies. An example of onomatopoesia is the concept Kuckuck::N ('cuckoo'), for which Adyghe *кукýу* [kʷəkʷə], Basque *kuku* [kuku] and Chukchi *к'эк'к'ук'* [qeqːuq] were clustered together. Those words are certainly not cognates in a sense that they can be traced back to the same word in an ancestor language, nevertheless a sequence comparison method (like LexStat is based on) should recognize the phonetic similarity of those words. With a threshold of 0.4, only the Basque and the Chukchi words are clustered together, whereas there is no cluster at all for this concept with $t = 0.1$. We would expect that from 119 cognate pairs that were found with a threshold of 0.1, at least the Basque-Chukchi pair would be one of them.

An instance of a pair classified as cognate where both forms actually can be traced back to a common ancestor is the Adyghe-Basque pair for April::N, i.e. the month of April. Both words (Adyghe ***апрель*** [aːprajlʲ]; Basque *apiril* [apiɾil]) can be – as well as the German and English words – traced back to Latin *aprilis* without any doubt. This pair also supports my claim that the result get even worse with lower thresholds given that it is not considered anymore when lowering the threshold to 0.4. Another quite similar example for this the Basque-Lithuanian pair for Linie::N ('line'), where both forms, Basque *linea* [linea] and Lithuanian *linija* [lʲînʲɪjɐ], can be traced back to a common Indo-European ancestor, as the Basque word is a clear loanword from Spanish *línea.* Again, this quite obvious pair was not detected with a threshold of 0.4.

On top of that, there were a few Adyghe or Chukchi words that were borrowed from Russian and are therefore cognates to the respective Lithuanian word. In order to completely avoid cognacy in this language sample, Lithuanian as a Balto-Slavic language maybe wasn't the wisest pick.The sample contains two languages that were very heavily influenced by Russian, another Balto-Slavic language. On the other hand, working with a database that "only" spans over Eurasia (and mainly its Northern part), it is nearly impossible to come up with a sample of six languages that don't have any words in common; also handling with some minor noise in the results isn't that much of an issue. The most obvious example I came across while screening my results of a (correctly) detected cognate pair that is due to the Russian influence on Adyghe and Chukchi was the Adyghe-Lithuanian pair for TISCH::N ('table'). The Adyghe word ***стол*** [stʷal] is a clear borrowing from the Russian word that even has the exact same orthography and also a very similar phonetic representation [stol]. The Lithuanian counterpart *stalas* [stǎ˗lɐs] has the same Balto-Slavic root and is correctly clustered together with the Adyghe word – at least for a threshold of 0.7.

Last but not least, I want to briefly comment on some supposed cognate pairs found by LingPy that are somehow plausible, but most likely due to random phonetic similarity. Out of the absurd amount of 570 detected cognate pairs for $S_1(l = 0)$ with $t = 0.7$, there were roughly 35 pairs that I found reasonable, including those mentioned above that are actually cognates. Those reasonable cognate pairs show a uniform distribution regarding the language pairs involved, which speaks for an arbitrariness rather than any sort of underlying structure. Also quantitatively speaking, one had to expect a bunch of arbitrary word pairs with similar phonetic representations, given that there were 1,016 concepts for 15 language pairs, leading to a total of 15,240 word pairs that were compared to each other. With that amount of input data, one should not be surprised to find a pair like Korean 가지 [kad͡ʑi] and Tamil குச்சி [kut͡ɕːi] (for the concept ZWEIG::N, 'branch') that shows an astonishing resemblance, but doesn't allow us to draw any conclusion on the relationship of the both languages. While I can't state with complete certainty that those two words are not related to each other, I at least consider it highly likely because there is no "mediator" language (some Chinese or Northern Indian language) that has a similar word form and it is very unlikely for Tamil to directly borrow a word from Korean or vice versa. Therefore it is very safe to assume that both forms developed separately from each other and only resemble each other due to pure chance. A very similar example from my results would be the Adyghe-Basque pair for TAG::N ('day') with the Adyghe word ***шхэн*** [ʃxan] being very similar to Basque *jan* [d͡ʒan].

Now one would expect to find at least those pairs with such an obvious phonetic resemblance in the results for the lower thresholds. But then again, the answer is no. Both examples mentioned above, as well as the significant majority of somewhat reasonable pairs, are not included in the results for $t = 0.4$. This obviously leads to the question, which pairs are then found with the lower thresholds, if not those that I would have expected due to phonetic similarity or actual cognacy. At first glance, the results for $t = 0.4$ and $t = 0.1$ seem completely arbitrary; there are hardly any pairs that appear to have any phonetic similarity at all. But after having another look on the results for the lower thresholds, I made an interesting find: Nearly all of the pairs in $C(S_1(l = 0), t = 0.1)$ are between Lithuanian and another language; for $t = 0.4$ this is not as evident yet, but still clearly visible.

