Západočeská univerzita v Plzni
Fakulta filozofická

Bakalářská práce

2021 **Karel Beneš**

Západočeská univerzita v Plzni
Fakulta filozofická

Bakalářská práce

Processing of translations between languages: software methods, artificial intelligence and their advantages and disadvantages

Karel Beneš

Plzeň 2020
Západočeská univerzita Plzni
Fakulta filozofická

Katedra anglického jazyka a literatury

Studijní program Filologie

Studijní obor Cizí jazyky pro komerční praxi

Kombinace angličtina – němčina

Bakalářská práce

Processing of translations between languages: software methods, artificial intelligence and their advantages and disadvantages

Karel Beneš

Vedoucí práce:

Bočková Renata, Mgr.

Katedra anglického jazyka a literatury

Fakulta filozofická Západočeské univerzity v Plzni

Plzeň 2021

Prohlašuji, že jsem práci zpracoval (a) samostatně a použil (a) jen uvedených pramenů a literatury.

Plzeň, Květen 2021 ……………………….

Poděkování

Rád bych poděkoval Mgr. Tomáši Hostýnkovi a Mgr. Renatě Bočkové z katedry anglického jazyka a literatury Fakulty filozofické Západočeké univerzity za ochotu při výběru tématu, vstřícnost při vedení bakalářské práce a panu Janu Švecovi z výzkumného centra NTIS – Nové technologie pro informační společnost při Fakultě aplikovaných věd Západočeské univerzity za poskytnutí cenných poznatků, tipů a zdrojů.

Table of contents

1 Introduction

The field, nowadays known as mathematical linguistics (although it has existed only since 1950s), has been known and studied since antiquity and in the last decades, it experiences rapid development in the form of electronic vocabularies, internet translators, language databases, spell checks in text processors etc. Mathematical linguistics is the field between mathematics and linguistics (and in the last years also informatics via the usage of artificial intelligence – AI). It uses many branches of mathematics, such as probability theory, graph theory, statistical methods or set theory. They will be described in this thesis in more detail.

The objective of this thesis is to describe the usage of mathematical and computer methods in linguistics with emphasis on translation between languages by using mathematical and statistical methods and software tools, and I will also explore their advantages and disadvantages.

I have chosen this topic, because I study foreign languages and they are along with mathematics and computers my hobby and connection of mathematics, linguistics and informatics brings a great potential for all three subjects.

In this thesis I will draw on these sources:

Sedlačíková, Blanka: *Historie matematické lingvistiky*; this book describes on its 160 pages the development of mathematical linguistics from antiquity to the struggle for Manuscripts

Partee, Barbara H.; Meuen, Alice Ter; Wall, Robert E.; *Mathematical Methods in Linguistics*; this book describes all parts of mathematics and linguistics important for mathematical linguistics

Pošta, Miroslav; *Technologie ve službách překladatele*; Miroslav Pošta – Apostrof; 1st edition; 2017; this book describes several software methods used by translators from the user´s point of view.

Čermák, František; *Korpus a korpusová lingvistika; Nakladatelství Karolinum; 1st edition; 2017; this book describes using of corpuses and corpus linguistics in more detail.*

Pořízka, Petr; *Tvorba korpusů a vytěžování jazykových dat metody, modely, nástroje*; Nakladatelství Filozofické fakulty Univerzity Palackého; 1st edition; 2014; this book describes software methods and their inner workings from the user´s point of view.

Vopěnka, Petr; *Vyprávění o kráse novobarokní matematiky*; 2nd publication; Práh; 2016; this book describes the set theory and its historical and philosophical context. context

Vopěnka, Petr; Úvod do klasické teorie množin; Nakladatelství Západočeské university v Plzni, Nakladatelství Fragment; 1st edition; 2011; this book also describes the set theory, but without the detailed historical and philosophical context.

Weaver, Warren; Shannon, Claude Elwood; *Mathematical Theory of Communication*; published in 1949; 1st edition; this book means the introduction of mathematical theory of information.

Bojar Ondřej, Rosa Rudolf, Tamchyna Aleš; *Chimera – Three Heads for English-to-Czech Translation*; this paper is about a project from the Faculty of Mathematics and Physics from the Charles University, Prague.

Singh, Shashi Pal; Kwnar, Ajai; Darbari, Hemant, Singh, Lenali; Rastogi, Anshika; Jain, Shikha; *Machine Translation using Deep Learning: An Overview*; This paper describes deep learning in the translation process

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, Jeffrey Dean; *Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation*; This paper describes the use of neural networks in the translation process

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, Jeffrey Dean; *Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation*; This paper describes machine translation system using neural networks by Google, bringing just one little modification to the previous system, but with huge impact

Kyunghyun Cho, Bart van Merrienboer Caglar Gulcehre, Fethi Bougares, Holger Schwenk, Dzmitry Bahdanau, Yoshua Bengio; *Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation*; This paper describes using encoders and decoders in the machine translation

Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan N. Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer and Jakob Uszkoreit; *Tensor2Tensor for Neural Machine Translation*; This paper describes using of one technology in the neural machine translation.

Popel, Martin, Tomková, Markéta, Tomek, Jakub, Kaiser, Lukasz, Uszkoreit, Jakob, Bojar, Ondřej, Žabokrtský, Zdeněk; *Transforming machine translation: a deep learning system reaches news translation quality comparable to human professionals*; This paper describes machine translation programme with multilayer encoders and decoders

Marta R. Costa-jussà, Alexandre Allauzen, Loïc Barrault, Kyunghun Cho, Holger Schwenk; *Introduction to the special issue on deep learning approaches for machine translation*; This paper describes using of deep learning in machine translation

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, Marc'Aurelio Ranzato; UNSUPERVISED MACHINE TRANSLATION USING MONOLINGUAL CORPORA ONLY; This paper describes another innovative approach to the machine translation – usage of the monolingual corpora only

Philipp Koehn, Rebecca Knowles; *Six Challenges for Neural Machine Translation*; This paper describes 6 weaknesses of current neural machine translation

Zákon č. 121/2000 Sb., Hlava I, Díl 5, §43 - this law is important for translators, because it forbids them breaking the electronic protection of translated texts

András Farkás; *LF Aligner*; this is one of the programmes used by translators and linguists

This bachelor´s thesis will have the following structure: In the first part a short introduction into the subject of the study of the mathematical linguistics and its compartments will be described. In the second part we will describe its history. In the third part we will describe modern mathematical and computational means of translation. And the last part will be the practical part where we will test some software tools and methods for translation and look for some advantages and disadvantages and also for some tips for the authors of these programmes.

2 The theory of mathematical linguistics

We can divide the mathematical linguistics into 3 branches according to the used mathematical means:

1. Quantitative linguistics (or statistical linguistics) – uses quantitative mathematical methods such as mathematical statistics or probability theory.
2. Algebraic linguistics – uses non-quantitative mathematical methods such as set theory or graph theory to description of formal languages and grammars.
3. Computational linguistics – uses computers and methods of informatics for studying languages. Since 1990s is also rapidly developing the so-called corporal linguistics, which studies languages with the help of wide files of language data stored on hard drives. Near to the computational linguistics exists also language engineering – this involves algorithms for natural language description and different software tools for machine translation, keeping and looking for information, grammatical correctors…

Sedlačíková claims in her book (2012), that mathematical linguistics as a separate discipline can be heard since the end of 1950s and the start of 1960s. Its true origins are however unclear, because, as Sedlačíková in her book further claims, the beginning of this branch of science is usually dated in 1957 and the 8th international linguistical congress in Oslo, but the activity in this area can be found earlier in America, Europe, Far East or the Soviet Union.

From the beginning there was a certain terminological inconsistency reflecting the delimitation of the mathematical linguistics as a whole, but also of its components. Mathematical linguistics does not try to replace the linguistics itself. The task of the mathematical linguistics is to describe the natural language by using of mathematical methods, to seek new questions, to confirm the results of linguistic research by using exact mathematical methods and also automatic disambiguation.

2.1 Quantitative linguistics

This term describes the part of mathematical linguistics using the quantitative mathematical methods. Quantitative linguistics has become a huge impulse at the end of 1940s in the work of C. E. Shannon and N. Wiener *Mathematical theory of information*. This theory studies quantitative characteristics of communication systems and has important practical applications (code economy, code resistance etc.)

Statistics is the most common mathematical method used in quantitative linguistics. It is the reason why the quantitative linguistics is sometimes in literature also called statistical linguistics and why termini such as phonological statistics, morphological statistics, lexical statistics or stylistic statistics.

Of all three branches of mathematical linguistics, quantitative linguistics has the longest tradition. The methods of quantitative linguistics had been used long before the term mathematical linguistics has been established. The basic role in its history plays the term frequency for practical reasons (for example building of the Morse code, learning of foreign languages). The German stenograph F. W. Käding wrote in 1897 the first frequency dictionary, Häufigkeitswörterbuch der deutschen Sprache. According to the findings of the Russian mathematician A. A. Markov from his statistical analysis of Eugene Onegin from 1913, it is possible to predict the probability of occurrence of consonants and vowels. The American linguist G. K. Zipf has in the 1920s and 1930s noticed some general relationships in natural languages based on the term frequency (so called Zipf´s laws) (Sedlačíková, 2012).

2.2 Algebraic linguistics

This is the branch of linguistics using the non-quantitative mathematical methods such as algebra, graph theory, set theory, topology or combinatorics. The formation of the algebraic linguistics started in the second half of 1950s, mainly to fulfill the needs of the machine translation. Algebraic linguistics is based on the generative and transformational grammar by Noam Chomsky, reconnaissance and categorial grammar by Y. Bar-Hillel, applicational-generative model of the language by S. K. Šaumjan, analytical models of the language or dependency grammar.

The edition of the book *Syntactic Structures* by Noam Chomsky in 1957 and the introduction of the first variant of the generative or transformational theory were the meaningful milestone for the development of the algebraic linguistics. According to this theory, the language is understood as a creative process, in which the sentences are generated by the final number of transformational rules from the kernel sentences. The number of possible sentences is infinite. This version of generative and transformational linguistics was purely formal, the second version had also a semantic part.

The opposite of the generative and transformational grammar is the reconnaissance and categorical grammar by Y. Bar-Hillel. It transforms the sentence into the chain of symbols and reveals relations between those symbols and determines the grammatical correctness of the sentence.

