International bilingual journal in
Humanities & social Sciences

Millennium
მილენიუმი

Academy for Digital
Humanities - Georgia

Vol.2
2024

The Peer-reviewed Bilingual Scientific Online Journal of the Academy for Digital Humanities

in Humanities and Social Science

დიგიტალური ჰუმანიტარიის აკადემია - საქართველოს
ორენოვანი საერთაშორისო ახალგაზრდული სამეცნიერო რეფერირებადი ონლაინჟურნალი

მილენიუმი

Millennium

Volume 2

Tbilisi / თბილისი

2024

The peer-reviewed bilingual scientific online journal **Millennium** was initiated by the founders of the Academy for Digital Humanities and is designed for young researchers working in Humanities and Social Sciences – master students, doctoral students and postdocs. Millennium aims to promote the academic development of a new generation of scientists by creating an academic platform for young researchers working in the Humanities and Social Sciences to publish scientific papers.

დიგიტალური ჰუმანიტარიის აკადემიის რეცენზირებადი ელექტრონული ბილინგვური სამეცნიერო ჟურნალი **„მილენიუმი“** დაარსდა დიგიტალური ჰუმანიტარიის აკადემიის დამფუძნებელთა მიერ და განკუთვნილია ჰუმანიტარიასა და სოციალურ მეცნიერებებში მოღვაწე ახალგაზრდა მკვლევართათვის - მაგისტრანტების, დოქტორანტებისა და პოსტდოქტორანტებისათვის. ჟურნალი „მილენიუმი“ მიზნად ისახავს მეცნიერთა ახალი თაობის აკადემიური განვითარების ხელშეწყობას - ჰუმანიტარულ დარგებსა და სოციალურ მეცნიერებებში მომუშავე ახალგაზრდა სპეციალისტებისათვის აკადემიური პლატფორმის შექმნას სამეცნიერო ნაშრომების გამოსაქვეყნებლად.

ISSN 2960-9887 (Online)

Content / სარჩევი:

I. Digital Kartvelology / დიგიტალური ქართველოლოგია

II. Corpus Linguistics / კორპუსლინგვისტიკა

III. Language Technologies / ენის ტექნოლოგიები

IV. Political Linguistics / პოლიტიკური ლინგვისტიკა

V. Comparative literature studies / შედარებითი ლიტერატურათმცოდნეობა

VI. History / ისტორია

Can Counting Words *make* Sense?
A Simple Application of Visual Tools for Basic Machine Learning in Linguistics

Marc-Daniel Rahn
(Frankfurt University)

DOI: https://doi.org/10.62235/mln.2.2024.8000

marcrahn51@gmail.com || ORCID: 0009-0003-6744-372X

Abstract: The field of Natural Language Processing (NLP) is rapidly advancing with the development of tools for automatic text and language processing. The introduction of Chat GPT and other similar tools has sparked discussions about the use of artificial intelligence (AI) in society and academia. These tools are typically based on machine learning (ML), which allows computer programs to learn from data and generate models to make decisions. While ML models can be trained on language data to predict answers to research questions, they are not always easily accessible or optimized for linguists. This paper proposes using simple graphical tools for machine learning to help linguists formulate research questions and hypotheses, thus allowing them to assess the potential for fruitful investigations with specific data sets. By providing a user-friendly interface, these tools aim to overcome the barriers that deter linguists from utilising machine learning techniques. To illustrate the application of such a tool for preliminary studies, three experiments are described in this paper, all of which represent real questions of interest from within the field of Caucasiology. In the appendix, a step-by-step guide to recreate the process is given.

Keywords: Machine Learning, Caucasiology, Linguistics, Graphical Tools

Introduction

Very new to computer Sciences? Here’s some words to know beforehand:

Program. A program is anything a computer can run, either interacting with the user or running in the background.

Algorithm. An algorithm is any piece of computer code that takes in some information and puts out some other information. For example, an algorithm may take in two numbers and put out another number (like their sum), or take in an image and put out the number of pixels in that image, or, as is the case for most applications described here, take in a dataset (like a corpus of text) and put out a model of what these texts are about.

Model. A model is an abstraction of how something works. For example, we have a model of how gravity works (the theory of gravity) and can use it to predict what an object will do when we let go of it. In the field of machine learning, a model is the end result of **training**. The well known ChatGPT (Brown et al. 2020) for example has been trained on a lot of text data, using its training algorithm, to build a model of how to create text in response to a prompt.

Prompt. This word is associated with generative AI. It is the input that language based models need to create a response to. Most AI-powered image creation tools use language based models to understand prompts given by the user, which are then used to create the image.

1. The Typical Buzzwords

Artificial Intelligence, Machine Learning, Neural networks, Transformers - all these are terms from computer science with a very concrete meaning. However, they are often colloquially used as mere buzzwords, standing for a multitude of techniques and used interchangeably. In this paper, my aim is to first clarify these terms and to show how they are actually used. Then, the field of machine learning will be explored a little further, with a focus on applications in linguistics. Lastly, three simple experiments are done to show the potential and possibility of machine learning even for non-computer-science linguists. Let us begin by clarifying the aforementioned terms:

Colloquially, **Artificial Intelligence** (AI) means that a computer is able to learn (anything) and can act autonomously. This is what is often referred to as **strong AI** in computer science. A true strong AI does not exist yet (c.f. Diez 2022). On the other hand **weak (or narrow) AI** is the ability of a program to learn a very specific task, like ChatGPT has the ability to write in a human-like way, while Stable Diffusion models can create images. As our experience with AI has grown over the last years, the general term AI has been shifting in meaning to a more concrete understanding more closely related to weak AI.[1]

The actual core behind the **models** mentioned above is called **machine learning**. The term refers to the ability of a software to build and adapt a model of something — let's say how to play a game or how to speak a language — by being fed data on correct solutions for the task at hand, while trying to derive statistical tendencies from within that data. Many such models exist, some rather simple and some very complex, but the point here is that the way the software reaches a conclusion at the end is *not* pre-programmed into it from the beginning. Instead, through statistical inference, the program can "learn" how to deal with new data. The end result of machine learning is the creation of a model that an AI can use to make decisions.

1.1 Machine Learning

1.1.1 Most common techniques

There are many different techniques for machine learning, which I will not go over in detail in this paper (c.f. Alpaydin 2020 for an introduction to all these methods). However, here's a quick overview of some that are applicable to linguistics:

Regression Algorithms are machine learning methods where a model is built by finding or approximating a mathematical correlation (also called "trend") between data points. These are good, for example, to predict what the temperature will be like next year given the temperature data of previous years.

Clustering Algorithms like k-nearest-neighbor and many others try to divide data sets into smaller groups and are good for classifying tasks where speed is a factor.

Neural Networks are modeled after a very simplified version of the human brain and are multi-purpose learners, hence they are good at many different tasks. The downside is that they need large amounts of data to be trained. For many linguistic questions, especially for low- or mid-

[1] It is important to note that AI refers to the capability of a program to make decisions and does not specify at all how the program got these capabilities. In this sense, even the program controlling an enemy in a computer game is sometimes referred to as an AI, even though these "AIs" are not actually capable of learning, but instead often work with premade if-then statements.

ressource languages, such amounts of data might simply not be readily available to the researcher, making this a difficult route to take.

Transformers are a relatively new family of models introduced by the breakthrough paper "Attention is all you need" by Vaswani et al. (2017). The main application for these models is language generation, and they are the basis of programs like ChatGPT. Transformers improve on the older idea of neural networks by introducing the concept of "attention", that is, the continual awareness of what the conversation is about. Given the complexity of these models, it would exceed the scope of this paper to explain them here.

However, for most real applications, like ChatGPT or Stable Diffusion, a **multitude of techniques** are used to achieve a result. Often many heuristics, that is pre-programmed rules, are added to a program to reduce the reliance on pure data, due to the difficulty of coming up with that much data for any given problem. While ChatGPT has learned many languages, it can't learn a language very well for which it does not have such enormous amounts of data. While the training data set for GPT4 has not been made public to our awareness, the Training data size for GPT3 was 45 Terabytes of plain text, which was filtered down to 570 Gigabytes for the actual learning process.