Figure 7 clearly shows a bias towards Lithuanian for all the thresholds, this bias

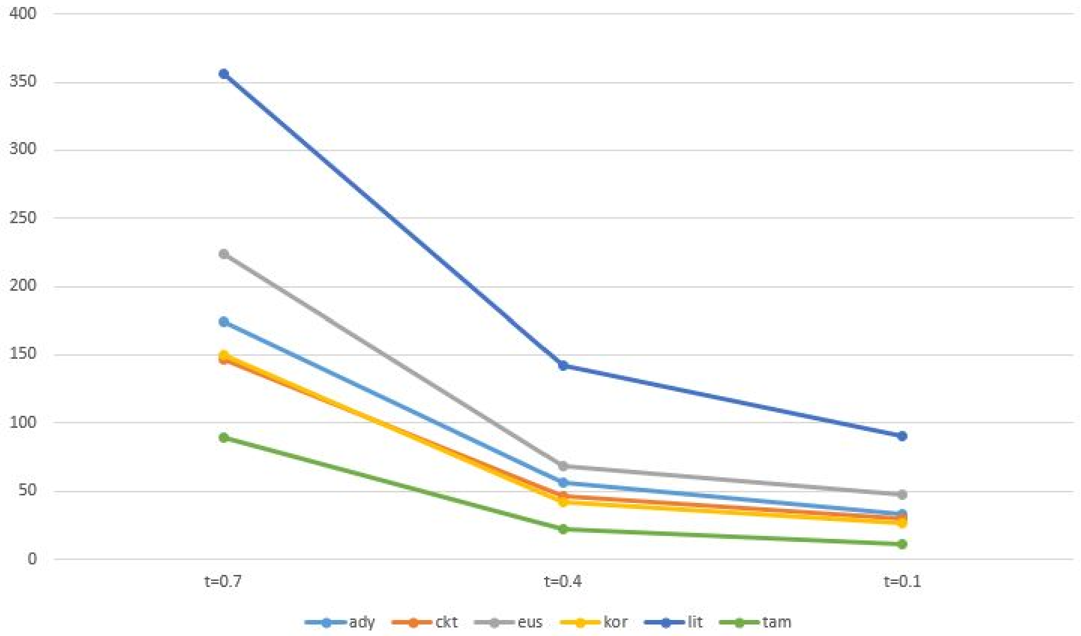

Figure 7: Detected cognate pairs for $S_1(l = 0)$ by language

however gets stronger for lower thresholds – which might not be as clear in the figure as it actually is: While Lithuanian is involved in 62.5% of the pairs found with $t = 0.7$, we find Lithuanian in 91 of the 119 pairs for $t = 0.1$, equalling 76.5%. For the intermediate threshold of 0.4, this number lies at 75.5%.

Then I checked the respective results for S_2 and found a very similar tendency, as shown in Figure 8. In this case, it is the Czech language that is involved in the majority of cognate pairs – more precisely, in 51.5% of the pairs for $t = 0.7$, 70.6% for $t = 0.4$ and 76.5% for $t = 0.1$. These numbers, especially for the two lower thresholds, are very similar to the findings for the role of Lithuanian in S_1; furthermore both graphs look almost identical, i.e. they have the same distributions with respect to the languages involved in the detected cognate pairs. The interesting part in that is naturally the role of Lithuanian in S_1 and respectively of Czech in S_2 – the other five languages in both samples show a seemingly normal distribution that could have been expected in a form like this. The main question now is why LingPy seems to prefer one particular language in each sample in a way that it considers very spurious pairs to be cognates, especially with lower threshold values. The first observation in this question is that both Lithuanian and Czech are Balto-Slavic languages. While this could be due to pure chance, it would certainly be an interesting approach to check if LingPy maybe has some kind of weird bias towards Balto-Slavic languages; at least in the way I used it. In order to do so, I generated two additional samples following the same logic as the two samples I've been working with – one sample[15] that imitates S_1 with six completely unrelated

[15] This sample will be called S_3 and includes Avar (Nakh-Daghestanian, Daghestanian), Buriat (Mongolic, Eastern Mongolic), Croatian (Indo-European, Balto-Slavic), Kannada (Dravidian, South Dravidian), Northern Yukaghir (Yukaghir) and Tundra Nenets (Uralic, Samoyedic).

and geographically separated languages, and another one[16] with six Indo-European languages from different branches. Both of these test samples again included one Balto-Slavic language.