Applicational-generative model includes the elements of structuralism and generative grammar. The language units are tagged with symbols and are derived using the so-called application (the method of mathematical logic, which relates to the relations between symbols).

The opposite to the aforementioned grammars (generative and transformational, reconnaissance and categorical, applicational-generative) are the analytical models created mainly by the Soviet grammarians, mathematicians and linguists, such as O. S. Kulaginova, R. L. Dobrushin, A. N. Kolmogorov, S. J. Fitialov, A. V. Gladkij or I. I. Revzin (Sedlačíková, 2012).

The starting term in this theory is the set of grammatically correct sentences. In these models has the set theory been used for the first time in the linguistics (Sedlačíková, 2012).

2.3 Computer linguistics

The computer linguistics has been in development since the end of 1950s alongside with the development of cybernetics, computer technology, quantitative and algebraic linguistics and other border fields. It researches the computer processing of the natural language, machine translation. To other problems solved by computer linguistics belong keeping and searching the information or working with corpora, which means with large files of language data.

Machine translation can be understood as translation between languages with the help of a machine. The machine translation has several problems – for translation is needed not only grammar building, but also the semantics. The semantics has not been processed enough to be formalized. Furthermore, it places significant demands on the memory and power of the computer.

3 Mathematic formulas created for translators and linguists and their history

3.1 Frequency, frequential and concordance dictionaries, stenography

Frequency determines the number of the particular phenomenon in the set of phenomena. *Frequential dictionary* is a list of words supplemented by their frequency calculated from wide representatively selected material. As a general rule, the expressions in the frequential dictionary are sorted in alphabetical or frequential order. Frequential dictionaries supplemented by information about the place of the word in excerpted text are called *concordance dictionaries*. Frequential and concordance dictionaries are used for studying the system and structure of literary works, functional styles etc., for statistical purposes, machine learning and as a source of information for linguists, mathematicians, cryptologists etc.

The stenography is a system of symbols that enables to write in the speed of the spoken language. The stenographical systems are nowadays divided into two groups:

1. Geometric systems, where the basic geometric shapes (straight line, circle etc.) and their combinations are being used as the symbols.
2. Italic systems, where the more complicated geometric shapes are being used

3.2 Andrey Andreyevich Markov

Andrey Andreyevich Markov was a Russian mathematician and author of article *Primer statističeskogo issledovanija nad tekstom „Jevgenija Onegina“ illjustrirujuščij svjaz ispytanij v cep* describing statistical analysis of the occurrence of consonants and vowels and subsequent application of the theory of Markov chains. In this article Markov mathematically confirmed the conformity of the observation of sounds with the hypothesis of the existence of a simple chain dependence.

3.2.1 The phonetic statistics of Eugene Onegin

Let´s have a chain of 20 000 Russian voices without hard barley and soft barley, where every voice can be vowel or consonant. Let´s assume that there is a constant probability p, that any given voice is vowel. This probability may be estimated with the help of the number of vowels in the text the same way as the following probabilities:

p_1	Vowel follows the vowel
p_0	Vowel follows the consonant
$p_{1,0}$	Vowel follows the consonant that follows vowel
$p_{0,1}$	Vowel follows the vowel that follows consonant
$p_{0,0}$	Vowel follows two consonants
$p_{1,1}$	Vowel follows two vowels
q	Given voice is consonant
q_1	Consonant follows the vowel
q_0	Consonant follows the consonant
$q_{1,0}$	Consonant follows the consonant that follows the vowel
$q_{0,1}$	Consonant follows the vowel that follows the consonant
$q_{0,0}$	Consonant follows two consonants
$q_{1,1}$	Consonant follows two vowels

Markov has compared these probabilities with the values from the text in two cases:

1. The presence of voices in the text corresponds with independent experiments.
2. Voices create the chain

Ad 1) Let´s split 20 000 voices into 200 groups by 100 voices and determine, how many vowels are there in each group. We´re going to get the following table (Sedlačíková, 2012):

The number of voices in the group	37	38	39	40	41	42	43	44	45	46	47	48	49
The number of groups	3	1	6	18	12	31	43	29	25	17	12	2	1

The weighted arithmetic mean of the number of voices in the group is 43,19 and the probability p=0,4319. Now we count the sum of the squares of the deviations of the number of vowels in every group from the weighted arithmetic mean. The result is 1022,78. By dividing of this value by 200, we get the variance value for each group v_v = 5,1139. If we do these calculations for groups by 200, 400 and 500 hundred voices, the weighted arithmetic means will be 86,4; 172,8 and 216 and the sums of the squares of the deviations of the number of vowels will be 827,6; 975,2 and 1004, that are near to 1022,78.

Let´s move from groups of voices to the individual voices. The variance value for each voice $v_v´$ = 5,1139/100 = 0,051139, which differs from the theoretical value assuming the independent experiments $p \times (1 - p) = 0{,}4319 \times 0{,}5681 = 0{,}24536239$. Dispersion coefficient

$$c_d = \frac{0{,}051139}{0{,}24536239} \sim 0{,}208$$

This value implies the continuity of our experiments (1 = independent experiments).

Now let´s change the order of these voices. In the first step, we will rearrange the groups of voices into square matrices this way:

$$M_{i,j} = \begin{pmatrix} 1 & \cdots & 10 \\ \vdots & \ddots & \vdots \\ 91 & \cdots & 100 \end{pmatrix}; i, j \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$$

Let´s now take every five of these matrices and create new groups by hundred voices this way: the first group will be created from all the first and sixth columns, the second group will be created from all second and seventh columns, the third group will be created from all the third and eighth columns, the fourth group will be created from all the fourth and ninth columns and the fifth group will be created from the fifth and tenth columns. Now we will count all the vowels in the first and sixth column, in the second and seventh column, in the third and eighth column, in the fourth and ninth column and in the fifth and tenth column. Now we get five numbers marked (1, 6), (2, 7), (3, 8), (4, 9) and (5, 10). The number of the vowels in these groups is equal to these sums:

$$\sum(1,6), \sum(2,7), \sum(3,8), \sum(4,9), \sum(5,10)$$

Let´s create a new table of frequency of the vowels (Sedlačíková, 2012):

The number of vowels in each group	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41
The number of groups	1	0	0	0	1	2	1	3	5	1	2	9	13	12	13	11
The number of vowels in each group	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57
The number of groups	17	16	15	10	10	16	10	10	5	5	3	3	3	0	1	2

Weighted arithmetic number of the numbers of vowels in each group is 43,19. However, the sum of the squares of their deviations from the weighted arithmetic mean is 5788,8 and the variance value is

$$\frac{5788{,}8}{200} \cong 28{,}944$$

If we go from the groups of voices to the individual voices, we get the variance value equal to 0,28944, which does not differ very much from the theoretical value

$$p \times (1-p) = 0{,}432 \times 0{,}568 = 0{,}245376$$

The dispersion coefficient is then equal to

$$c_d = \frac{28944}{24537{,}6} \cong 1{,}18$$

We also may connect these voices by 2, 4 and 5 and repeat this process. If we count their sums of the squares of their deviations from their weighted arithmetical means (3551,6; 3089,2; 1004 from 86,4; 172,8; 216), where 1004 is almost 6x smaller then 5788,8, we get the implication of their "dependence".

Ad 2) Now we will count the approximate value of probabilities p_1 and p_0. Among these and 20 000 voices and 8 638 vowels, there are 1104 chains "vowel-vowel". We may also count these probabilities this way:

$$p_1 = \frac{1104}{8638} \cong 0{,}128$$

$$p_0 = \frac{7534}{11361} \cong 0{,}663$$

Difference between these results implies the dependence. Let´s mark the difference between these probabilities as

$$\delta = p_1 - p_0 = 0{,}128 - 0{,}663 = -0{,}535$$

Let´s count the theoretical dispersion coefficient as

$$c_d = \frac{1+\delta}{1-\delta} = \frac{465}{1535} \cong 0{,}3$$

Now let´s sum up all the chains “vowel-vowel-vowel“ and “consonant-consonant-consonant“ in the chain of voices:

$$p_{1,1} = \frac{115}{1104} \cong 0{,}104$$

$$q_{0,0} = \frac{505}{3827} \cong 0{,}132$$

Now let´s assume, that

$$p \cong 0{,}432$$

$$q \cong 0{,}568$$

$$p_1 \cong 0{,}128$$

$$q_1 \cong 0{,}872$$

$$p_0 \cong 0{,}663$$

$$q_0 \cong 0{,}337$$

$$p_{1,1} \cong 0{,}104$$

$$q_{0,0} \cong 0{,}132$$

Now we can determine following values:

$$\delta = p_1 - p_0 \cong -0{,}535$$

$$\varepsilon = \frac{p_{1,1} - p_1}{q_1} = -\frac{24}{872} \cong -0{,}027$$

$$\mu = \frac{q_{0,0} - q_0}{p_0} = -\frac{205}{663} \cong -0{,}309$$

We will substitute these values into the formula for coefficient of dispersion:

$$\frac{[q(1-3\varepsilon)(1-\mu) + p(1-3\mu)(1-\varepsilon) - 2(1-\varepsilon)(1-\mu)](1-\delta) + 2(1-\varepsilon\mu)}{(1-\delta)(1-\varepsilon)(1-\mu)}$$
$$= \frac{1+\delta}{1-\delta}\left[\frac{1+\varepsilon}{2(1-\varepsilon)} + \frac{1+\mu}{2(1-\mu)}\right] + \frac{(q-p)(\mu-\varepsilon)}{(1-\varepsilon)(1-\mu)}$$

The result is 0,195, which is very near to the value 0,208 calculated in 1).

3.3 George Kingsley Zipf

George Kingsley Zipf was an American linguist known for his research in psychological and physiological factors in the production and perception of speech and for his research of relations frequency of words and their rank, frequency of word and the number of different words with the same frequency, frequency of word and number of their meanings.

3.3.1 The first Zipf´s law

$$r \times f = c$$

r…rank (order of the word in the list sorted by frequency)

f…frequency of word

c…constant

Zipf has based this law on a hypothesis that in the language there are tendencies to maintain the balance between two forces – the force of unification causing the highest frequency and the lowest number of words and caused by the speaker´s economy and the force of diversification causing the highest number of words and the lowest frequency. The value of the constant depends on the length of the text.