With all these terms, it is important to understand that "**learning**" is not used in its colloquial everyday meaning. Human "learning" is a complex psychological process that cannot yet be fully explained. What is meant by learning here is the recognition of statistical correlations in the data. The extent to which this can be compared with human learning is an open debate in psychology, biology, information science and philosophy. Ultimately, learning itself is not the goal for machine learning algorithms - it is rather a necessary means to solve problems posed to the machine by humans, as Alpayidin (2020) clarifies:

> *"Note that unlike in psychology, cognitive science, or neuroscience, our aim in machine learning is not to understand the processes underlying learning in humans and animals, but to build useful systems, as in any domain of engineering."*
>
> *(Alpaydin 2020:14)*

Brief overview: Applications for machine learning *in linguistics*

There are many applications for machine learning techniques in linguistics, and this list is probably not exhaustive. Also, not all techniques fit every application, and new techniques are found all the time. However one can say that the most prominent applications for machine learning in computer linguistics right now are:

- Automatic word class annotation ("POS tagging")
- Automatic syntactic annotation
- Automatic morphological annotation
- Automatic annotation of semantic roles
- Optical character recognition (OCR for short)
- Text classification according to topic, genre, author, etc. ("Classification")
- Text synthesis (e.g. chatbots such as Chat GPT)
- Automatic translation ("machine translation")
- Recognition of spoken language ("Speech Recognition")
- Synthesis of spoken language ("Speech Synthesis")

1.1.2 Basic types of Machine Learning approaches

Despite the multitude of methods that have been introduced so far in this paper, all these have some things in common. Machine learning algorithms in general can be classified into two families of applications, sometimes combined into an iterative process, but ultimately being fundamentally different from each other. The next section explains these two routes.

1. **Supervised Learning:** *A clear task that can already be done by humans, but the computer should learn to do it instead of us to save us time.*

   Supervised learning is a machine learning technique where a computer learns from previously labeled data and applies this knowledge to new data. For instance, to teach a computer what a car looks like, it would need a large dataset of car images labeled as "car" and non-car images labeled as "not car". The computer would then analyze these images and identify common properties of cars, such as having four wheels and wheels being on the bottom. *The computer doesn't need to understand these properties*, but it - for example - recognizes that images labeled as "car" always have a circle of black pixels at the bottom. This learned knowledge can then be used to classify new, unfamiliar images. However, it's important to ensure that the training data is not biased or unbalanced. If, for example, most cars in the images are red, the computer may classify any red object in the middle of an image as a car, regardless of whether it is a car or not. To avoid this, we must **supervise** the computer by checking its results, then realizing that it is having trouble distinguishing “being red” from “being a car”, and then feed it more data showing red objects being labeled as “not car”, so that it can learn that “being a car” does not depend on “being red”.

2. **Unsupervised Learning:** *The data should be searched for yet unknown correlations which are difficult for humans to recognize.*

   This is called “unsupervised learning”. Here, we want the computer to find correlations (often called “trends” in statistics) in a set of data on its own. For example, the computer could again receive many images of cars and and not cars and try to find similarities (i.e. patterns) among these images. However, it would not know what it was really looking for, but would search for all the correlations (“regularities”, Alpaydin 2020:12) that it could find. In doing so, it would try to group all the available images in different ways. This is known as “clustering” (Alpaydin 2020: 11). This method does not need annotated data at all, but it also does not produce the same kind of results as supervised learning. Instead, the computer might choose to cluster the data in ways that are quite hard to understand for humans at all. It is not immediately clear what the computer actually recognizes and what criteria are used to establish these correlations. It may be that we as humans would obviously sort the images according to “cars” and “non-cars”; however, the computer could sort the images, regardless of their content, according to “lots of gray” and “lots of color”, or according to the color of the background or the uniformity of the texture, and accordingly group a very shiny cat together with the cars to “shiny”. Thus, with unsupervised learning in its basic form, we have practically no influence on these processes. Some approaches for unsupervised learning, such as decision trees, may put out what decision factors the computer chose, but again, these factors may be very unintuitive to humans.

1.1.3 Specific techniques for language processing: Bag of Words, N-Grams and Word Embeddings

In order to progress to the actual research question, we need to clarify three more concepts from the realm of machine learning: Bag of Words, N-Grams and Word Embeddings. All three are techniques to create numbers from a text or a collection of texts, since machine learning ultimately only works on numbers. Simply put, these are:

Bag of Words: We count the words in a given stretch of text, e.g per sentence, per paragraph etc. The result of "You are a nice person and a nice teacher" would be a list of all the words (or, a bit more experimental, parts of speech) in this sentence with a number next to each word for how often it occurred in that sentence.

You	Are	a	nice	Person	and	Teacher
1	1	2	2	1	1	1

PRON	V	DET	ADJ	N	CONJ
1	1	2	2	2	1

N-Grams: Instead of counting words, we count stretches of two or more words. This way we get a count of how often groups of words appear. The result for the sentence above for a 3-gram would be all groups of three words appearing next to each other (with overlaps). For the sentence above, 3-grams would give us: (you are a), (are a nice), (a nice person), (nice person and), (person and a), (and a nice) and (a nice teacher). All of these groups would then get a count of one, because no group occurs more than once in that stretch of text. One can easily see that this makes more sense in longer collections of text, and not so much in a single sentence. **Skip-grams** is a related technique referring to a method of predicting a missing word in a sequence of words.

Word Embeddings: Here it gets a bit more complicated. The first breakthrough on this technique was the program "Word2Vec" by Mikolov et al. (2013). We count how often any word occurs together with any other word in a specified maximal distance. This means in a text with 200 different words, we would get as a result a list of two hundred words, each with a count of how often they occurred together with any other word - basically a 200 by 200 table. If words do not occur together, the count is zero, if they do occur together, the count is a number greater than zero, namely how often they did occur together. This forms one vector of 200 numbers per word. These vectors can then be embedded into a geometric space consisting of 200 "dimensions" (quite hard to imagine visually for a human), and then the distance between any two such vectors can be calculated. The smaller the distance in this space, the more closely related two words are within the given data set. Word embeddings proved to be a major breakthrough in the linguistic utility for machine learning and are the basis for most sophisticated language models nowadays. However, they are not as easy to set up and take a relatively long time to calculate, which can be a limiting factor on their usability for a single researcher.

1.2. Most Common Problem Sources for Machine Learning in Linguistics

1.2.1. Character Errors *(Giving the Computer bad source material)*

These kinds of errors often result from bad optical character recognition (OCR) when trying to scan physical texts, especially if there are other layout elements than text (tables, columns, lists, images). These can however also be misspellings in a corpus, for example when using data from websites such as Facebook, Instagram, Twitter etc. To combat this, a certain amount of **Text Pre-Processing** is typically applied, cleaning the text files by using filters and regular expressions. This is not about changing the data, but rather about removing non-data from the data. Typical technical things to remove are emojis, URLs, HTML tags and so on. Depending on the research question, one might also want to remove punctuation, stop words[2], words in other languages and so on. One should keep in mind though that this is also a way to accidentally skew the data (See 1.3.3.), so this should be carefully documented and done with caution.

1.2.2. Inconsistent Annotations *(Teaching the computer wrong things)*

If our data would be an annotated corpus, then the way we prepare not only the entries (words and sentences etc.) in that corpus, but especially the metadata attached to these entries plays a tremendous role in the training process. However, many corpus projects have been worked on by many different people over certain time periods, often years or decades. Therefore, both diachronic (over time) and synchronic (between the annotators) inconsistencies are a typical occurrence, confusing the computer.

1.2.3. Skewed Data *(Giving the computer one-sided examples)*

Even more subtle than inconsistencies in the annotations is the problem of skewed data. What if our data was not really representative of the phenomenon we want to research in the first place? And how can we even find out if that is the case? This is a general problem in corpora, but here it is even more pronounced, since the computer can not reason or draw from experience in the same way that a trained linguist could. In general, any corpus is not to be confused with the actual language it is created from, since it can only ever be a snapshot of that language. (c.f. Durell 2015, Rissanen 2018)

1.2.4. Overfitting *(Letting the Computer obsess over Detail)*

A common error occurring in machine learning is called overfitting. When overfitting, the model is too precise or the computer spent too many steps on learning, leading to a degradation of actual understanding. Instead the computer learns the data "by heart".[3] Thus typically, the training of a machine learning model is stopped artificially after a time, because otherwise it would lose its capability to work on data not seen before.

[2] Stop words can be defined differently by research question, but the most common stop words are function words that occur very often in a language, such as articles or prepositions.

