University of Belgrade

Faculty of Mathematics

Branislava B. Šandrih

IMPACT OF TEXT CLASSIFICATION ON NATURAL LANGUAGE PROCESSING APPLICATIONS

Doctoral Dissertation

Belgrade, 2020

Univerzitet u Beogradu

Matematički fakultet

Branislava B. Šandrih

UTICAJ KLASIFIKACIJE TEKSTA NA PRIMENE U OBRADI PRIRODNIH JEZIKA

doktorska disertacija

Beograd, 2020

Podaci o mentoru i članovima komisije

Mentor:

dr Aleksandar Kartelj, docent
Univerzitet u Beogradu, Matematički fakultet

Članovi komisije:

dr Gordana Pavlović-Lažetić, redovni profesor
Univerzitet u Beogradu, Matematički fakultet

dr Vladimir Filipović, redovni profesor
Univerzitet u Beogradu, Matematički fakultet

dr Cvetana Krstev, redovni profesor
Univerzitet u Beogradu, Filološki fakultet

dr Ruslan Mitkov, redovni profesor
Univerzitet u Vulverhamptonu

Datum odbrane:

____

Dissertation Data

Doctoral dissertation title: IMPACT OF TEXT CLASSIFICATION ON NATURAL LANGUAGE PROCESSING APPLICATIONS

Abstract: The main goal of this dissertation is to put different text classification tasks in the same frame, by mapping the input data into the common vector space of linguistic attributes. Subsequently, several classification problems of great importance for natural language processing are solved by applying the appropriate classification algorithms.

The dissertation deals with the problem of validation of bilingual translation pairs, so that the final goal is to construct a classifier which provides a substitute for human evaluation and which decides whether the pair is a proper translation between the appropriate languages by means of applying a variety of linguistic information and methods.

In dictionaries it is useful to have a sentence that demonstrates use for a particular dictionary entry. This task is called the classification of good dictionary examples. In this thesis, a method is developed which automatically estimates whether an example is good or bad for a specific dictionary entry.

Two cases of short message classification are also discussed in this dissertation. In the first case, classes are the authors of the messages, and the task is to assign each message to its author from that fixed set. This task is called authorship identification. The other observed classification of short messages is called opinion mining, or sentiment analysis. Starting from the assumption that a short message carries a positive or negative attitude about a thing, or is purely informative, classes can be: positive, negative and neutral.

These tasks are of great importance in the field of natural language processing and the proposed solutions are language-independent, based on machine learning methods: support vector machines, decision trees and gradient boosting. For all of these tasks, a demonstration of the effectiveness of the proposed methods is shown on for the Serbian language.

Keywords: natural language processing, machine learning, computational linguistics, text classification, terminology extraction, authorship identification, sentiment classification, classification of good dictionary examples

Scientific field: Computer Science

Scientific subfield: Natural Language Processing

UDC number: 004.85:519.765(043.3)

Podaci o doktorskoj disertaciji

Naslov doktorske disertacije: UTICAJ KLASIFIKACIJE TEKSTA NA PRIMENE U OBRADI PRIRODNIH JEZIKA

Rezime: Osnovni cilj disertacije je stavljanje različitih zadataka klasifikacije teksta u isti okvir, preslikavanjem ulaznih podataka u isti vektorski prostor lingvističkih atributa. Nakon toga se primenom odgovarajućih klasifikacionih algoritama rešavaju neki klasifikacioni problemi koji su od velikog značaja za obradu prirodnih jezika.

Disertacija se bavi problemom validacije prevoda bilingvalnih parova, tako da je krajnji cilj konstruisanje klasifikatora koji pruža zamenu za ljudsku evaluaciju i koji, primenjujući raznovrsne lingvističke informacije i metode, donosi odluku o tome da li par predstavlja prevod između odgovarajućih jezika.

U svakom rečniku, korisno je da uz rečničku odrednicu stoji i primer kako se ona koristi u jeziku. U ovom slučaju reč je o problemu klasifikacije dobrih primera upotrebe. U tezi je razvijan metod koji automatski zaključuje da li je za datu odrednicu primer dobar ili loš.

U disertaciji se razmatraju i dva slučaja klasifikacije kratkih poruka. U prvom slučaju, klase su autori poruka, a zadatak je svakoj poruci dodeliti njenog autora iz tog fiksnog skupa. Ovaj problem naziva se identifikacija autorstva. Drugi razmatrani problem klasifikacije nad kratkim porukama naziva se analiza raspoloženja i stavova, odnosno analiza sentimenata. Ako se krene od pretpostavke da kratka poruka nosi u sebi pozitivan ili negativan stav o nekoj stvari, ali i da može biti isključivo informativna, klase mogu biti: pozitivan, negativan i neutralan stav.