3.3.2 The second Zipf´s law

$$a \times b^2 = c$$

a…the number of words at the same frequency

b…the frequency

c…constant value

This law applies in the middle frequency band and for all languages. According to this law, there is an indirect proportion between the frequency and the number of words with this frequency.

3.3.3. The third Zipf´s law

$$\frac{m}{\sqrt{f}} = c$$

m…the number of meanings of a word

f…the frequency of the word

c…constant value

In this law Zipf has tried to affect the semantic aspect of language with the statistical analysis. Although, as Blanka Sedačíková in her book Historie matematické lingvistiky notes, the law applies in general only for formal words (Sedlačíková, 2012).

3.4 The theory of information

It is a mathematical discipline dealing with the broadcast, coding and measuring of the information. Her founders, the English mathematician and engineer Claude Elwood Shannon and American physicist and mathematician Warren Weawer, reacted this way in the turn of

1940s and 1950s on the development of cybernetics. According to the book Historie matematické lingvistiky, this theory was hardly understandable for non-mathematicians, so Review of C. E. Shannon and W. Weawer The Mathematical Theory of Communication by Ch. F. Hockett introduced this theory to linguists (Sedlačíková, 2012).

Mainly two concepts from the theory of information have become the main focus of attention of quantitative linguistics – entropy and redundance. Entropy can be defined as the average amount of information in the result of the attempt or, for short, as the degree of uncertainty of the attempt. The entropy can be calculated with this formula, where N is cardinal number of the given set (Vopěnka, 2011; Vopěnka, 2016; Sedlačíková, 2012), p_i the probability of occurrence of the i-th element for $i \in \{1, 2, \dots N\}$:

$$H = -\sum_{i=1}^{N} p_i \log_2 p_i$$

At the same time this condition must be applied:

$$\sum_{i=1}^{N} p_i = 1;\ p_i \geq 0$$

Entropy is maximal, when all the elements are equally probable, zero, when one of the elements has probability 1 and the other elements have probability 0, and additive, when there is some final scheme A with entropy H^A, some final scheme B with entropy H^B, then entropy of the complex system AB, H^{AB}, is equal to $H^A + H^B$

Final scheme A is a set of mutually incompatible phenomena A_i with probability of occurrence $p(A_i)$ for i = 1, 2…N and can be written as: $A = \begin{pmatrix} A_1 \cdots & A_N \\ p(A_1) \cdots & p(A_N) \end{pmatrix}$

Bit (short for binary digit) is a unit for the amount of information given by the alphabet of one element and two states, so $bit = \log_2 2^1$

Let´s now extend general formulas. Blanka Sedlačíková has created 4 imitations of texts written in several languages in her book Historie matematické lingvistiky based on 4 entropies – H_0 is entropy based the number of letters in any given alphabet, H_1 is entropy based on a probability of occurrence of any given letter, H_2 is entropy based on a relative frequency of pairs of letters, H_3 is entropy based on a relative frequency of trinities of letters (Sedlačíková, 2012).

Let´s start with H_0. The Czech alphabet has 42 letters (we will count space between words as a letter, too). The formal scheme for this entropy looks like this:

$$0 = \begin{pmatrix} - & a & \text{á} & \cdots \\ \frac{1}{42} & \frac{1}{42} & \frac{1}{42} & \cdots \end{pmatrix}$$

This way, texts like ďj mzgučxýďyaýweaožá can be created. The entropy of this imitation is 5,39 (Sedlačíková, 2012).

Let´s go on with H1. First, we have to find out the probabilities of occurrence of given letters. The final scheme would look like this:

$$1 = \begin{pmatrix} - & a & \cdots \\ 0{,}163 & 0{,}054 & \cdots \end{pmatrix}$$

The final text could look for instance like this: žia ep atndi zéuořmp. The entropy of this text is 4,67 (Sedlačíková, 2012).

Now let´s go on with H2. We have to find out the relative frequencies of pairs of letters (and conditional probabilities of occurrence in dependence on the immediately preceding letter) This situation would describe 42 final schemes like these:

$$- = \begin{pmatrix} - & a & \cdots \\ p(-|-) & p(a|-) & \cdots \end{pmatrix}$$

$$A = \begin{pmatrix} - & a & \cdots \\ p(-|a) & p(a|a) & \cdots \end{pmatrix}$$

$p(a|-)$ is the probability of letter a following the space. We can count the entropy of every scheme with this formula:

$$H(B|A_i) = \sum_j p(B_j|A_i) \log_2 p(B_j|A_i)$$

For general entropy we must know their mean value:

$$E\{H(B|A_i)\} = \sum_i p(A_i) H(B|A_i)$$

After substituting the formula for conditional entropy, we get on the right side

$$-\sum_i \sum_j p(A_i) p(B_j|A_i) \log_2 p(B_j|A_i)$$

and after substituting the formula $p(A)p(B|A) = p(A \cap B)$can we derive the formula for H_2

$$-\sum_i \sum_j p(A_i \cap B_j) \log_2 p(B_j|A_i)$$

The final text with entropy H_3 would be like dnes a vase miléklár and the general formula for calculating the entropy would be

$$H_n = -\sum_i \sum_j p(A_i(n-1) \cap B_j) \log_2 p(B_j|A_i(n-1))$$

$A_i(n-1)$ is the probability of (n-1) foregoing letters.

Let´s have a look on entropy in another way. Every message can be encoded with alphabet with n characters to a sequence with average number of characters of encoded message per character of original message equal to

$$\frac{H}{\log_2 n}$$

If we know the entropy, we can also estimate the number of all sequences with n characters. The number is 2^{nH}. This entropy works on the level of letters, so there is also a relative entropy (n is the order of entropy):

$$h = \frac{H_n}{H_0}$$

We can also determine the entropy on the level of words, morphemes and phonemes. If we know the entropy of letters, the number of characters and of the word forms in the text, we can build the equation:

$$H_{\infty}^{w} \times number\ of\ word\ forms\ in\ the\ text = H_{\infty}^{let} \times number\ of\ letters\ in\ the\ text$$

The entropy of words can be determined from these equations by dividing both sides by the number of word forms. On the right side of this equation, we get a fraction which is nothing more than the average length of word w (in letters), which has to be increased by 1 (the space between words is also a character). The formula for entropy on the level of words is then following:

$$H_{\infty}^{w} = H_{\infty}^{let} \times (w + 1)$$

And here we face the limitations of using the mathematical theory of information in linguistics. The easier and faster imitation of a language with the help of morphemes or words is not possible, because the natural language is not language with the finite number of states (Sedlačíková, 2012).

Another important term from the theory of information, which is used in linguistics, is redundancy. It can be used instead of the relative entropy and the formula for it is

$$R_n = 1 - \frac{H_n}{H_0}$$

The main importance of redundancy is the assurance of reliability and it can also be used to thy comparative study of languages, as shows the following table (Sedlačíková, 2012):

Language	R_0	R_1	R_2	R_3	R_∞
Czech	0	0,13	0,28		
Russian	0	0,13	0,30	0,40	0,72-0,82
English	0	0,16	0,30	0,35	0,71
German	0	0,14			0,66

4 Computer technologies for translators and linguists from the creators´ point of view

In this part, the inner workings of several software tools for translators and linguists will be described.

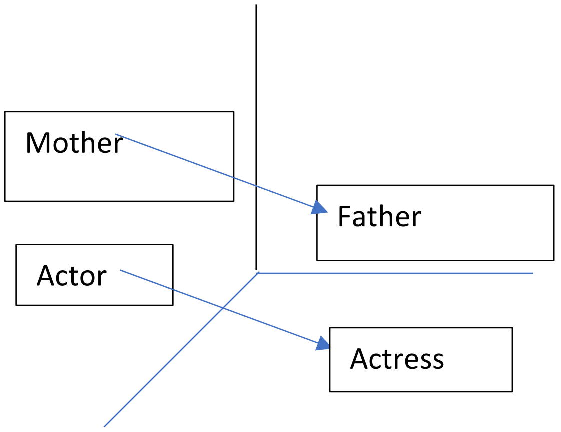
4.1 Machine translation, deep learning, word2vec

Machine translation is a method of the conversion of the source language to the target language without necessary human control. For a result with great accuracy, we need a well-trained system. Deep learning is a new technique enabling the system to learn like a human. Deep neural network (DNN) is a neural network with more than one hidden layer. Deep neural networks can be used to translate without using large database of rules. Translation process starts with word alignment.

In word alignment the input is parallel sentence pair and the output is a pair of the most related words. Let´s have a source sentence S=s_1, s_2, s_3…s_m and the target sentence T = t_1, t_2,

t_3…t_n, then the set $A = \{(a, b), 1 \leq a \leq m, 1 \leq b \leq n\}$, (a, b) denotes ($s_a$, t_b). Because of the maintenance of the history and accurate prediction of text alignment, the recurrent neural networks are a better choice than feed forward neural networks (Singh, Kwnar, Darbari, Singh, Rastogi, Shikha; 2017).

Word embedding is a representation of a continuous space vector, generated by the method word2vec, and a key method to find the vector value of words



Word representation in continuous space

V represents the corresponding value of the words. According to the picture 1, the corresponding value of the word *actress* can be calculated from the corresponding values of all the other words in the picture from the equation $V_{[Actress]} = V_{[Mother]} + V_{[Father]} - V_{[Actor]}$

The next step is rule selection/extraction. In this step, the rules are selected on a basis of a word alignment and the reordering model is trained by word aligned bilingual text. (Singh, Kwnar, Darbari, Singh, Rastogi, Shikha; 2017)

Reordering and predict the sentence structure is the next step. The non-linear combination of recursive neural network and recurrent neural network (R2NN) is the best choice for this step (Singh, Kwnar, Darbari, Singh, Rastogi, Shikha; 2017)

The next step is a language modelling. The neural network predicts a sequence of outputs by sequence of inputs. Two RNNs are required for this task – one for encoding, the other for decoding. (Singh, Kwnar, Darbari, Singh, Rastogi, Shikha; 2017)

The last step is a joint translation. This model is used to predict the target word with the help of history of source and target words.

Because of the difficulty of training the RNN for Word Alignment, an alternative in the form of bilingual corpus can be used (Singh, Kwnar, Darbari, Singh, Rastogi, Shikha; 2017).