[3] Imagine a hypothetical student that can read a text once and remember it perfectly - they will be very good at answering questions that could be read directly out of the texts, and thus pass many tests perfectly, but they will struggle to come up with their own answers to problems that require a level of abstraction, because due to their perfect memory, they never *had to* truly understand what they were memorizing.

1.2.5. Underfitting *(Giving the computer unfulfillable tasks)*

When underfitting, the type of model the program is building in the learning process is too simple to capture the reality of a phenomenon. For example, a line can approximate points on a curve, but it can’t really capture the curve as a whole, as we see in the following graphic (Fig. 1). Underfitting can be avoided by making use of more complex models.

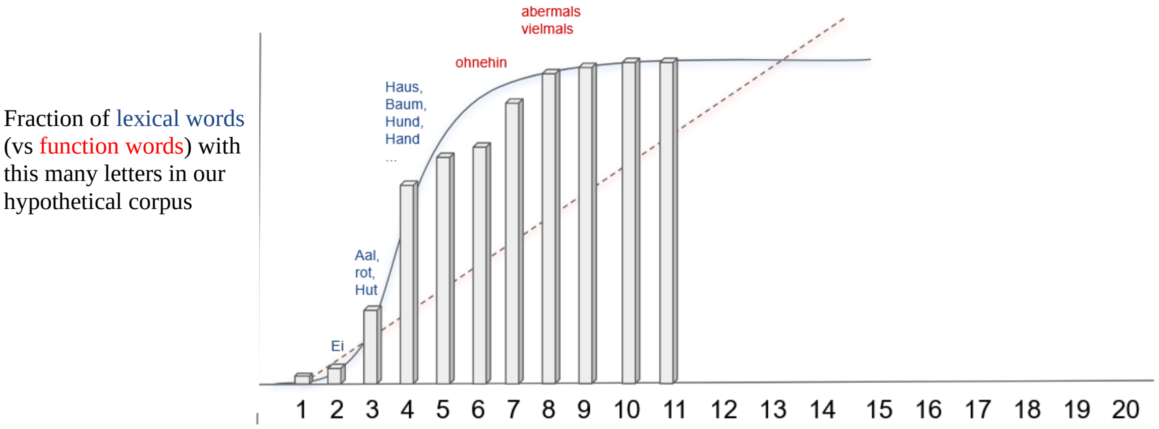


Fig. 2: Example (not real data) of a logistic regression algorithm. The gray bars show the data situation in a hypothetical training corpus: with increasing word length, function words become increasingly rare, while lexical words make up a greater percentage of all words . The algorithm attempts to find a (logistic) curve that approximates the data as closely as possible (the smooth curve). The dashed line corresponds to a linear regression algorithm. However, this can only ever approximate the data poorly (Underfitting). Logistic regression can now, for example, estimate the question “What is the probability that a word with 12 letters is a function word?” (Rahn, 2023, Translated by the author).

If that’s not already enough: some more words to know from machine learning

- **Pre-Training:** Training a model in a general task, to then train it even more on a specific task. For example it may (but i’m not an expert here!) make sense to train a model on the georgian language before training it on the imeretian dialect of georgian.
  This might also be done if there is simply not enough data on the imeretian dialect to train a model from scratch on that data alone. This is akin to first learning to drive a car in general, then learning how to drive a specific car, for example a race car.

- **Deep Learning:** This refers to an especially complex neural network being trained. Nowadays, most neural network learning is deep learning. Deep learning is especially powerful on large data, but the more complex a neural network is, the harder (or basically: impossible) it is to understand how it comes to its conclusions, since it is not easily possible for us to look into the information encoded in the neural network.

- **Active Learning**: This is an extreme case of supervised learning (c.f 1.2.1). Here, every data point fed to the machine learning algorithm is chosen by hand, basically shortening the time between training, evaluation, and re-training the model. This potentially yields the best results on smaller data sets, but is extremely ressource and time-intensive.

2 . Modern Graphical Tools for Linguists

Understanding these techniques in their detail (we just skimmed the surface here) may seem like an extremely daunting task to many, especially if they are not coming from the field of computer science, and actually using these techniques may seem just impossible. As is often the case with “anything computer” then, software experts are recruited which often are not coming from the research discipline in question, making the process of working with each other difficult and misunderstandings a common occurrence. The researcher, say a linguist, often does not understand the software expert, and at least as importantly, the software expert does not necessarily understand the questions that the linguist is posing to him in their full extent. This is where modern tools come in that let linguists do basic machine learning tasks on their own computers, with limited technical knowledge, and without the need to learn a programming language.

2.1 Orange Data Mining

There are many applications that allow linguists to tap into statistical model generation., with the most popular applications probably being NLTK (Bird et al. 2009), SpaCy (Honnibal et al. 2020), Gensim (Řehůřek & Sojka 2010) and Stanford Core NLP (Manning et al 2014) - most of these however require coding skills in different programming languages to use. On the other hand, there are pre-built programs like KHcoder (Higuchi 2016), Lancsbox (Brezina et al. 2015) and many others, some freely available, some quite expensive, that can do machine learning and classification tasks for you.

Orange Data Mining (Demsar et al. 2013) is special in a way, because it is more flexible than completely pre-written programs which do a fantastical job on the task they are made for, but do not allow for real alteration of their inner workings for any particular task. In orange however, one can assemble one's own “program” from building blocks without any coding, without installing special packages and without running command line tools etc.

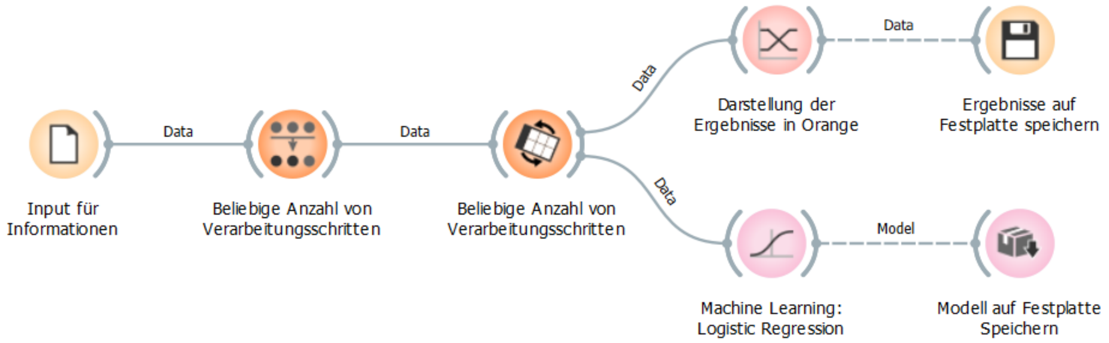

Input für Informationen
Data
Beliebige Anzahl von Verarbeitungsschritten
Data
Beliebige Anzahl von Verarbeitungsschritten
Data
Darstellung der Ergebnisse in Orange
Data
Ergebnisse auf Festplatte speichern
Data
Machine Learning: Logistic Regression
Model
Modell auf Festplatte Speichern

Fig. 3: A simple program sequence in orange. The circles represent individual nodes. Information is fed in on the left, which is then processed further. This processed data can then be visualized with a graph, for example, or it could be learned by a machine learning model. The results of these processes can be saved for later use.

In a previous work (Rahn 2023), I used Orange Data Mining to “solve” some example cases of “Pre-Studies” a linguist might do to understand if the data they are working with contained any tendencies that might be useful to their research question. My goal with this was that a linguist could easily understand all the steps taken and reproduce them without doing them on blind faith. My secondary goal was to only use simple methods, which would only work if tendencies were present at all and easy to find for the program. With tools like this, linguists can do their own pre-research, see if they even find potential trends in their corpus data, and then can think about how to refine these research endeavors by applying more sophisticated models of machine learning to them. With these insights, we come to our actual research question:

2.2 Overarching Research Question

Is it possible to use a simple "bag of words" approach to help linguists formulate hypotheses and get preliminary results for their research questions so that they can later be substantiated or falsified by more complex methods (e.g. word embeddings)?

Hypothesis: *If we can use the simplistic Bag of Words methods to find statistical tendencies lining up with reality, then we should be able to predict results with a better-than-chance rate of success.*

So, if we can find statistical tendencies lining up with reality, having any kind of predictive power on a given set of data with simplistic methods, then we can assume that with more sophisticated methods, better results can be achieved, and we can then think about going to a data scientist with those results and let them refine the methodology in numerous ways.