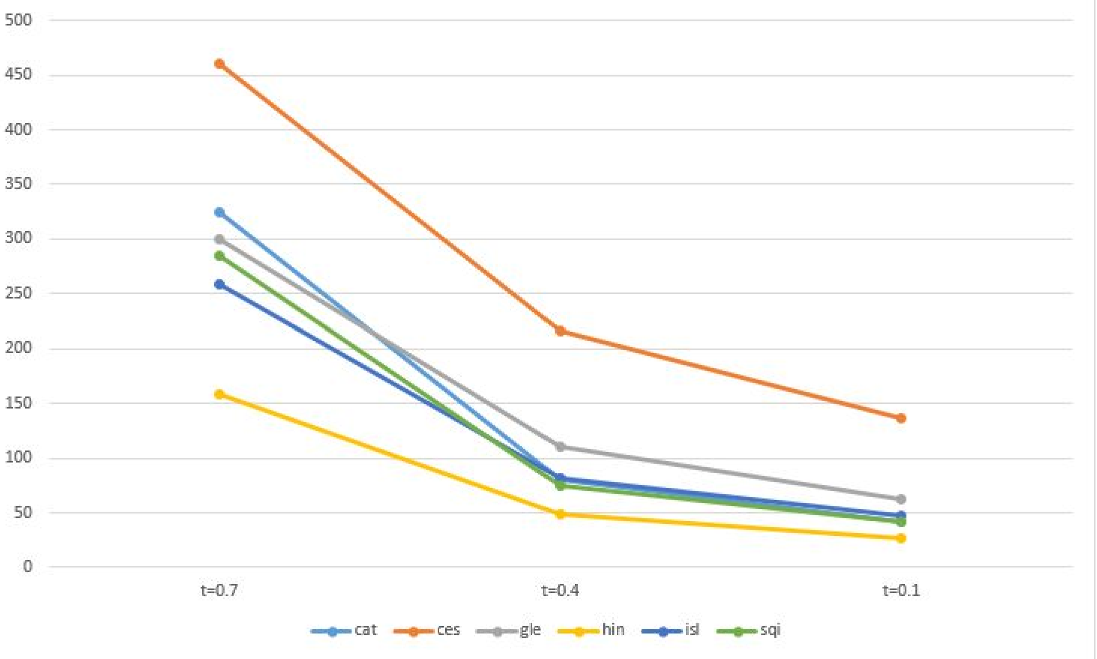

Figure 8: Detected cognate pairs for $S_2(l = 0)$ by language

This analysis for S_3 and S_4 showed similar results, but in both cases, there were two outlier languages rather than just one. Again, there is a base of (in this case four) languages where the particular languages are relatively close to each other in terms of found cognate pairs and where the values seem to be normally distributed; but then again on the other hand, there are some languages that LingPy seems to prefer, or to judge differently. Those outlier languages were Buriat and Northern Yukaghir for S_3, as for S_4 we're dealing with Farsi and Modern Greek. It is important to notice that those outlier statuses are not mainly due to the amount of cognate pairs found between the two respective languages, but rather due to the general cognacy judgement between those languages and the others. For instance, Italian is involved in 95 detected cognate pairs for $t = 0.1$ – 60 of them between Italian and Farsi, another 19 between Italian and Greek and only 16 between Italian and one of the remaining three languages. In S_3, with a threshold of 0.1 Kannada only shares one supposed coganate pair with Croatian, and not a single one with Tundra Nenets and Avar – yet, LingPy managed to find 18 cognate pairs between Kannada and Buriat as well as 10 pairs between Kannada and Northern Yukaghir. This illustrated asymmetry is ubiquitous throughout all four samples I ran this analysis on: LingPy's clustering algorithm – for whatever reason – seems to judge cognacy differently for some languages than for others. The two control samples on the other hand side showed that it is seemingly arbitrary which family and/or branch a language belongs

[16] This sample will be called S_4 and contains Farsi (Indo-Iranian), Italian (Italic), Modern Greek (Greek), Norwegian Bokmål (Germanic), Ukrainian (Balto-Slavic) and Welsh (Celtic).

to, the assumption that there could be some correlation with this phenomenon to Balto-Slavic languages was denied by the two latter samples. Taking a short look into the results from the control samples, we find again – as already mentioned for S_1 and S_2 – that the supposed cognate pairs that involved one outlier language were generally way less plausible than those that were found between two of the languages that behaved normally.