4.2 Machine translation using only monolingual corpora

In this method the machine translation model starts with an unsupervised naïve translation model. (Lample, Conneau, Denoyer, Ranzato, 2018). Let´s have a source sentence with m words. The encoder computes sequence of m hidden states (vectors in $\mathbb{R}^n$, where n is a dimension of latent space). A decoder takes an input I and generates the output sequence a = (a_1, a_2, a_3…a_z), where every word corresponds to the vocabulary of selected language. The decoder chooses the word a_i with the help of word a_{i-1} which has the highest probability of being the next one. Both encoder and decoder are trained by minimizing an objective function. This objective function measures the ability to process the noisy version of the training sentence. (Lample, Conneau, Denoyer, Ranzato, 2018). The noisy version is generated by dropping and swapping words or by translation in the previous iteration. (Lample, Conneau, Denoyer, Ranzato, 2018) The training of this model starts from word-by-word translation model, which improves itself iteratively. (Lample, Conneau, Denoyer, Ranzato, 2018)

4.3 Charles University Block-Backtranslation-Improved Transformer Translation (CUBBITT)

The six-layer encoder encodes the numerical representation of the original sentence into a deep representation and decoded by a six-layer decoder to the target sentence. Every layer of both decoder and encoder consists of self-attention and feed-forward layers. There is also encoder-decoder attention layer in the decoder with an input which was created in the last layer of encoder. (Popel, Tomková, Tomek, Kaiser, Uszkoreit, Bojar, Žabokrtský, 2020)

4.4 Google´s Multilingual Neural Machine Translation System

It is simple method of translating between multiple languages using a single model. There´s no need to change the neural machine translation model. The artificial token indicating the required target language has been added. Benefits of this method are: simplicity (scaling to more languages is trivial, the total number of necessary models cut down), low-resource language improvements (all parameters shared by all the modelled language pairs), zero-shot translation (it can learn to translate in the pairs it has never seen before). The structure of this system is the same as the structure of Google´s Neural Machine Translation. The only change is one modification to the input data – an artificial token indicating the target language. The token can be shown on example translation from English to Spanish from (Johnson, Schuster, Le, Krikun, Wu, Chen, Thorat, Viégas, Wattenberg, Corrado, Hughes, Dean, 2017): How are you? -> ¿Cómo estás? (GNMT) vs. <2es> How are you? -> ¿Cómo estás? (GMNMTS)

5 Computer technologies and software from the translator´s point of view

In this part the possibilities of the corpuses and software tools for the linguists and translators will be described. According to the book *Technologie ve službách překladatele* by Pošta (2017), these options of corpuses are the most valued by linguists and translators: the comparison of the frequency of two competitors, looking for collocations and finding out or verification of grammatical phenomena, usage, semantic nuances, connotations or associations in monolingual corpuses and looking for equivalents in the second language in parallel corpuses.

5.1 Kinds of corpuses

Corpuses are divided by several criteria. The first criterion is size. According Pošta (2017), the fifth version of the Czech national corpus from the year 2010 contains 3,84 billion words. The second criterion is the topic or focus There are representative corpuses created from different types of text or specialized corpuses. The third criterion is the degree of linguistic processing. There are "raw" corpuses built from the texts in their original form and corpuses with additional information about the texts or important linguistical information. Corpuses can also be divided into monolingual and parallel. Monolingual corpuses contain texts in one language, parallel corpuses are built from two or more monolingual corpuses and contain the original text and also its translations connected with original texts. Corpuses can further be divided into comparable and polylingual corpuses.

In the comparable corpuses there are two folders of texts of texts in the same language. In one folder there are original texts only, in the second folder there are only translations. According to Pošta (2017), the texts must be similar and at the same ratio.

Polylingual corpuses are created from two or more monolingual corpuses by the same criteria as the comparable corpuses, but this time each monolingual corpus is in another language.

For evaluation of the corpus, we use terms such as frequency and frequency distribution. Frequency is the number of occurrences of any given word in the given corpus. Frequency distribution is a function for creation of the overview of different words in the results of searching. Frequency and frequency distribution can also be used to compare the occurrences of words across different text types or different sociolinguistic variables (education or sex of the speakers, dialect area…), to find out frequency distributions of different attributes and to create the frequency dictionaries. Besides this, we can also do some other statistical functions and analyses, such as counting the absolute and relative frequency.

The absolute frequency is the real number of occurrences of any given word in in corpus. We tag it **f(x)**, where **f** stands for frequency and **x** for the examined word. It is basic statistical data and every other statistical data is derived from it.

The relative frequency is frequency derived from the absolute frequency and is related to its size or to its particular part. For instance, we can count the relative frequency of any given verb **v** against the whole corpus **c** or just all the verbs of the corpus. The formula for the relative frequency is following

$$f_{rel} = \frac{f_{abs}}{N} \times f_{norm}$$

N stands for the number of all the words in the corpus, f_{abs} is the absolute frequency of the word **x** in the corpus and f_{norm} is normalization number. Normalization number normalizes the quantitative data, which enables users to compare the frequencies across different corpuses regardless of their size.

The term collocation has been introduced to linguistics in 1930s by J. R. Firth and further developed by his pupils J. Sinclair and M. A. K. Halliday. It indicates multiword expressions, such as technical terms, stable phrases or purely random word combinations (so called n-grams).

Colligation differs from collocation in its focus – collocation is lexical-semantic linkage, colligation is grammatical or lexical-grammatical linkage.

Collocations can be sorted into these 3 groups:

1. Free combinations
    a. The expressions are substitutable typical is also wide functional field and compositional semantics
2. Tight combinations
    a. Restricted functional field and number of options
3. Proper collocations
    a. For instance, Canary Islands, carbon dioxide

However, this sorting does not cover the whole problematics of collocations, because there are semantically anomalous collocation connections, there are borders between collocates and phraseologisms. Another factor is the behaviour of neutral colocation units with wide functional field and monocolocable units.

5.2 Tests of the collocational significance

Concordance tools use numerous statistical tests for searching and evaluation of the collocations and colligations, the most used in linguistics are MI3-score, Dice, logDice, logLikelyhood and Chi-squared test. Let´s introduce the most common tests, MI-score and t-score.

MI-score test (MI = mutual information) is based on the mathematical theory of information and operates with the probability of collocates **x** and **y**. The general formula is defined as

$$I(x,y) = \log_2 \frac{P(x,y)}{P(x) \times P(y)}$$

I(x,y)…the mutual information of phenomena **x** and **y**

P(x)…probability of occurrence of the word **x**

P(y)…probability of occurrence of the word **y**

P(x,y)…probability of occurrence of the word **y** in the context of word **x**

However, the probabilities of words in language are unknown, so this formula has to be modified. The probabilities will be approximated by the relative frequency of the word to the size of corpus (N):

$$P(x) = \frac{f(x)}{N}; P(y) = \frac{f(y)}{N}; P(x,y) = \frac{f(x,y)}{N}$$

After the substitution and edition can we get this formula for MI-score:

$$mi(x,y) = \log_2 \frac{N \times f(x,y)}{f(x) \times f(y)}$$

Mi-score measures the strength of association between 2 words, because – mathematically – it is the logarithm of the ratio of probabilities of their occurrence together and independently. This test is viable for discovering of thinner collocations and the core of phrasemes.

T-score is based on the statistical method of testing the hypotheses using the t-test. The formula is defined as

$$T = \frac{(f(x,y) - \frac{f(x) \times f(y)}{N})}{\sqrt{f(x,y)}}$$

where f is the frequency of given phenomenon, x and y are given phenomena and N is the size of the corpus. The t-test is based on the difference between observed and expected result. T-score determines, if the numbers of occurrences of individual words and their pairs respond to the random distribution of words in the corpus and generally the measure of collocability of the elements **x** and **y**. It is viable for searching the colligations. It does not measure the force of searched associations, but it does measure the probability of existence of this collocational or colligational association. MI-score overestimates collocations with low frequency and t-score overestimates collocations with high frequency. For these differences MI-score and t-test are used combined, but caution is required when interpreting the results (Petr Pořízka). These statistical thresholds have been set to for non-random connections: t-score > 2 and MI-score > 3. Let´s connect these scores with the classification of collocations:

1. Free combinations – low MI-score and t-score
2. Bound combinations – higher MI-score and t-score
3. Proper collocations – high MI-score

5.3 The Czech national corpus

The Czech national corpus consists of tens of different corpuses, mostly Czech corpuses, but also foreign language corpuses, including polylingual parallel corpus InterCorp, whose version 13 includes the Czech language and 39 other foreign languages. The most important information for translators is concordance. This information is to be found in the interface KonText. In this interface, there are to be found 3 numbers: a simple number of occurrences of a word in a given corpus, i. p. m. (instances per million) and ARF (average reduced frequency), which adapts the frequency of the word to its dispersion.

5.4 Tools for computer assisted translation

Computer assisted translation is a set of software tools, which includes translation memory. The translation memory is a database connecting the original sentences with their translations. Pošta (2017) opines, that the era of computer assisted translation has started in 1960s after the machine translation hadn´t fulfilled the expectations. The computer-assisted translation has these functions and options:

- Document analysis – analysis of percentage of matches and the extent of the document
- Segmentation of the text – the programme divides the text into segments to work with them further

- Seeking matches – the programme goes through the translation memory and suggests similar and identical segments
- Pre-translation – the programme inserts the matches to the translation field
- Searching concordances, full-text search of words and word formations in the translation memory
- Self-propagation – the translation may be inserted into all the iterations of the translated segment
- Connection with electronic dictionary
- Connection with terminological databases
- Whisperer
- Fast insertion of the placeables
- Quality control
- Aligner – this module aligns the original text from one file and the translation from another file and creates the translation memory
- Module for machine translation
- Retrofit – carrying the changes from the target file back to the computer-assisted translation software
- Locking of chosen segments
- Filtering
- Creating the resulting document in the same format and with the same formatting as of the original document.

5.5 Machine translation

The successful model of machine translation is called phrasal statistical translation. Another model of machine translation, neuronal translation, seems to have very positive results, but also more unexpected mistakes or more omitted words (Pošta).

Some translators use online services, such as Google Translator or Microsoft Translator to get half-finished translation and perform postediting. The usage of these services may be forbidden by the contract or the law protecting sensitive data.

Another kind of translators are cloud translators on the subscription basis. These can be divided into services providing general or specialized translator and services enabling its users to upload translation memories to process them and provide the users personalized solutions. The usage of these services may also be forbidden by the contract or laws protecting sensitive data.