2.3 Specific Research Questions

In this specific case, we tried the bag of words methods in two classical ways and one not-so-usual way to "solve" three actually interesting test research questions:

1) Is it possible to assign verses from German Rustaveli translations to their authors using a bag-of-words approach (i.e. counting word frequencies)?

2) Is it also possible with such an approach to assign verses from different Georgian epics to the respective text?

3) Is it possible to use Bag of Words to assign Khinalug participles to different functional categories?

To clarify what we mean by "is it possible" is that we can reliably achieve better-than-chance results through these means, not that these questions can be definitely answered by these. So, for example, if there are three georgian epics that a verse might belong to in 2), and we can correctly predict the epic 50% of the time in a reliable manner (that is, testing for statistical significance), then we will count this as a success, because with three options to pick from, a chance pick would be right only 33% of the time. This might understandably sound unsatisfying to any linguist who would like to achieve 100% reliability in a method, but keep in mind that the goal here is to do some kind of quick to prepare **pre-research**, not a full inquiry on the topic, which might require far more advanced techniques not readily available to any linguist researcher.

> *"[With machine learning,] we believe we can construct a good and useful approximation. That approximation may not explain everything, but may still be able to account for some part of the data. We believe that though identifying the complete process may not be possible, we can still detect certain patterns or regularities. This is the niche of machine learning."*
>
> (Alpaydin 2007:2, Context added by the Author)

2.4 Using Orange to prepare the data and run the experiments

To prepare for the tasks, a series of steps had to be taken which are listed here.

I. Preparation of the Training Data

In order to use machine learning on any data set, Orange Data Mining needs to be able to read the data. The program can read Google **Spreadsheets**, so all data was prepared as such, with each Spreadsheet consisting of many rows, but only two columns.

- **Column A** was the data at hand, so for example a word, a stretch of words, or in the case of Khinalug a stretch of POS-tags.
- **Column B** was the correct solution for the classifying task, so that the program could learn what correct solutions would look like. For the Rustaveli translations, these were the names of the translator that produced the sentence given in Column A, for the Georgian epics it was the author, and for Khinalug it was the type of Participle in that sentence. It is important that these annotations are very consistent.

II. Preparation of the Test Data

This was done exactly like the training data, but the correct answers in Column B were not given but had to be supplied by the program after training. The training data was taken from the same corpus, but no datum that appeared in the training data set was used in the test data set.

III. Graphical Programming in Orange

Lastly, a program was “written” in Orange Data Mining to actually do the machine learning. The exact process of this is described **in the Appendix** to this paper, specifically for the case of Khinalug, since this was the most complex application of the program. The other tasks used simplified versions of that same program. The methodology and results for each program are outlined in the following section.

2.4.1 Task 1: Classifying Georgian Epics

Research Question

The question here is whether Bag of Words can be used to build a simple method for reliably assigning verses to one of three Georgian texts. These were “Tamariani”, written by Chakhruchadze (18th century), “Šāhname”, translated from Persian into Georgian between the 16th and 17th centuries, and “The Knight in the Panther Skin”, created by Rustaveli at the end of the 11th century, but not written down until the 17th century. These texts were taken from TITUS (Gippert, 1995a). The main difficulty is that the verses of these texts often contain only a few tokens, as Georgian can convey a lot of meaning with just a few (but morphologically complex) words.

Methodology

The texts were collected from TITUS (Gippert, 1995a) and transferred into Excel tables, with each verse corresponding to one row. These were then divided into two sets of data, with the first 1296 verses of each text used as training data, resulting in a total of 3888 training examples. From the remaining text material, 100 sentences per text were randomly selected as test data. Punctuation marks and special characters were removed in preprocessing, while function words were retained. Using the bag of words approach, the data was fed to three different algorithms, and the overall result was determined by a majority vote.

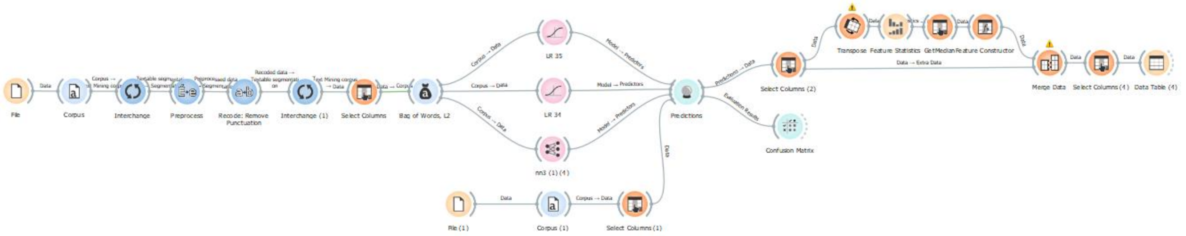


Fig. 4: Program flow in Orange Data Mining. Pre-processing on the left, the actual machine learning in the middle, and evaluation on the right using a self-built “voting” variant.

Results*:* Despite the small number of words per verse, the results were surprisingly good, reaching an accuracy of over 85% for the first two algorithms (Logistic Regression) and over 78% for the third (Neural Network).

Classification Accuracy (CA) of the three models			
Modell	**CA (1296)**	**CA (648)**	**CA (324)**
Logistic Regression 35	85,81 %	58,40 %	53,89%
Logistic Regression 34	85,52 %	57,83 %	54,51 %
Neural Network	78,93 %	64,01 %	61,17 %

Table 1: Classification Accuracy of the three different models. The training set was halved in two consecutive steps as a test for how the accuracy would diminish on a smaller training set.

2.4.2 Task 2: Classifying German Translations of “The Knight in the Panther Skin”

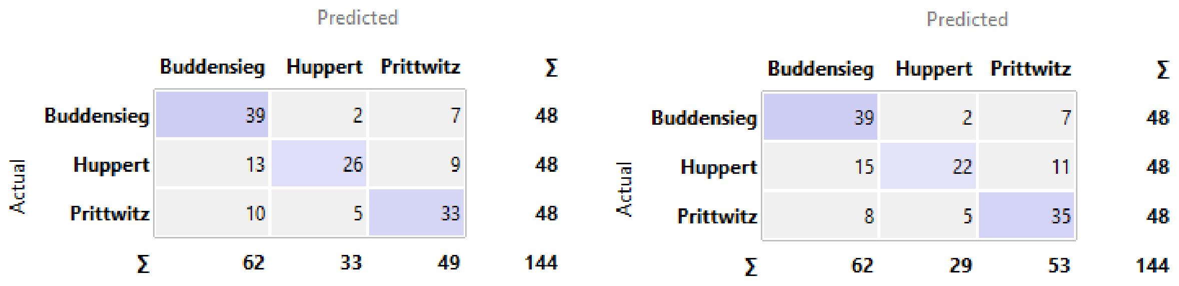
The question was based on a specific research interest by Dr. Manana Tandashvili. There are three well-known poetic translations of the Georgian national epic “The Knight in the Panther Skin” for German, namely by Hermann Buddensieg (1976), Marie Prittwitz (Unknown, printed in 2011) and Hugo Huppert (1955). However, Marie Prittwitz's original manuscript from the 1930s was lost and thus not available to scientific research until recently, only being rediscovered decades later and then published in 2011, long after the other two translations had been published. The relationship between these texts is therefore unclear, but some similarities in the translations are noticeable, especially in additions and interpretations in the text where different word choices and structures would actually have been expected due to the interpretative scope (Manana Tandashvili, PC, August 5, 2023). In this sense, it was of interest to highlight the similarities and differences between these texts. The idea was as follows: If machine learning can accurately assign verses from these texts to their authors, it could be informative for further research to take a closer look at those verses that the algorithm had difficulties with.

Methodology

The texts of the three translations are publicly available. Of these texts, the first four verses of each of the first 27 chapters (i.e. 16 verses each) were transferred to an Excel table (one line per verse) and the authors were annotated, resulting in a total of 1296 training examples (432 per author). Specific attention was paid to balancing the training data. All punctuation and other special characters were removed from the training data; Function words were kept because they are relevant to identifying each author's style. This training data was in turn analyzed using

“Bag of Words” and the resulting numbers were fed into three machine learning algorithms (two types of “linear regression” with different settings and a neural network). As a test set, 144 random verses (48 per author) from other parts of the epic were selected. Simple majority voting was again carried out between the models for the final result.