Navedeni zadaci su od velikog značaja u oblasti obrade prirodnih jezika i rešavani su na način koji je nezavisan od jezika, primenom metoda mašinskog učenja: metod podržavajućih vektora, stabla odlučivanja i gradijentnog pojačavanja. Za sve probleme, demonstracija rada predloženih metoda pokazana je na slučaju srpskog jezika.

Ključne reči: obrada prirodnih jezika, mašinsko učenje, računarska lingvistika, klasifikacija teksta, ekstrakcija terminologije, identifikacija autorstva, analiza osećanja, odabir dobrih rečničkih primera

Naučna oblast: Računarstvo

Uža naučna oblast: Obrada prirodnih jezika

UDK broj: 004.85:519.765(043.3)

Contents

1 Introduction

Ever since the first computers emerged, people have been fantasising about having a fully conscious machine. For more than 60 years already, computers that can communicate with humans have been an inspiration for a number of science fiction books and films. One thing is common to all of these fantasies: computers are able to understand humans, answer questions and give advice, but they also follow instructions they are given without any complaints. In some scenarios, a computer becomes smarter than a human who made it – and this may be one of the greatest fears of humanity nowadays.

There are numerous examples showing that humanity needs computers able to understand and generate natural languages:

Content summarisation Before writing a paper, seminar work, thesis etc., one usually has to read numerous texts and pages of references and academic literature. A computer program that can quickly process the text, provide a meaningful summary of its content or even answer questions about is of great practical value in such situations;

Content analysis In order for a company to be able to get feedback from the customers, its staff has to carefully read thousands of emails or reviews, that requires a lot of time and effort. Having a computer that could read, understand, analyse and categorise all received emails/reviews would be very significant;

Speech recognition and speech synthesis It can be very useful to have a device that is able to understand and answer questions, such as asking for directions while the driver's hands are busy, or for people with disabilities;

Text recognition and automatic answer recommendation A world in which humans are not able to get answers from a favourite search engine about a location of a nearest ATM, best restaurants nearby, the cheapest flights on a certain date or even on how to fix a device etc, is hard to imagine nowadays;

Translation No matter whether it is about ordering in a restaurant or following a manual for a newly bought device written in an unknown language, having a computer that can translate from one language to another is very valuable.

In these and many other examples, it can be seen that humans have a need for an assistive artificial intelligence. The goal of the field of "Machine Learning" (ML) is clear from the name itself: how to make machines learn. In order for computers to be able to understand human requests, they first have to be taught how to understand human language – and this is exactly the scope of the field of "Natural Language Processing" (NLP). In order to create intelligent and obedient machines, it is essential to model a human language in a way that machines can understand it. One of the solutions would be to create models that correspond to human brain architecture. Architecture of a *perceptron* machine, invented in 1957 at the Cornell Aeronautical Laboratory by Frank Rosenblatt (Rosenblatt, 1957), was inspired by the organisation of neurons in human brain. It was designed for the image recognition. Yet, the idea was neglected due to the insufficient computational resources. This has changed in the past two decades, due to the advancements of hardware components. With the advance of computational power and available memory resources, Artificial Neural Networks (ANNs) rapidly gained popularity in practical applications. Despite already long-lasting efforts in developing ANNs, human brain is still far too complex to model.

In order to create intelligent computers that can understand and respond to human languages, NLP copes with many different challenges. What are humans consciously or subconsciously doing during the process of language understanding? First, they know the meaning of words. In case of unknown word, humans consider the context. Obtaining an universal lexicon, even for a single language, is not feasible. Each language has finite but unlimited number of words;[1] another reason is that natural language is changing, and new words appear all the time, while some other words disappear from a language. Even if it were possible to have a lexicon that contains all words belonging to a language, understanding of the meaning would still often be questionable. One of the reasons for this is a homonymy: having same words with different meanings. For example, a surface can be *flat*, but a family can live in a *flat*, as well. In that case, humans subconsciously take the context into consideration. There is a famous sentence by Firth (1957), quote: "You shall know the word by the company it keeps". Presence of phraseological units is another reason: if, for example, the idiom "it is raining cats and dogs" were interpreted by its literal meaning, completely wrong conclusions would be drawn. Here lies the complexity of a natural language. In order to model a natural language perfectly, the mathematical model would have to take into consideration all of these aspects of complexity.

As Jurafsky and Martin (2019) point out, engaging in complex language behaviour requires various kinds of knowledge of language:

- Phonetics and Phonology – knowledge about linguistic sounds;
- Morphology – knowledge of the meaningful components of words;
- Syntax – knowledge of the structural relationships between words;
- Semantics – knowledge of words meaning;
- Pragmatics – knowledge of the relationship of words meaning to the goals and intentions of the speaker;
- Discourse – knowledge about linguistic units larger than a single utterance.