The third kind of translators are desktop translators, which the user installs into the computer. Examples of these translators are Moses and Slate Desktop.

The quality of machine translation depends directly on the size parallel corpus or translation memory, except cases, where smaller translation memories corresponding with the topic and type of the text and the idiolect of the translator.

During the postediting human translator goes through the text segment-by-segment, compares the original text and the translation and makes changes in the translation. The kinds of mistakes depend on the used machine translation model and usage of modules for control and correction. Increased occurrence of words in the wrong form, inappropriate word order, incorrectly selected meanings and synonyms, insufficient idiomaticity, missing, indwelling or nonsensical words, ambiguous translations, negations, non-preservation of capital and small letters and words from the original language are to be expected.

The goal of preediting is to minimize the number of mistakes. The so-called controlled language can be used for it. In this controlled language, there is permitted vocabulary, simplified grammar and strict rules for editing of the document.

6 Creating of the translation memory and corpus, data formats and coding of the characters

The process of creating the translation memory and corpus and formats od the data will be described from the user´s point of view step-by-step.

6.1 Data formats and coding of the characters

All the language data have to be converted into the plain text with the suffix .txt with the correct character coding. During the conversion, all the formatting of the original document will be removed.

There are three types of coding: ASCII, ANSI and Unicode. ASCII (American Standard Code for Information Interchange) contains 128 numerically expressed positions for each letter (for instance A = 65, a = 97, M = 77, m = 109, T = 84, t = 116), regardless of font or typeface. ANSI and Unicode were built on ASCII. ANSI (American National Standards Institute) consists of 256 positions - 128 positions from ASCII and positions 129-256 for the national alphabets. International Organisation for Standardisation allocates unique code for each character set (ISO 8859-1 for Western European languages or ISO 8859-2 for Central European languages). There are other parallel character sets for these language groups, such as Windows-1250 for Central European languages from the company Microsoft, which differs from ISO 8859-2 in coding of the characters ž, š, ť, Ž, Š, Ť. It has been resolved with the introduction of Unicode, allocates every character from every known language its unique code, which means 65 536 – 2 billion of positions for coding – it depends on the type of Unicode. The basic types of Unicode are UCS (Universal Character Set) and UTF (UCS Transformation Format). These types have their coding UCS-2, UCS-4 UTF-8 and UTF-16. The most common coding is UTF-8. Unicode reduces the disadvantage of ANSI, where the same character has in different ANSI sets different code.

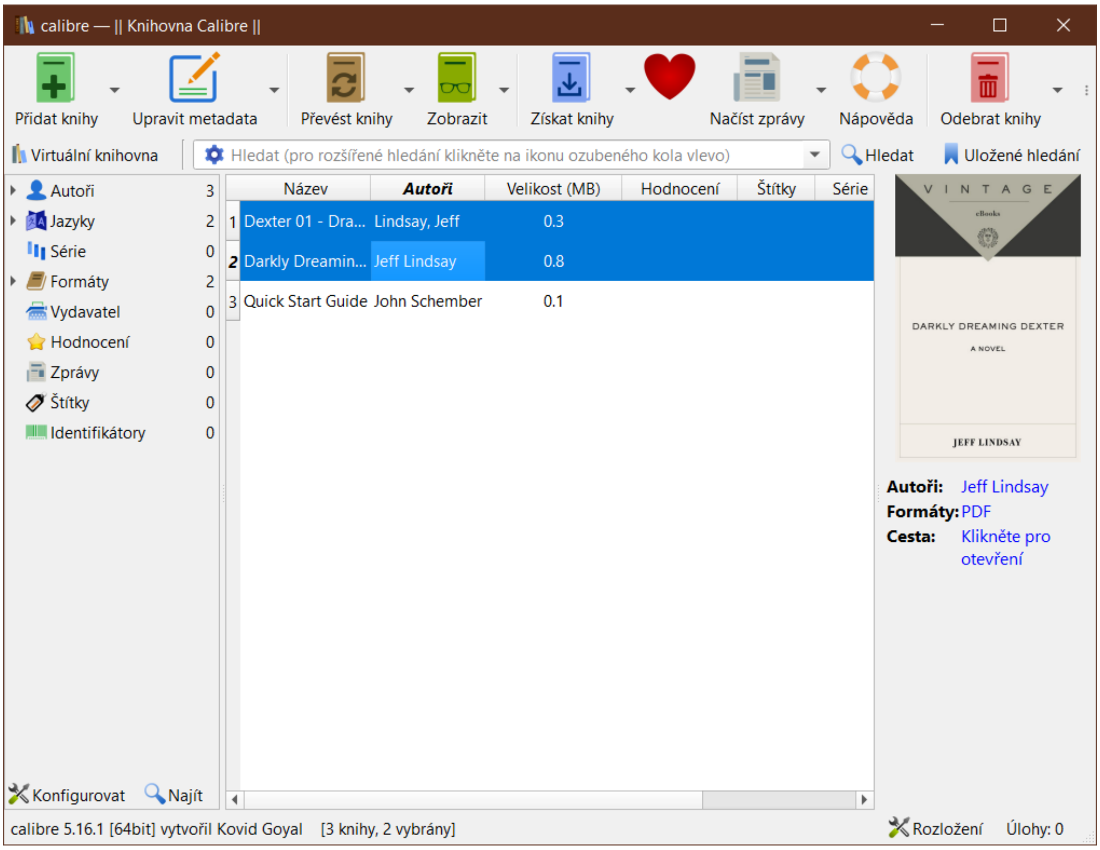
It means, that for instance for Czech language can we choose between 3 types of coding:

1. Universal Unicode (UTF-8)
2. ISO Central Europe (ISO-8859-2 or ISO Latin 2)
3. Windows-1250

6.2 Optical character recognition (OCR) and transformation of e-books into the plain text

When the translator needs to scan the document or the book for translation and then convert it to the text or transform it from the picture, the programme for the optical character recognition is needed. Full versions of these programmes (for instance ABBYY FineReader) include the spell check and learning mode – the programme shows the translator the recognised characters and the translator will check the correctness of the recognition or possibly corrects the character. The text does not need to be converted correctly. However, the sentence cannot be split in two by improperly placed end of the paragraph, the page number or the header of the document. That will be checked with the help of the function Find and replace.

In the case of the transformation of e-book into the plain text, the translator is not allowed (in accordance with §43 of Act 121/2000 Coll.) to transform e-book protected by the DRM technology. E-book without this protection (probably in the format EPUB) may be transformed in the programme Calibre, but the translator must work with the e-book in accordance with the Copyright Act. This process will be further shown on book Darkly dreaming Dexter and its Czech translation Drasticky děsivý Dexter by Jeff Lindsay.



Autoři	3
Jazyky	2
Série	0
Formáty	2
Vydavatel	0
Hodnocení	0
Zprávy	0
Štítky	0
Identifikátory	0

	Název	Autoři	Velikost (MB)	Hodnocení	Štítky	Série
1	Dexter 01 - Dra...	Lindsay, Jeff	0.3			
2	Darkly Dreamin...	Jeff Lindsay	0.8			
3	Quick Start Guide	John Schember	0.1			

Autoři: Jeff Lindsay
Formáty: PDF
Cesta: Klikněte pro otevření

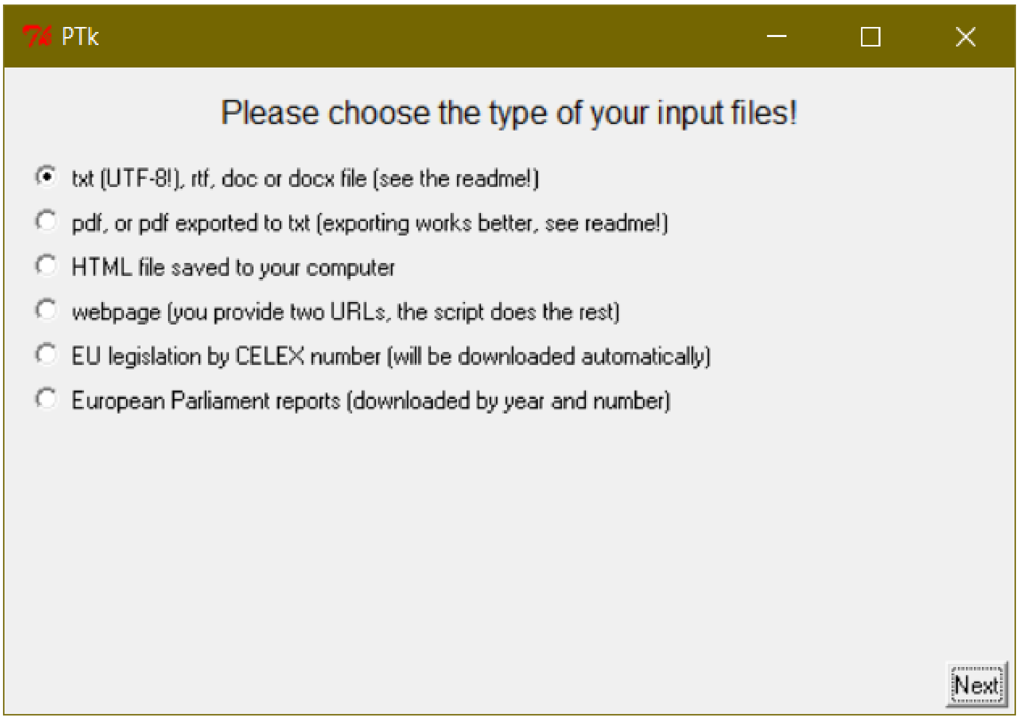
calibre 5.16.1 [64bit] vytvořil Kovid Goyal [3 knihy, 2 vybrány]

Figure 1Calibre with the books chosen for transformation

To keep the formatting of the books, we will have to save the documents in TXT in coding UTF-8.

In the next step, we will be working with Microsoft Word and its function Find and replace. With this function we will remove redundant tabs, optional divisions or paragraphs ending with a hyphen or with a small letter.

In the next step we will align the sentences. This step is essential for creating the parallel corpus, translation memory and even the bilingual book. With the help of special programme, we will align the translation sentences to the original sentences. We can also align 2 translation sentences to 1 original sentence or 1 translation sentence to 2 original sentences. There are 3 main methods for sentence alignment: alignment according to the length of the sentence, alignment according to the dictionary and alignment according to the word similarity. The most used approach is alignment according to the length of the sentence, which is also known as Gale´s and Church´s algorithm, because the length of the original sentence is usually equal to the length of the translation sentence (Pošta). There are several software aligners, we will be using LF Aligner written by translator András Farkas. After downloading the ZIP archive with programme from https://sourceforge.net/projects/aligner/ and launching aligner\LF_aligner_4.21.exe, we will choose the type of our input files. In our case it will be TXT files.