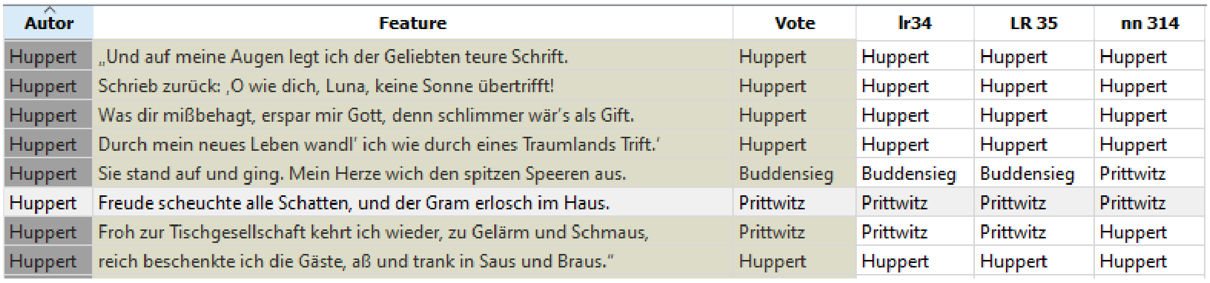
Results: The performance of the algorithms was not bad with an accuracy of around 68-72% (33% would be pure guessing since there were three categories), but not particularly good either, although this does help answer the original research question. Looking at which sentences the algorithms had difficulties with, it is evident that especially the verses by Huppert (1955) were often assessed as either “Buddensieg” or “Prittwitz” (see Fig. 8).



		Predicted			
		Buddensieg	Huppert	Prittwitz	Σ
Actual	Buddensieg	39	2	7	48
	Huppert	13	26	9	48
	Prittwitz	10	5	33	48
	Σ	62	33	49	144

		Predicted			
		Buddensieg	Huppert	Prittwitz	Σ
Actual	Buddensieg	39	2	7	48
	Huppert	15	22	11	48
	Prittwitz	8	5	35	48
	Σ	62	29	53	144

Fig. 5: Confusion matrices of the algorithms used. On the left a logistic regression algorithm, on the right a neural network. If the programs had not made any errors, then all the blue fields would show 48 and all the gray would show 0. All in all, most of the mix-ups occurred with “Huppert”.



Autor	Feature	Vote	lr34	LR 35	nn 314
Huppert	„Und auf meine Augen legt ich der Geliebten teure Schrift.	Huppert	Huppert	Huppert	Huppert
Huppert	Schrieb zurück: ‚O wie dich, Luna, keine Sonne übertrifft!	Huppert	Huppert	Huppert	Huppert
Huppert	Was dir mißbehagt, erspar mir Gott, denn schlimmer wär's als Gift.	Huppert	Huppert	Huppert	Huppert
Huppert	Durch mein neues Leben wandl' ich wie durch eines Traumlands Trift.'	Huppert	Huppert	Huppert	Huppert
Huppert	Sie stand auf und ging. Mein Herze wich den spitzen Speeren aus.	Buddensieg	Buddensieg	Buddensieg	Prittwitz
Huppert	Freude scheuchte alle Schatten, und der Gram erlosch im Haus.	Prittwitz	Prittwitz	Prittwitz	Prittwitz
Huppert	Froh zur Tischgesellschaft kehrt ich wieder, zu Gelärm und Schmaus,	Prittwitz	Prittwitz	Prittwitz	Huppert
Huppert	reich beschenkte ich die Gäste, aß und trank in Saus und Braus."	Huppert	Huppert	Huppert	Huppert

Fig. 6: All models estimate the highlighted verse as written by Prittwitz, even though it comes from Huppert. In the example directly above or below, the models do not agree. It is probably particularly interesting to compare sentences like the highlighted one with the other versions.

Specifically, both models tend to assess texts by Huppert as coming from either Prittwitz or Buddensieg, but not the other way around. The Logistic Regression algorithm showed an accuracy of 81.25% when assessing the Buddensieg verses and 78.72% for the Prittwitz verses, but only an accuracy of 54.17% for the Huppert verses. This results in the average accuracies shown in Table 3 of approximately 67%. However, a t-test between the distribution of predictions (62, 33, 49) and a uniform distribution (48, 48, 48) only results in a statistical significance of p = 0.51 (barely not significant). A study with more data could possibly shed more light on this connection. I refrain from interpreting these results here, leaving this to the experts in the field of Rustaveli translations, but the result surely is encouraging further investigation. After the actual study, a performance test was carried out in relation to the size of the training data by halving it and repeating the calculation. Since there would still be a lot of text material available, the training data could be duplicated again, but this was not done within the limited scope of this investigation.

Mean Classification Accuracy (CA) of the chosen models		
Model	**CA (Full training data)**	**CA (Half training data)**
Logistic Regression 35	67,05 %	63,21 %
Logistic Regression 34	66,83 %	64,60 %
Neural Network	66,65 %	66,74 %

Table 2: Average performance of the different methods, best performance per category highlighted.

4.2.1 Classifying Participles in Khinalug

In Khinalug, a northeast caucasian language with approximately 3000 speakers, participles can be divided into five categories: Attributive, modal, phrase-final and back-and-forth movement, abbreviated as A, M, P and B in the following text. (Rind-Pawlowski, PC, 15.07.2023). The question is whether the Bag of Words method is able to classify sentences containing a participle into these categories. Only a small training data set was available for training, making the task more very difficult for the machine.

Methodology

A morphologically annotated corpus of Khinalug was kindly made available to me by Dr. Rind-Pawlowski. From this corpus, 166 sentences with only one participle were selected and annotated as containing that participle type. The sentences were chosen at random and were not balanced (i.e. in this specific case there were more sentences in category "A" than "M"). As the amount of training data was so small, the occurring **parts of speech** (which were already annotated in the corpus) **instead of the words themselves** were counted per sentence with "Bag of Words". A sentence such as "The man is sitting." would thus have been read by the machine as "DET N AUX PTCP". The machine was not provided with any data other than the POS annotation of these 166 sentences. Eight different models were trained and the three with the best performance were selected. Then, unseen sentences were given to the machine, with the three models again voting for the end result forming the output.

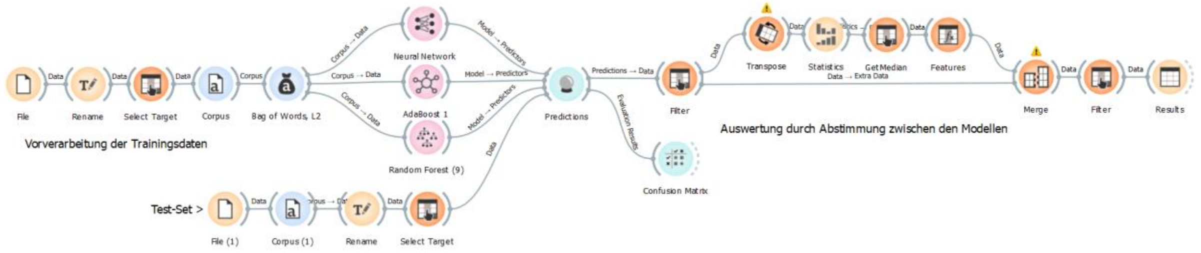


Fig. 7: The program sequence for the evaluation of the Khinalug participles (larger in Appendix B) - on the left the preprocessing, in the middle (in pink) the actual machine learning, on the right the evaluation via voting between the models. The test set is introduced at the bottom left.

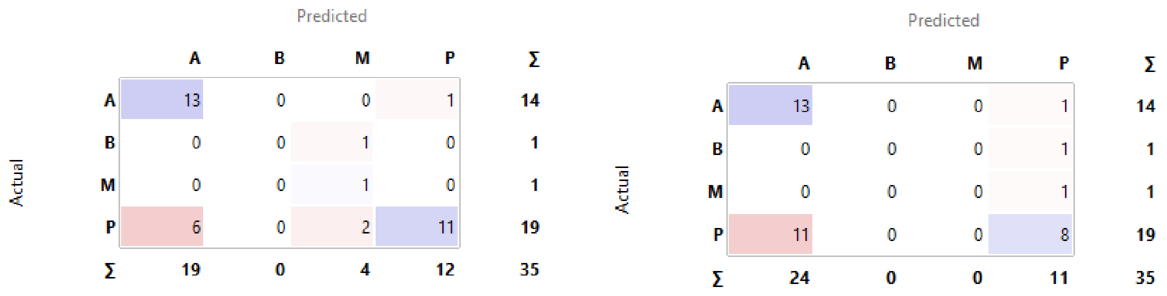
The outstanding feature of this study was that a technique that is normally used to classify entire texts was instead used to classify sentences. This approach was only possible by using the parts of speech instead of the individual words, as otherwise the training dataset would have been too small for machine learning. Since only sentences containing exactly one participle were selected, the classification of the sentences could be interpreted as a classification of those participles. The machine learning methods used were firstly a neural network, secondly AdaBoost, which is based on decision trees, and thirdly a random forest, which is also based on decision trees.