As I don't know how the clustering algorithms of LingPy exactly work, I can only speculate over potential reasons for this odd scheme. LingPy usually works quite well with the parameters I used for the generation of the scorer and the clustering; so I would expect the problem to be somewhere else. A possible case is that there was a problem with applying the scorer to different data from the one it has been trained on. The usual workflow for LingPy would be to obtain a scorer and use it for the same input data, I however trained the scorer on the unmanipulated NorthEuraLex database, saved it and then imported it again to use it on my manipulated input data. Without any deeper knowledge on LingPy's algorithms, I do not want to disregard the possibility that LingPy doesn't quite know how to work with an external scorer and that the unexpected results arise from this problematic intersection. Another possibility we must of course not forget is that there is some data related to the scorer that I overlooked and therefore forgot to include; although I tried to make sure that I took care of every relevant file.

Another issue that seems to be the case is that LingPy has a strong bias towards input data where there are many actual cognates. Considering the results found for S_2 with a threshold of 0.7, while it is true that there were some weird pairs (mainly including Czech), many of the detected pairs were indeed very plausible. For lower thresholds, in both samples there was an apparent tendency that LingPy mainly held on to bad pairs while disregarding good ones. So generally speaking, the results that were obtained with $t = 0.7$ were still the best ones with respect to their quality; but then again we need to take into account that with these parameters, there is a total of 387 cognate pairs that was detected for $S_1(l = 0)$ – which obviously is a way too high number and around 70% of the amount of cognate pairs found for S_2 where all the six languages are related to each other. Given that, I assume that LingPy (or at least the clustering algorithm LexStat I used) was mainly trained on input data with many true cognates, which leads to this mentioned bias. Now when it is exposed to an input data where there are almost no cognates at all, it becomes way too generous in its judgement and clusters words together that are only remotely similar, just because it assumes that there must be some cognates in the input data.

This severely shifts the focus of this thesis, as it is impossible to answer the initial question – we can't really judge how well the automated cognate detection worked with input data that was aggregated with related concepts. We can only try to deduce how well it could have worked if the results were reliable, which hardly can be based on empirical data but mainly on speculation. One expected pattern that could indeed be observed – despite the bad quality of the results – is the convergence of found cognate pairs with higher path lengths for both samples. While it can't be stated for sure that this will also be the case with more reliable results, it does indicate that this assumption is true.

However, in order to really discuss about the main initial question whether polysemy networks can be used to enhance automated cognate detection, we would need reliable results in the first place. Therefore, one would have to find a way to obtain good results from the initial, unmanipulated data samples; and once this can be assured, the next would be to manipulate the input data making use of polysemy networks. Whether there was an error in the application, whether LingPy isn't just capable of handling completely unrelated languages yet or whether it doesn't work well with an external scorer that was trained on data that is different from the data it is being used on, fact is that the results I obtained in this thesis don't work as an adequate base for discussing how worthwhile it is to include polysemy networks to a certain extent.

5 Conclusion

Concluding this thesis, the main statement that can be made is that the obtained results were unexpected in such a way that they did not only shift the focus of the discussion, but they also made it nearly impossible to answer the questions that arose in the beginning. My goal was it to implement polysemy networks in wordlists in a simple and not quite elaborated manner, as a sort of a pilot project in order to see whether it would be worthwhile to develop more elaborated methods that aim in a similar direction. This was based on the assumption that the automated cognate detection tool that I used would provide reliable results, which clearly wasn't the case. With respect to this, this thesis didn't answer any questions, but in contrast just generated more questions that could not be answered.

Nevertheless, I do consider the present approach itself not worthless at all. Polysemy networks have indeed proven to be a very useful approach to model semantic shift and therefore are a very valuable resource in historical linguistics. Just the information of how likely a certain change in meaning is to happen is very useful for the reconstruction of cognacies – for expert judgement as well as for automated methods – and in the next step for establishing relations between languages or language families. Therefore, one would need to find a method that provides good and reliable cognacy judgement and clustering, independent of the structure or the size of the input. Once those circumstances are given, there is a great potential for future research in historical linguistics aiming to take polysemy networks – and therefore lexical change – into account.

References

Anttila, R. (1972). *An Introduction to Historical and Comparative Linguistics*, New York: Macmillan.

Brown, C. H., Holman, E. W., Wichmann, S. and Velupillai, V. (2008). Automated classification of the world's languages: a description of the method and preliminary results, *STUF-Language Typology and Universals Sprachtypologie und Universalienforschung* **61**(4): 285–308.

Campbell, L. and Poser, W. J. (2008). *Language classification: History and method*, Cambridge: Cambridge University Press.