Please choose the type of your input files!

● txt (UTF-8!), rtf, doc or docx file (see the readme!)

○ pdf, or pdf exported to txt (exporting works better, see readme!)

○ HTML file saved to your computer

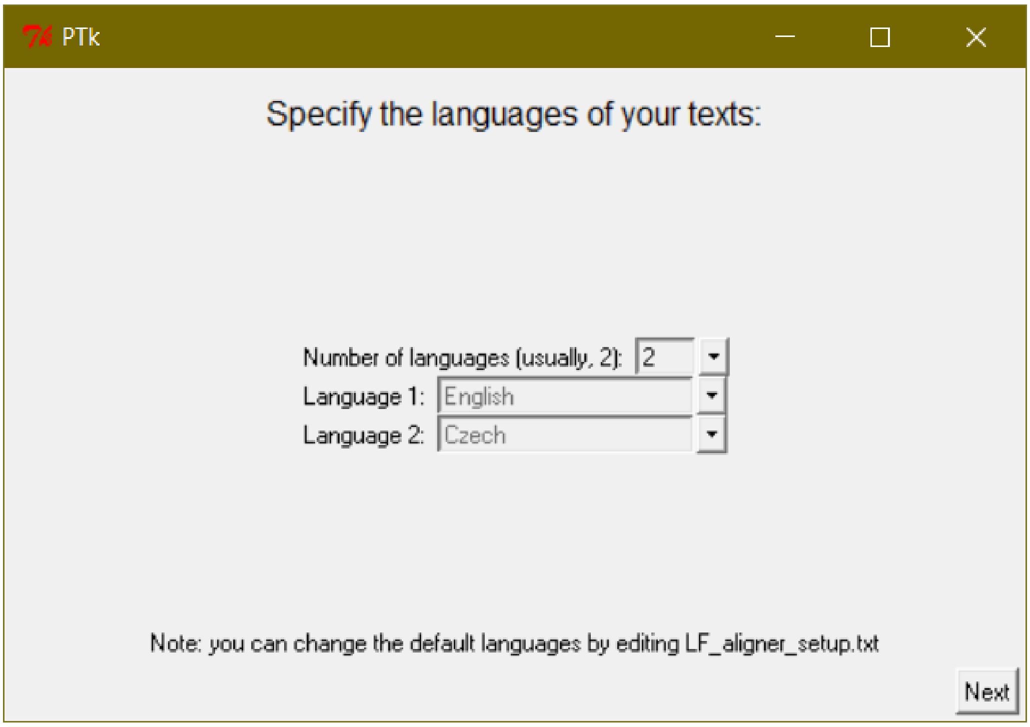
○ webpage (you provide two URLs, the script does the rest)

○ EU legislation by CELEX number (will be downloaded automatically)

○ European Parliament reports (downloaded by year and number)

Figure 2Choosig the type of inout files

In the next step we will choose the default languages of our texts.



PTk

Specify the languages of your texts:

Number of languages (usually, 2): 2

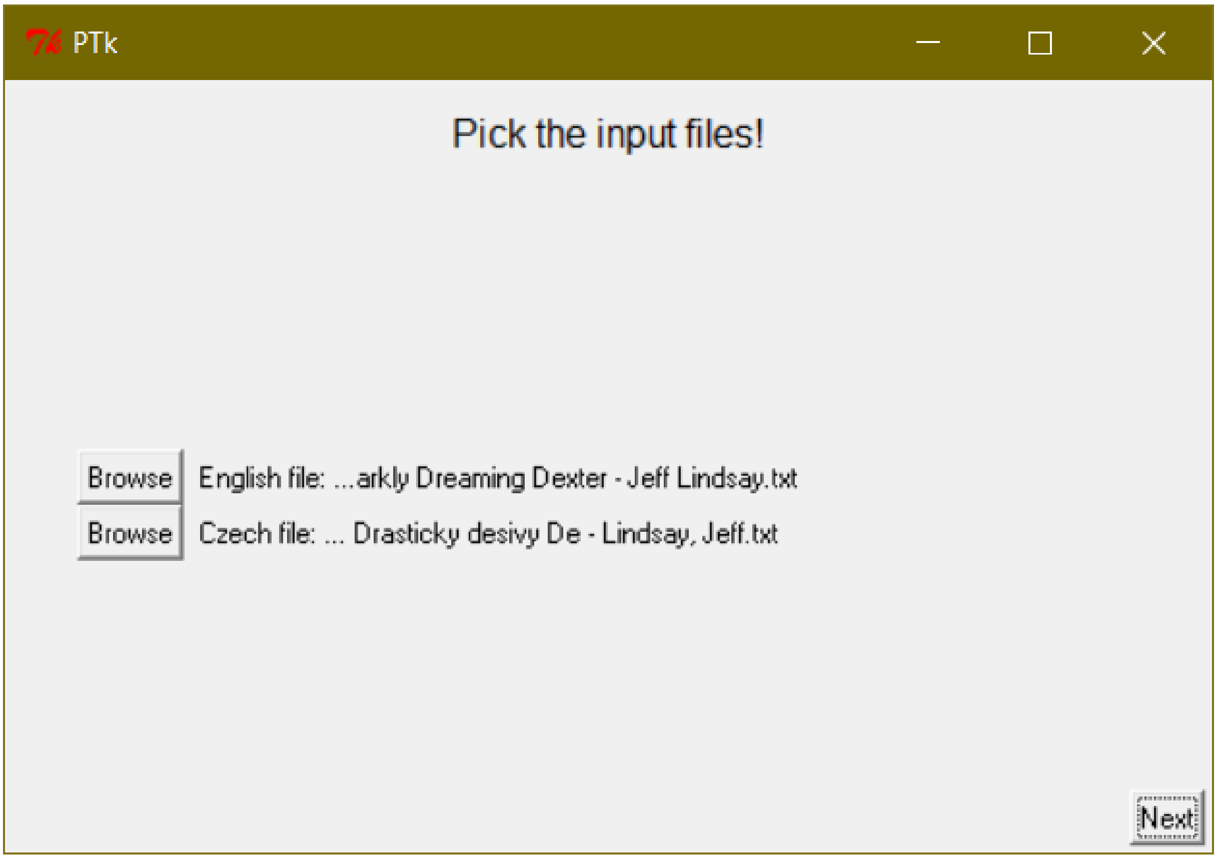
Language 1: English

Language 2: Czech

Note: you can change the default languages by editing LF_aligner_setup.txt

Figure 3The setting of default languages

In the next step we will choose the input files – in our case Darkly dreaming Dexter and Drasticky děsivý Dexter.



PTk

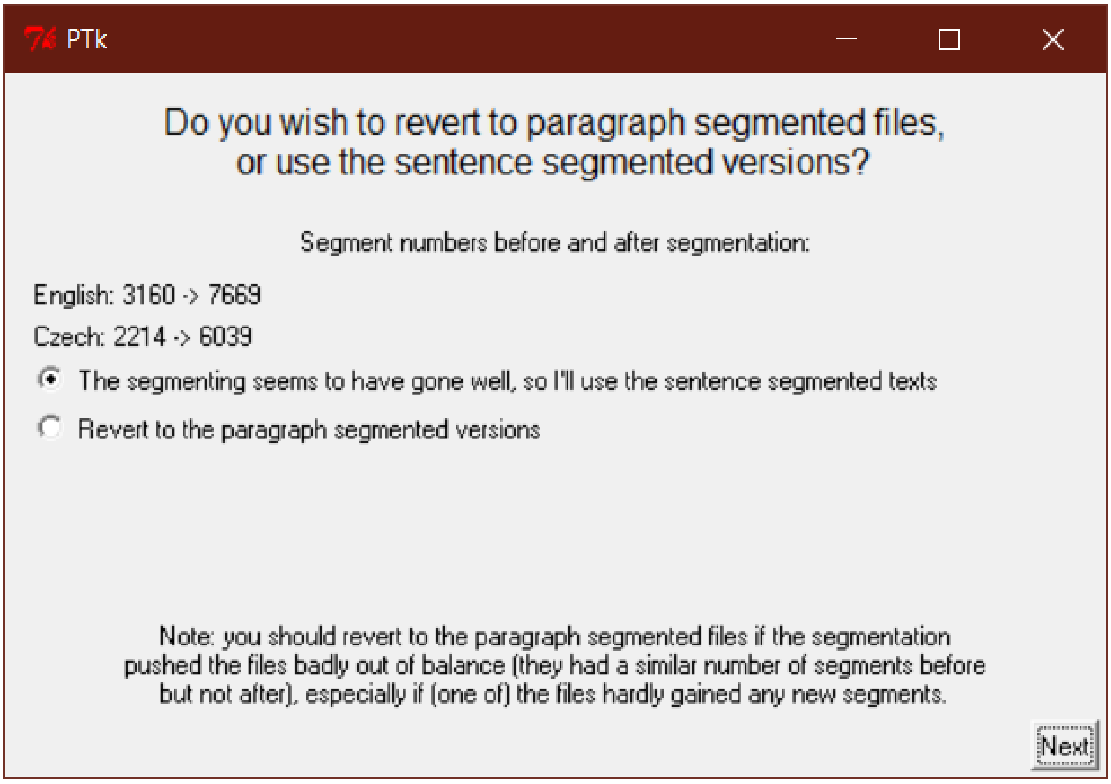
Pick the input files!

Browse English file: ...arkly Dreaming Dexter - Jeff Lindsay.txt

Browse Czech file: ... Drasticky desivy De - Lindsay, Jeff.txt

Figure 4Choosing the texts to align

In the next step we have to choose between sentence and paragraph segmentation.



Do you wish to revert to paragraph segmented files, or use the sentence segmented versions?