Results: Considering the chosen method, which may seem unsuitable and very unorthodox for the task at first, and the fact that the training data was not balanced, the results were not bad with over 70% accuracy (see Table 2). With four categories, 25% accuracy would be expected if the models only guessed randomly. However, if they only guessed "A" (which was the most common point in the training data), 59% accuracy would be expected. This second point is the biggest weakness of this study. To address this weakness, more training data would be needed, which unfortunately was not available.

Classification Accuracy (CA) of the models	
Modell	**Classification Accuracy (CA)**
Random Forest	71,4 %
AdaBoost	71,4 %
Neural Network	60,0 %

Distribution of training data		
Category	**Absolute**	**Relative**
A	98	59,03 %
B	6	3,61 %
M	17	10,24 %
P	45	27,11 %
Total	166	100 %

Table 3: Comparison of the performance of the models used (given as a percentage). Classification Accuracy indicates how often the machine was correct on the test data. The table on the right is an overview of the actual distribution of the categories in the (evidently unbalanced) training data.



Predicted

Actual	A	B	M	P	Σ
A	13	0	0	1	14
B	0	0	1	0	1
M	0	0	1	0	1
P	6	0	2	11	19
Σ	19	0	4	12	35

Predicted

Actual	A	B	M	P	Σ
A	13	0	0	1	14
B	0	0	0	1	1
M	0	0	0	1	1
P	11	0	0	8	19
Σ	24	0	0	11	35

Fig. 8: On the left the predictions of "Random Forest", on the right those of the neural network. Blue fields are "correct", red fields are "incorrect", i.e. mix-ups. Most of the errors stem from the fact that "P" was incorrectly classified as "A" (red field bottom left). The confusion matrices show the unbalanced nature of the training data: Since A is by far the most common type of participle in the training data (as in real language use), the machine assumes A rather than another type for new, unseen data. This shows that the majority of errors consist of a different type being classified as A.

5. Conclusion

5.1 Results of the Experiments

The aforementioned experiments were all conducted in their own right, but also to answer a common research question, namely: "*Is it possible to use a simple "bag of words" approach to help linguists formulate hypotheses and get preliminary results for their research questions so that they can later be substantiated or falsified by more complex methods (e.g. word embeddings)?"*

In the experiments we found better-than-chance results even for cases where we did not expect them at all (like Khinalug). One should keep in mind though that *not all* machine learning models contained in Orange Data Mining achieved such results. That however shows us that for all three cases *some* machine learning methods were able to extract information out of the very limited data they were given, and that the data in our three experiments has at least potential for more sophisticated machine learning. This can be said with more certainty for the first two experiments (Rustaweli translations and Georgian Epics) than for the last experiment (Khinalug), since with the last experiment, there is a higher chance of statistical misrepresentation using this method. In other words, the last experiment has too small of a data set for conclusive results.

However, these preliminary results merit two thoughts: For the first two experiments, it might be useful to engage with computational linguists and to improve the methods used to achieve better results. On the other hand, it will probably be worthwhile expanding the set of annotated data for the third experiment, since the data has shown to probably be useful. There are multiple ways this could be done - either tagging sentences as “containing one of these types” or tagging the types specifically on the words, which would allow a program to search for sentences where these tags occur. The second approach is probably better, because it can be used for more than just this single application. This being said, the method proposed here clearly *can* help linguists decide what step to take next with their data. Therefore, it is clear that we can answer the research question in the positive.
What does this now mean, and why is it important?

5.2 Outlook

Firstly, Machine Learning and Artificial Intelligence are extremely rapidly developing fields, and both can easily become a kind of black box in the near future where only a few experts how they actually work anymore, even though anybody can use them via some interfaces (like it is the case with ChatGPT or StableDiffusion)

Secondly, these two fields seem to be almost inevitable in the future. More and more AI models are published every day for all kinds of use cases, and tools for linguistics are not an exception. However, most of these linguistic tools are either completely pre-made (but then can’t be changed to fit a specific research question) or act more as function libraries, needing a solid foundation in programming before being able to use them.

Having tools for *building* (and not only using) simple machine learning applications without pre-existing knowledge of programming makes it possible to understand at least roughly what is happening under the hood, enabling researchers to find tendencies and flaws in their data before using more complicated tools.

This is especially important for research on endangered or low resource languages, where funding by the private sector is hard to obtain and generally less likely. With such tools, it may become easier to make a case for funding a research based on preliminary results that have already been obtained.

Our hope is that this way, machine learning and artificial intelligence can be techniques that the whole scientific community can benefit from, instead of even the basics only being understood by highly trained experts working for private companies which do not share their code nor their insights with the wider public.

References

1. **Alpaydin 2020**: Alpaydin, E., *Introduction to machine learning.* MIT press.
2. **Bird, Klein & Loper 2009**: Bird, S., Klein, E., & Loper, E., *Natural language processing with Python: analyzing text with the natural language toolkit*. O'Reilly Media, Inc.
3. **Brezina, McEnery & Wattam 2015:** Brezina, V., McEnery, T. & Wattam, S., Collocations in context: A new perspective on collocation networks. *International Journal of Corpus Linguistics, 20(2), 139-173.*
4. **Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. 2020**: Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D., Language models are few-shot learners. *Advances in neural information processing systems, 33*, 1877-1901.
5. **Demsar, Curk, Erjavec, Gorup, Hocevar, Milutinovic, Mozina, Polajnar, Toplak, Staric, Stajdohar, Umek, Zagar, Zbontar, Zitnik, Zupan 2013:** Demsar J., Curk T., Erjavec A., Gorup C., Hocevar T., Milutinovic M., Mozina M., Polajnar M., Toplak M., Staric A., Stajdohar M., Umek L., Zagar L., Zbontar J., Zitnik M., Zupan B., Orange: Data Mining Toolbox in Python, *Journal of Machine Learning Research* 14 (Aug): 2349-2353.
6. **Diez 2020, September 01**: Diez, F., What is Artificial Intelligence and Machine Learning? *Blog of the EP-KI Team at Fraunhofer Institut, Germany.* https://www.itwm.fraunhofer.de/en/departments/fm/latest-news/blog/ki-maschinelles-lernen-blog_EN.html (Last accessed on 2024/07/06.
7. **Durrell 2015:** ‘Representativeness’,‘Bad Data’, and legitimate expectations. in: Gippert, J. / Gehrke, R. (Hrsg. / ed.) 2015. 13-33.
8. **Gippert 1995a:** J. Gippert, TITUS. Das Projekt eines indogermanischen Thesaurus, LDV-Forum 12/2, 1995, 35-47 (vgl. http://titus.uni-frankfurt.de/texte/titusldv.htm)
9. **Higuchi 2016:** Higuchi, K., KH Coder 3 reference manual. *Ritsumeikan University, Kyoto.*
10. **Honnibal, Montani, Van Landeghem & Boyd 2020:** Honnibal, M., Montani, I., Van Landeghem, S., & Boyd, A., spaCy: Industrial-strength natural language processing in python.
11. **Kalyan 2023**: Kalyan, K. S., A survey of GPT-3 family large language models including ChatGPT and GPT-4. *Natural Language Processing Journal, 100048.*
12. **Manning, Surdeanu, Bauer, Finkel, Bethard & McClosky 2014 (June)**: Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., & McClosky, D., The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations* (pp. 55-60).

13. **Mikolov, Chen, Corrado & Dean 2013:** Mikolov, T., Chen, K., Corrado, G., & Dean, J., *Efficient estimation of word representations in vector space. CoRR, abs/1301.3781.*

14. **Rahn 2023:** Rahn, M.D., *Machine Learning in der Korpuslinguistik mit Orange Data Mining anhand von Beispielen aus der Kaukasiologie* [unpublished]. Universität Frankfurt.

15. **Řehůřek & Sojka 2010:** Řehůřek, R., & Sojka, P., Software framework for topic modelling with large corpora.