Croft, W., Beckner, C., Sutton, L., Wilkins, J., Bhattacharya, T. and Hruschka, D. (2009). Quantifying semantic shift for reconstructing language families, *83rd Annual Meeting of the Linguistic Society of America, San Francisco*, pp. 8–11.

Dellert, J. (2014). Evaluating cross-linguistic polysemies as a model of semantic change for cognate finding, *Workshop on semantic technologies for research in the humanities and social sciences (STRiX), Gothenburg*.

Dellert, J. (2017). *Information-Theoretic Causal Inference of Lexical Flow*, PhD thesis, University of Tübingen.

Dellert, J. and Jäger, G. (2017). NorthEuraLex 0.9. (Available online at http://www.northeuralex.org).

Dolgopolsky, A. (2008). *Nostratic Dictionary*, Cambridge: McDonald Institute for Archaeological Research.

Greenberg, J. H. and Ruhlen, M. (2007). *An Amerind Etymological Dictionary*, Stanford: Stanford University.

Hammarström, H., Forkel, R. and Haspelmath, M. (2018). Glottolog 3.3. Jena: Max Planck Institute for the Science of Human History. (Available online at http://glottolog.org, Accessed on 2019-03-20.).

Hjelmslev, L. (1961). *Prolegomena to a Theory of Language*, Madison and Milwaukee: The University of Wisconsin Press.

Hock, H. H. (1986). *Principles of historical linguistics*, Berlin: Mouton de Gruyter.

Jäger, G. (2015). Support for linguistic macrofamilies from weighted sequence alignment, *Proceedings of the National Academy of Sciences* **112**(41): 12752–12757.

List, J.-M. (2017). Sequence Comparison with LingPy. Availabe online at http://lingulist.de/tutorials.html.

List, J.-M., Greenhill, S., Anderson, C., Mayer, T., Tresoldi, T. and Forkel, R. (2018). CLICS^2. An improved database of cross-linguistic colexifications assembling lexical data with help of cross-linguistic data formats, *Linguistic Typology* **22**(2): 277–306.

List, J.-M., Greenhill, S. and Gray, R. D. (2017). The potential of automatic word comparison for historical linguistics, *PLOS ONE* **12**(1): 1–18.

List, J.-M., Greenhill, S., Tresoldi, T. and Forkel, R. (2018). LingPy. A Python library for historical linguistics. Version 2.6.4. URL: http://lingpy.org, DOI: https://zenodo.org/badge/latestdoi/5137/lingpy/lingpy. With contributions by C. Rzymski, G. Kaiping, S. Moran, P. Bouda, J. Dellert, T. Rama, F. Nagel. Jena: Max Planck Institute for the Science of Human History.

List, J.-M., Terhalle, A. and Urban, M. (2013). Using network approaches to enhance the analysis of cross-linguistic polysemies, *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013)–Short Papers*, pp. 347–353.

McMahon, A. M. S. (1994). *Understanding language change*, Cambridge: Cambridge University Press.

Münch, A. and Dellert, J. (2015). Evaluating the potential of a large-scale polysemy network as a model of plausible semantic shifts, *6th Conference on Quantitative Investigations in Theoretical Linguistics (QITL-6), Tübingen.*

Nichols, J. (2003). The epicentre of the Indo-European linguistic spread, *in* R. Blench and M. Spriggs (eds), *Archaeology and Language I*, London and New York: Routledge.

Nichols, J. (2010). Macrofamilies, Macroareas, and Contact, *in* R. Hickey (ed.), *The Handbook of Language Contact*, Chichester: Wiley-Blackwell.

Perrin, L.-M. (2010). Polysemous qualities and universal networks, invariance and diversity, *Linguistic Discovery* **8**: 1–22.

Ringe, D. A. (1992). On calculating the factor of chance in language comparison, *Transactions of the American Philosophical Society* **82**(1): 1–110.

Steiner, L., Cysouw, M. and Stadler, P. (2011). A pipeline for computational historical linguistics, *Language Dynamics and Change* **1**: 89–127.

Traugott, E. C. and Dasher, R. B. (2002). *Regularity in Semantic Change*, Cambridge: Cambridge University Press.

Zalizniak, A. A. (2008). A catalogue of semantic shifts: Towards a typology of semantic derivation, *in* M. Vanhove (ed.), *From polysemy to semantic change. Towards a typology of lexical semantic associations*, Amsterdam and Philadelphia: John Benjamins, pp. 217–232.

A Declaration of Authorship

I hereby declare that I am the sole author of this bachelor thesis and that I have not used any sources other than those listed in the bibliography and identified as references. I further declare that I have not submitted this thesis at any other institution in order to obtain a degree.

A.

Tübingen, 16th of August 2019