Segment numbers before and after segmentation:

English: 3160 -> 7669

Czech: 2214 -> 6039

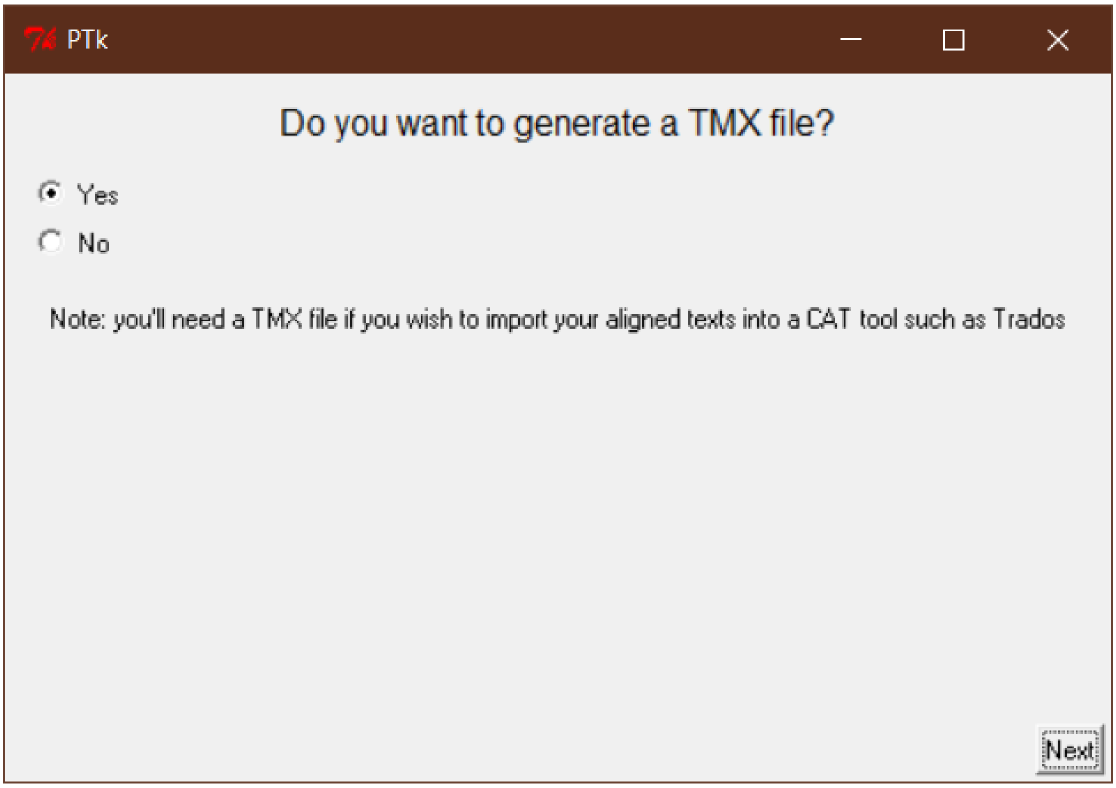
● The segmenting seems to have gone well, so I'll use the sentence segmented texts

○ Revert to the paragraph segmented versions

Note: you should revert to the paragraph segmented files if the segmentation pushed the files badly out of balance (they had a similar number of segments before but not after), especially if (one of) the files hardly gained any new segments.

Figure 5Choosing between two ways of segmentation

In the next step we choose if we want to generate a TMX file.



Do you want to generate a TMX file?

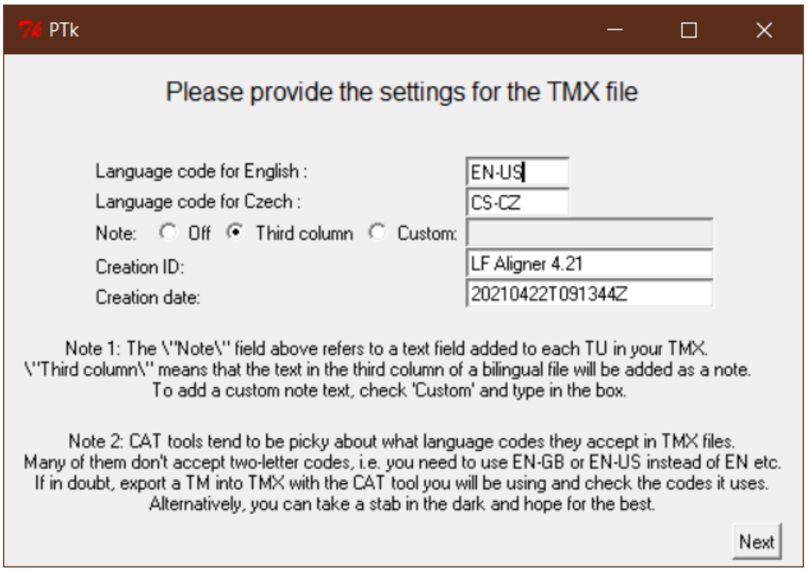
● Yes

○ No

Note: you'll need a TMX file if you wish to import your aligned texts into a CAT tool such as Trados

Figure 6We will want to use the file in CAT

In the last step we will generate the TMX file.



Please provide the settings for the TMX file

Language code for English : EN-US
Language code for Czech : CS-CZ
Note: ○ Off ● Third column ○ Custom:
Creation ID: LF Aligner 4.21
Creation date: 20210422T091344Z

Note 1: The \"Note\" field above refers to a text field added to each TU in your TMX.
\"Third column\" means that the text in the third column of a bilingual file will be added as a note.
To add a custom note text, check 'Custom' and type in the box.

Note 2: CAT tools tend to be picky about what language codes they accept in TMX files.
Many of them don't accept two-letter codes, i.e. you need to use EN-GB or EN-US instead of EN etc.
If in doubt, export a TM into TMX with the CAT tool you will be using and check the codes it uses.
Alternatively, you can take a stab in the dark and hope for the best.

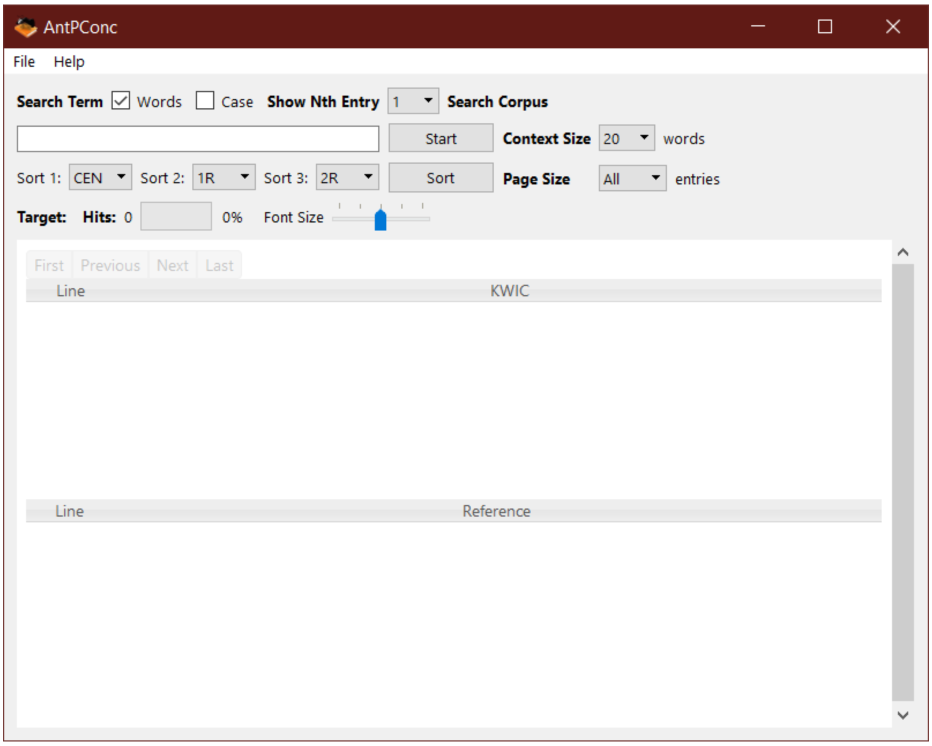
Figure 7It is important to use correct language codes

Translators who want to get aligned text files from translation memory have 2 possibilities:

1. Translation memory has less then 50 thousand segments – the translator uses SDLTmConverter
2. Translation memory has more than 50 thousand segments – the translator uses the full version of SDLTmConverter, Heartsome TMX Editor or Olifant. These programmes can also be used to view and edit the translation memory.

6.3 Creating the language corpus

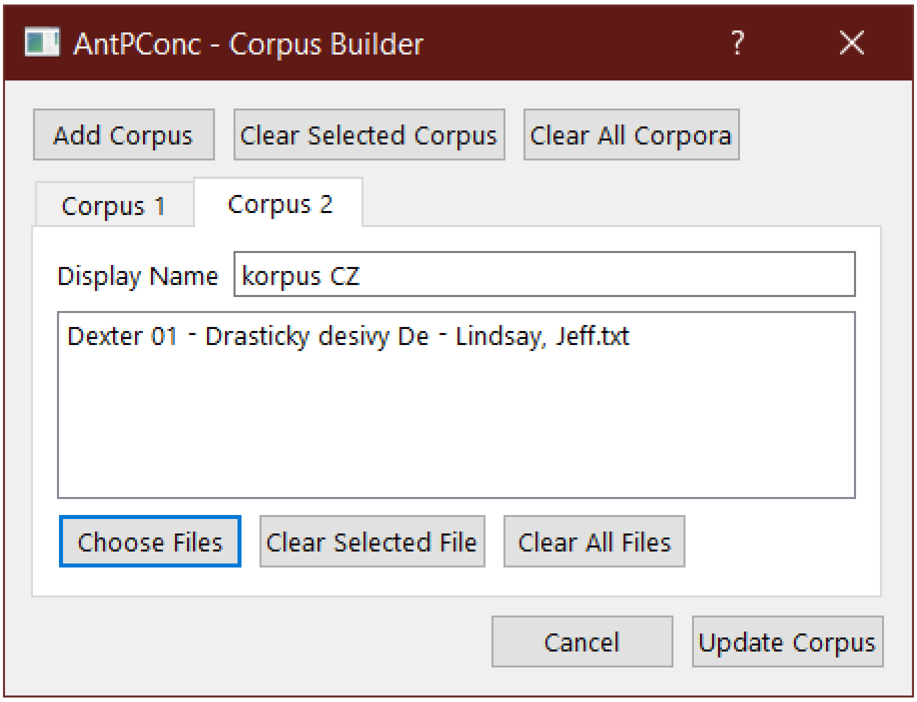
To create the corpus, the corpus manager is needed. We will use the programme AntPConc.