16. **Rissanen 2018:** Rissanen, M., Three problems connected with the use of diachronic corpora. *ICAME journal, 42(1),* 9-12.

17. **Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez & Polosukhin 2017:** Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., & Polosukhin, I.,Attention is all you need. *Advances in neural information processing systems, 30.*

Historical Text References

1. Rustaveli, Š., & Huppert, H. (1955). *Der Recke im Tigerfell: altgeorgisches Poem*. Herausgegeben von der Gesellschaft für kulturelle Verbindung der Georgischen SSR mit dem Ausland, Rütten & Loening.

2. Rustaveli, Š. (1976). *Der Mann im Pantherfell: altgeorgisches Epos; Nachdichtung von Hermann Buddensieg*. Verlag Sabtschota Sakartwelo.

3. Rustaveli, Š., Prittwitz, M., & Chotiwari-Jünger, S. (2011). *Der Ritter im Tigerfell: ein altgeorgisches Epos*. Shaker.

4. Rustaveli, Š. Vepxisṭq̇aosani. The Old Georgian Poem on The Knight in The Panther's Skin. *On the basis of the editions by Aḳaḳi Šaniʒe (S), Tbilisi 1975 and A. Baramiʒe / Ḳ. Ḳeḳeliʒe / A. Šaniʒe (K), Tbilisi 1957 edited by Jost Gippert and Vaxṭang Imnaišvili, Frankfurt am Main, 31.1.1996.*

5. Čaxruxaʒe. Tamariani: The Old Georgian Ode to Queen Tamar. *On the basis of the editions by Nikolaj Jakovlevič Marr, Drevnegruzinskie odopiscy (XII v.), I. Pevec Davida Stroitelja; II. Pevec Tamary, S.-Peterburg 1902 and Ivane Lolašvili, Ʒveli kartveli mexoṭbeni I: Čaxruxaʒe, Keba mepisa Tamarisi, Tbilisi 1957, edited by Jost Gippert and Vaxṭang Imnaišvili, Frankfurt am Main, 10.3.1996.*

6. Firdausi, A. Šāhname. *On the basis of the editions Šah-names anu mepeta c̣ignis kartuli versiebi. Ṭeksṭi gamosca da c̣inasiṭq̇vaoba da leksiḳoni daurto Iusṭine Abulaʒem, (Sakartvelos saisṭorio da saetnograpo sazogadoebis gamocema. Ʒveli kartuli mc̣erloba, ṭ. 1) Tbilisi 1916 and Abu-l Q̣asim Pirdousi Ṭuseli: Šahname. Kartuli versiebi, ṭ. II / Abu-l Kasym Firdousi: Šahname. Gruzinskie versii, tom II / Abou'l Kasim Firdouçi: Le Chah-Nameh. Versions géorgiennes, t. II. Ṗrop. Iusṭ. Abulaʒis, Doc. Al. Baramiʒis, Ṗ. Ingoroq̇vas, Ṗrop. Ḳ. Ḳeḳeliʒis da Ṗrop. Aḳ. Šaniʒis redakciit, ḳomentarebit da leksiḳonit, Ṭpilisi 1934. digitized by Jost Gippert, Frankfurt 1996-1999; text correction and reedition by Tamaz Abašiʒe and Xatuna Todua, Tbilisi, 2001; ARMAZI version by Jost Gippert, Frankfurt am Main, 31.12.2001 / 17.3.2007.*

აქვს თუ არა სიტყვების დათვლას აზრი? ვიზუალური ხელსაწყოების მარტივი გამოყენება საბაზისო მანქანური სწავლებისათვის ლინგვისტიკაში

მარკ-დანიელ რანი
(ფრანკფურტის უნივერსიტეტი)

DOI: https://doi.org/10.62235/mln.2.2024.8000

marcrahn51@gmail.com || ORCID: 0009-0003-6744-372X

შესავალი

სტატია „აქვს თუ არა სიტყვების დათვლას აზრი?“ მიზნად ისახავს მანქანური სწავლების შესახებ ინფორმაციის მიწოდებას მკვლევართათვის, თუ როგორ შეიძლება გამოიყენონ მათ მარტივი მანქანური სწავლების პროგრამები და ტექნიკური ინსტრუმენტები, რათა განახორციელონ წინასწარი კვლევები ინფორმატიკის ცოდნის, კერძოდ, კოდირების გამოცდილების გარეშე. სტატიაში აღწერილია ტექნოლოგიური ინსტრუმენტი Orange Data Mining და მისი გამოყენების შედეგები სამი ექსპერიმენტის ბაზაზე. სტატია განკუთვნილია არაინფორმატიკოსებისათვის, უფრო კონკრეტულად - ლინგვისტებისათვის, რის გამოც საჭიროდ ჩავთვალეთ შესავალში კომპიუტერული მეცნიერების იმ საკვანძო / გასაღები სიტყვების განმარტება, რომლებიც მანქანური სწავლების სფეროში გამოიყენება, რათა სტატიაში განხილული საკითხით დაინტერესებულ პირებს მივაწოდოთ ზუსტი ცოდნა მანქანური სწავლების ძირითადი ცნებებისა და მათი აღმნიშვნელი ტერმინების შესახებ. სტატიის შესავალში ჩამოთვლილი ტერმინები, უმეტესად, ხშირად ხმარებული და, ერთი შეხედვით, ნაცნობი ტერმინებია, თუმცა საჭიროდ მივიჩნიეთ მათი მნიშვნელობის ახსნა მარტივი, სასაუბრო ენით, რათა მსჯელობა გასაგები ყოფილიყო კომპიუტერული ცოდნის არმქონე მკვლევართათვისაც.

1. მანქანური სწავლების არსი, მეთოდები და ინსტრუმენტები

სტატიის მომდევნო, მეორე ნაწილში ჩვენ ვსაუბრობთ მანქანური სწავლების ტექნიკების შესახებ, რომლებიც მიზნად ისახავენ მოდელების შექმნას. ამასთან აქცენტი კეთდება იმ ტექნიკებზე, რომელიც რელევანტურია ლინგვის-

ტიკის ამოცანების გადასაჭრელად. ესენია: კლასტერული ალგორითმების, რეგრესიის ალგორითმების, ნეიროლოგიური ქსელებისა და ახალი ტრანსფორმატორების ტექნოლოგიის გამოყენება ისეთ პროგრამებში, როგორიცაა ChatGPT.

სტატიაში ასევე აღნიშნულია, რომ მაშინაც კი, თუ ამ პროცესს „სწავლას" ვუწოდებთ, ის არ უნდა შევადაროთ ადამიანის უნარს ისწავლოს, პირველ რიგში იმიტომ, რომ ჩვენ ბოლომდე არ გვესმის რას ეფუძნება ადამიანის სწავლის პროცესი და როგორ ფუნქციონირებს ის, და მეორეც, იმიტომ, რომ მანქანური სწავლების ტექნიკები საჭიროებს ციფრულ მონაცემებს, რომლებიც გარდაიქმნება სიებად და რიცხვების შემცველ ცხრილებად, რათა შესაძლებელი იყოს მათზე სტატისტიკური ანალიზის გაკეთება, რაც მანქანური სწავლების მიდგომის საფუძველს წარმოადგენს.

სტატიის ამ ნაწილში ასევე ახსნილია მანქანური სწავლების პროცედუალური ცნებები: **სწავლება ზედამხედველობის მეშვეობით** და **სწავლება ზედამხედველობის გარეშე**. პირველი - **სწავლება ზედამხედველობის მეშვეობით** - გულისხმობს ისეთი ამოცანის შესრულების პროცესს, რომელიც შეიძლება გადაიჭრას მკაფიო, უკვე ცნობილი მიდგომების მეშვეობით, და რომლის შესრულებაშიც კომპიუტერს შეუძლია ჩაანაცვლოს ადამიანი, ხოლო მეორე - **სწავლება ზედამხედველობის გარეშე** გულისხმობს ისეთი ამოცანის შესრულების პროცესს, რომელიც კომპიუტერისაგან მოითხოვს ჯერ კიდევ უცნობი სტატისტიკური ტენდენციების პოვნას მონაცემთა დამუშავების გზით.