Search Term ☑ Words □ Case Show Nth Entry 1 Search Corpus
Context Size 20 words
Sort 1: CEN Sort 2: 1R Sort 3: 2R
Page Size All entries
Target: Hits: 0 0% Font Size

Figure 8User Interface of AntPConc

In this software tool we will generate the parallel corpus from 2 texts – one original (Darkly dreaming Dexter) and one translation (Drasticky děsivý Dexter). In this tool, we create two corpuses – one with the original text, one with the translation text.


AntPConc - Corpus Builder

Display Name: korpus CZ

Dexter 01 - Drasticky desivy De - Lindsay, Jeff.txt

Figure 9 Building two corpuses - korpus ENG and korpus CZ

6.4 UNIX and Linux tools for corpus processing

Operating system Linux contains native application for effective operations with texts, such as coreutils, sed, (g)awk or (e)grep.These tools can be used for whole series of corpus operations, such as creating the frequency dictionaries. For these tasks, the tools (g)awk, sed and (e)grep can be used for it. For Windows users there are software emulators running on both 32-bit and 64-bit version of Windows, such as Cygwin or MinGW.

Awk is computer language for processing the text files, which uses association fields (fields indexed by chain keys). His newer and open-source version is called gawk. Application sed (short for stream editor) was written for text transformations and it is a non-interactive application controlled from terminal, which can process even large text files. Grep (name is derived from the command **g/re/p**, which orders the programme: “search globally for lines matching the regular expression re, and print them “) is terminal-controlled programme which loads the data from text files and with using regular expressions writes down all the lines corresponding with the given search mask. We can find three different variants with different extension functions egrep, fgrep and rgrep. These names are equivalent to the following enrolments:

egrep to **grep -E**

fgrep to **grep -F**

rgrep to **grep -r**

Editor sed can be used for numerous tasks relating to the automatic text processing, for instance for conversion from one character set to another character set.

In the command line, we can create dictionary by using command **tr** and POSIX metaexpressions [:upper:] and [:lower:] and convert spaces to the characters of new line. For sorting words in the alphabetical order, we will use function **sort**. To count and print the token, we will use the command **uniq-c**.

7 Conclusion and reflection on the future of translation and translators in the light of technology

This bachelor´s thesis has shown us, how important is it for translators and translatologists to understand modern computer technologies and the principles behind them. Now can we weigh the pros and cons of computer linguistics.

According to the Moore´s law, the computing power per unit prize doubles every 18 months. It means that thanks to the exponential growth of the computing power, the more and more robust databases and better and better algorithms, the computer assisted translation tools will suggest better translations, which can motivate the translators to create these databases and that can motivate the producers of dictionaries to create and produce more of them. However, awareness of the value of language date must come hand in hand with technical development. Intelligent data storage and protection is in their best interest. If their contract with customers does not allow them to keep the translation memories for their future projects, the translator loses a part of his reward.

Those translators working for translation agencies will have to face their higher substitutability. They will face the question, if the pros of working for the agency outweighs the cons connected with their higher substitutability. We may be looking forward on more teams and associations of translators where their members will "pool" their translation memories and corpuses.

Other aspect is the need of vocational education in technologies and business aspects for translators. It has not been earlier needed, but nowadays it is necessity and it is a task for schools and lectors to provide the opportunities for lifelong education for language professionals. (Pošta, 2017) The higher usage of machine translation, CAT and corpuses also means higher computer performance requirements and higher investments into the hardware and software.

The development of computer technologies brings also new questions for linguists. They will have to find out, if these technologies affect the development of language itself – and if yes, how do they affect it. Will people prefer the more frequent words and forget the less frequent ones? Will the vocabulary become poorer and poorer? And if so, how will we face that?

The offer of software translators will probably be richer and richer, which means that the number of translated texts will probably be higher. This transforms the translator into posteditor with more difficult job. That brings the higher risk of burnout or other mental health problems (Pošta).

The offer of tools for computer assisted translation will probably also be richer. The translator will have to know when use which tool, because not every offered tool is usable in every translation phase and in every translation.

The job of translator will still have its place in the future, because even the best machine translators, corpuses or computer assisted translation tools cannot replace the human feeling for language, judgment and creative approach needed for adaptation of the translations into cultural context, realities or marketing. It may actually bring more interesting opportunities for translators, because these translations and these genres require non-routine approach.

8 Bibliography

Sedlačíková, Blanka: *Historie matematické lingvistiky*; Nadace Universitas v Brně, AKADEMICKÉ NAKLADATELSTVÍ CERM, s. r. o. and Česká matematická společnost; 1st edition; 2012

Partee, Barbara H.; Meuen, Alice Ter; Wall, Robert E.; *Mathematical Methods in Linguistics*; Kluwer Academic Publishers; 1990

Pošta, Miroslav; *Technologie ve službách překladatele*; Miroslav Pošta – Apostrof; 1st edition; 2017

Čermák, František; *Korpus a korpusová lingvistika; Nakladatelství Karolinum; 1st edition; 2017*

Pořízka, Petr; *Tvorba korpusů a vytěžování jazykových dat metody, modely, nástroje*; Nakladatelství Filozofické fakulty Univerzity Palackého; 1st edition; 2014

Vopěnka, Petr; *Vyprávění o kráse novobarokní matematiky*; 2nd edition; Práh; 2016

Vopěnka, Petr; Úvod do klasické teorie množin; Nakladatelství Západočeské university v Plzni, Nakladatelství Fragment; 1st edition; 2011

Weaver, Warren; Shannon, Claude Elwood; *Mathematical Theory of Communication*; published in 1949; 1st edition

Bojar Ondřej, Rosa Rudolf, Tamchyna Aleš; *Chimera – Three Heads for English-to-Czech Translation*; URL: https://www.aclweb.org/anthology/W13-2208.pdf; [Date: 30. 11. 2020]

Singh, Shashi Pal; Kwnar, Ajai; Darbari, Hemant, Singh, Lenali; Rastogi, Anshika; Jain, Shikha; *Machine Translation using Deep Learning: An Overview*; URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8003957; [Date: 30. 11. 2020]

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, Jeffrey Dean; *Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation*; URL: https://arxiv.org/pdf/1609.08144.pdf; [Date: 30. 11. 2020]

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, Jeffrey Dean; *Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation*; URL: https://www.mitpressjournals.org/doi/pdfplus/10.1162/tacl_a_00065; [Date: 30. 11. 2020]

Kyunghyun Cho, Bart van Merrienboer Caglar Gulcehre, Fethi Bougares, Holger Schwenk, Dzmitry Bahdanau, Yoshua Bengio; *Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation*; URL: https://arxiv.org/pdf/1406.1078.pdf; [Date: 30. 11. 2020]

Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan N. Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer and Jakob Uszkoreit; *Tensor2Tensor for Neural Machine Translation*; URL: https://arxiv.org/pdf/1803.07416.pdf; [Date: 30. 11. 2020]

Popel, Martin, Tomková, Markéta, Tomek, Jakub, Kaiser, Lukasz, Uszkoreit, Jakob, Bojar, Ondřej, Žabokrtský, Zdeněk; *Transforming machine translation: a deep learning system reaches news translation quality comparable to human professionals*; URL: https://www.nature.com/articles/s41467-020-18073-9; [Date: 30. 11. 2020]

Marta R. Costa-jussà, Alexandre Allauzen, Loïc Barrault, Kyunghun Cho, Holger Schwenk; *Introduction to the special issue on deep learning approaches for machine translation*; URL: https://www.sciencedirect.com/science/article/pii/S0885230816303965; [Date: 30. 11. 2020]

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, Marc'Aurelio Ranzato; UNSUPERVISED MACHINE TRANSLATION USING MONOLINGUAL CORPORA ONLY; URL: https://arxiv.org/pdf/1711.00043.pdf; [Date: 30. 11. 2020]

Philipp Koehn, Rebecca Knowles; *Six Challenges for Neural Machine Translation*; URL: https://arxiv.org/pdf/1706.03872.pdf; [Date: 30. 11. 2020]

Zákon č. 121/2000 Sb., Hlava I, Díl 5, §43, URL: http://zakony.centrum.cz/autorsky-zakon/cast-1-hlava-1-dil-5-paragraf-43; [Date: 30. 11. 2020]

András Farkás; *LF Aligner;* URL: https://sourceforge.net/projects/aligner/; [Date: 30. 11. 2020]

9 Abstract

This bachelor thesis entitled “Processing of translations between languages: software methods, artificial intelligence and their advantages and disadvantages“ deals with the connection of mathematics, informatics and linguistics in the field of the mathematical linguistics with emphasis on the translation between languages.

The work is structured in four parts – three theoretical parts and one practical part. The first part is focused on the theory and history of the mathematical linguistics and describes the development of mathematical linguistics from the point of view of a mathematician, including the usage of statistical analysis.

The second part is focused on the description of several computer methods of machine translation and their inner workings from the point of view of the IT expert.

The third part is focused on the description of several software methods of machine translation, computer assisted translation and corpus linguistics from the point of view of the translator as the end user.

The fourth part describes practical procedure of creating the corpus and translation memory from the book by Jeff Lindsay “Darkly dreaming Dexter“ and its Czech translation “Drasticky děsivý Dexter“.

10 Resumé

Tato bakalářská práce nazvaná „Zpracování překladů mezi jazyky: softwarové metody, umělá inteligence a jejich výhody a nevýhody“ se zabývá spojením matematiky, informatiky a lingvistiky v oblasti matematické lingvistiky s důrazem na překlad mezi jazyky.

Práce je rozdělena do čtyř částí – tří teoretických částí a jedné praktické části. První část je zaměřena na teorii a historii matematické lingvistiky a popisuje vývoj matematické lingvistiky z matematikova úhlu pohledu, včetně užití statistické analýzy.

Druhá část je zaměřena na popis několika softwarových metod strojového překladu a jejich vnitřního fungování z informatikova úhlu pohledu.

Třetí část je zaměřena na popis několika softwarových metod strojového překladu, počítačem podporovaného překladu a korpusové lingvistiky z úhlu pohledu překladatele jakožto koncového uživatele.

Čtvrtá část popisuje praktický postup vytváření korpusu a překladové paměti z knihy od Jeffa Lindsaye „Darkly dreaming Dexter“ a jejího českého překladu „Drasticky děsivý Dexter“.