აქვე მიმოვიხილავთ საგანგებოდ გამორჩეულ მეთოდებს, რომელიც გამოიყენება ლინგვისტიკაში მანქანური სწავლების დროს. როგორც აღინიშნა, მანქანური სწავლება დამოკიდებულია იმაზე, თუ რამდენად ხელმისაწვდომია მონაცემები რიცხვების სახით, ციფრულად. ქვემოთ ჩამოთვლილია მონაცემთა დამუშავების ის ტექნიკები, რომლებიც გამოიყენება ტექსტების რიცხვებად გარდასაქმნელად. ესენი არიან: სიტყვების ტომარა (**Bag of words**), N-გრამები (**N-grams**) და skip-გრამები (**skip grams**), სიტყვათა კომბინაცია (**Word Embeddings**).

სტატიაში ასევე ვსაუბრობთ მანქანური სწავლების პროცესში შეცდომების გამომწვევ მიზეზებზე. განსაკუთრებით აღსანიშნავია ისეთი ტიპური შემთხვევები, როგორებიცაა:

1. ოპტიკური სიმბოლოების არასწორი ამოცნობის შედეგად მიღებული **წაკითხვის შეცდომები**;

2. ენობრივ კორპუსებში **არაერთგვაროვანი ანოტაციების** არსებობით გამოწვეული შეცდომები;
3. ისეთი მონაცემების დამუშავება, რომლებიც არ წარმოადგენენ რეპრეზენტაციულ მონაცემებს მოცემულ ფენომენთან მიმართებით;
4. მოდელის **გადამეტებული „მორგება“** ტექსტზე, რაც ნიშნავს კომპიუტერის ზედმეტ ვარჯიშს ძალიან მცირე მონაცემებზე;
5. მოდელის „**დაქვეითება**“, რაც ნიშნავს საკვლევი ობიექტისათვის ძალიან მარტივი მოდელის არჩევას.

3. გრაფიკული ხელსაწყოები

სტატიის მეორე თავში საუბარია გრაფიკულ ინსტრუმენტებზე, რომლებიც შექმნილია საგანგებოდ მანქანური სწავლებისთვის და ხელმისაწვდომია მკვლევართათვის პროგრამირების ძალიან მცირე ან წინასწარი ცოდნის გარეშეც. აქ ყურადღება გამახვილებულია ხელმისაწვდომ ინსტრუმენტზე Orange Data Mining, რომელიც შემუშავებულია სლოვენიის ლუბლიანას უნივერსიტეტის მიერ და წარმოადგენს ღია კოდის პროგრამას.

სტატიაში წარმოდგენილია ექსპერიმენტი, რომელიც მიზნად ისახავდა აღნიშნული ინსტრუმენტის გამოცდას ლინგვისტურ მასალაზე. მიუხედავად იმისა, რომ ეს ინსტრუმენტი არ არის შექმნილი საგანგებოდ ლინგვისტური კვლევებისთვის, ექსპერიმენტს უნდა გამოევლინა, თუ რამდენად სასარგებლო შეიძლება იყოს ლინგვისტებისთვის მარტივი მანქანური სწავლების თანამედროვე მიდგომა დახვეწილად მორგებული მეთოდებისა და ხანგრძლივი ტრენინგის გარეშე. შესაბამისად, ჩვენთვის მნიშვნელოვანი იყო არა კონკრეტულად ამ ინსტრუმენტის ტესტირების შედეგები, არამედ ზოგადად, ამგვარი ინსტრუმენტების ვარგისიანობისა და სანდოობის საკითხის გარკვევა. აქ გადამწყვეტი იყო კვლევის მასშტაბი: ჩვენ მიერ ჩატარებული ექსპერიმენტი შეიძლება განხორციელდეს ერთი ენათმეცნიერის მიერ პროგრამირების ცოდნის გარეშე მოკლე დროში. ამასთან, ექსპერიმენტს უნდა გამოევლინა, შეუძლია თუ არა ამგვარ ექსსპერიმენტებს სარგებლობის მოტანა კვლევის შედეგების ადეკვატურობის თვალსაზრისით. უფრო კონკრეტულად, რამდენად გამოსადეგია მანქანური სწავლების ინსტრუმენტის გამოყენება წინასწარი კვლევის ჩასატარებლად, სანამ კონკრეტული საკითხით დაინტერესებული მკვლევრები დაიქირავებენ კომპიუტერული ლინგვისტიკის სპეციალისტებს აღნიშნული საკითხის საფუძვლიანი კვლევის ჩასატარებლად.

3. მეთოდოლოგია

მანქანური სწავლების ინსტუმენტები, რა თქმა უნდა, ვერასოდეს ჩაანაცვლებენ კომპიუტერული ლინგვისტიკის სპეციალისტების შრომას, რადგან ამგვარი ინსტრუმენტები, მათ შორის სტატიაში განხილული Orange Data Mining, ძალიან მარტივია და შორს არის სრულყოფილებისაგან, თუმცა სტატიის მიზანია იმის ჩვენება, რომ თუნდაც დაბალი ხარისხის მანქანური სწავლების გამოყენების შედეგები სულაც არ არის უსარგებლო. ამის საჩვენებლად ჩვენ განვახორციელეთ მანქანური სწავლების ინსტუმენტის Orange Data Minin ტესტირება კავკასიოლოგების მიერ დასმულ სამ საკვლევ საკითხზე, რომელსაც უნდა გამოევლინა, რამდენად ეფექტურია მისი გამოყენება, რათა დაეხმაროს ლინგვისტებს ჰიპოთეზების ჩამოყალიბებაში და წინასწარი შედეგების მიღებაში საკვლევ თემასთან დაკავშირებით, რათა შემდგომში მათი ვერიფიცირება/ფალსიფიცირება უფრო რთული მეთოდებით განხორციელდეს? კერძოდ, ტესტირების შედეგად პასუხი უნდა გაგვეცა სამ კონკრეტულ შეკითხვაზე:

1) შესაძლებელია თუ არა მანქანური სწავლების ინსტუმენტების გამოყენებით რეალური შედეგი მივიღოთ თარგმანმცოდნეობაში მთარგმნელთა იდენტიფიკაციის თვალსაზრისით? კონკრეტულად, ტესტირება განხორციელდა შოთა რუსთაველის „ვეფხისტყაოსნის“ გერმანული თარგმანების ბაზაზე.
2) შესაძლებელია თუ არა მანქანური სწავლების ინსტუმენტების გამოყენებით სხვადასხვა ნაწარმოების იდენტიფიცირება? ტესტირება განხორციელდა „ვეფხისტყაოსნის“, „თამარიანისა“ და „შაჰნამეს“ ტექსტებზე.
3) შესაძლებელია თუ არა ანოტირებულ ტექსტში გრამატიკულ ელემენტთა კლასიფიკაცია ფუნქციონალური კატეგორიების მიხედვით? ტესტირება განხორციელდა ჰინალუღურ ენაში მიმღეობებთან მიმართებით.

სამი ექსპერიმენტიდან თითოეულის შედეგი შეჯამებულია შეცდომის სიხშირის რანგირების სახით. ყველა ექსპერიმენტში წარმოდგენილი იყო მანქანური სწავლების მინიმუმ ერთი მიდგომა 60%-ზე მეტი კლასიფიკაციის წარმატების კოეფიციენტით სამი ან ოთხი განსხვავებული კატეგორიის მიხედვით. შედეგების სიზუსტის დასადგენად ტრენინგი განვახორციელეთ სხვადასხვა მოცულობის მონაცემებზე, რათა დაგვედგინა ტრენინგისათვის აუცილებელ მონაცემთა ოპტიმალური მოცულობა სიზუსტის გაზრდის ან វარდნის გამოსავლენად.

4. შედეგი

ექსპერიმენტის შედეგების ვალიდურობამ დაადასტურა ჩვენი მოლოდინი, რომ ეფექტურია აღნიშნული მეთოდის გამოყენება, განსაკუთრებით წინასწარი კვლევების სფეროში, და ასევე მისაღებია მანქანური სწავლების ტექნოლოგიური ინსტრუმენტის Orange data mining გამოყენება კვლევებში, განსაკუთრებით საფრთხეში მყოფი ან დაბალი რესურსის ენებისთვის. ამგვარი ექსპერიმენტების ჩატარება გამოავლენს თანამედროვე ტექნოლოგიურ შესაძლებლობას, თუ როგორ შეუძლიათ მკვლევრებს წინასწარი ცოდნის მოპოვება მარტივი ინსტრუმენტების გამოყენებით, რომელიც, როგორც ჩანს, უფრო და უფრო გარდაუვალი ხდება დღევანდელ და, განსაკუთრებით, მომავალ კვლევებში.