[1] This can be justified on the example of adjective comparison: one can be smarter; but in jargon, another one can be smarterer, etc.

Another important field that addresses the problem of teaching linguistic knowledge to computers is Computational Linguistics (CL). It involves looking at the nature of a language, its morphology, syntax, and dynamic use, and drawing any possible useful models from this observation in order to help machines to process language. According to the 2012 Association for Computing Machinery (ACM) Computing Classification System,[2] NLP is a direct sub-field of the artificial intelligence. The ACM does not classify CL as the sub-field of the Computer Science (CS). Yet, the Association for Computational Linguistics (ACL) defines CL as a scientific study of language from a computational perspective. Efforts of computational linguists are aimed at providing computational models of various kinds of linguistic phenomena.[3] Simply put, the difference is that CL tends more towards linguistics, and answers linguistic questions using computational tools. NLP develops applications that process a language and is inclined more towards CS. NLP has more applied nature than CL which in turn, has more to do with theories.

One of most popular NLP applications is Text Classification (TC). The classification task, in general, is an old and every-day problem. People perform classification all the time: classifying human beings according to their gender, animals according to their species, cars according to their price, etc. The principle of Text Classification does not differ from the general classification formulation: assignment of predefined categories to a certain object. This thesis focuses on four specific cases of text classification.

1.1 Text Classification

Text Classification uses dataset:

$$\mathcal{D} = \{(\mathbf{X_1}, y_1), (\mathbf{X_2}, y_2), \ldots, (\mathbf{X_N}, y_N)\}$$

which consists of N samples $\mathcal{X} = \{\mathbf{X_1}, \mathbf{X_2}, \ldots, \mathbf{X_N}\}$.

If the aim is to assign each sample to one class, this is called *classification*. Namely, the goal is to approximate a mapping function

$$c : \mathcal{X} \longrightarrow \mathcal{C}$$

where $\mathcal{X}$ is a collection of samples and $\mathcal{C}$ is a set of distinct class labels.

The process of assigning an n-dimensional vector $(x_1, x_2, \ldots, x_n)$ to any object $\mathbf{X} \in \mathcal{X}$ is called *mapping to a feature space*. Each vector dimension is called *a feature* or *an attribute*. The motivation behind this is to obtain a mathematical representation of an object, regardless of its nature. For example, depending on the task, a person can be represented by gender, height, country of birth, annual income, level of education etc. Features with discrete values are called *categorical*, while the ones with continuous values are *numerical* features.

For example, let the objects that need to be classified according to the opinion they contain, be the following sentences:

POS_1 The food was great and the staff was kind ... great hotel!;

[2] ACM Computing Classification System, https://dl.acm.org/ccs/ccs_flat.cfm

[3] The ACL, www.aclweb.org/archive/misc/what.html

POS_2 I would recommend this great hotel to everyone;

POS_3 It was great staying in this great place with great service!;

NEG_1 The food was terrible!;

NEG_2 Terrible staff, dirty rooms . . . all in all, terrible;

NEG_3 It was terrible staying in this terrible hotel with terrible hygiene;

NEU_1 The hotel is OK;

NEU_2 Not great, but also not terrible;

The first two sentences carry a positive, and the other two sentences contain a negative opinion. Firstly, these sentences should be represented as feature vectors. Let the feature space contain the following components:

- times *great* occurred in the sentence;
- times *kind* occurred in the sentence;
- times *terrible* occurred in the sentence;
- times *dirty* occurred in the sentence;

Then the above sentences can be represented as feature vectors given in Table 1.1.

TABLE 1.1: Sentences represented as feature vectors

	nr_great	nr_kind	nr_terrible	nr_dirty
POS_1	2	1	0	0
POS_2	1	0	0	0
POS_3	3	1	0	0
NEG_1	0	0	1	0
NEG_2	0	0	2	1
NEG_3	0	0	3	0
NEU_1	0	0	0	0
NEU_1	1	0	1	0

Now, each of these sentences is mapped into a 4-dimensional space. The next optional step is to perform feature selection, i.e. reduce number of variables in the data by selecting the most important ones. If, for example, nr_great and nr_terrible dimensions are determined as the most important ones, then these sentences can be represented in a 2-dimensional plane, as shown in Figure 1.1.

As it can be seen from Figure 1.1, sentences form three clusters. Task presented to the classifier is the following: how to separate these samples adequately, estimating the separating hyper-planes, so that, when new samples (sentences) are introduced and mapped into the same feature space, the containing opinion is also well predicted?

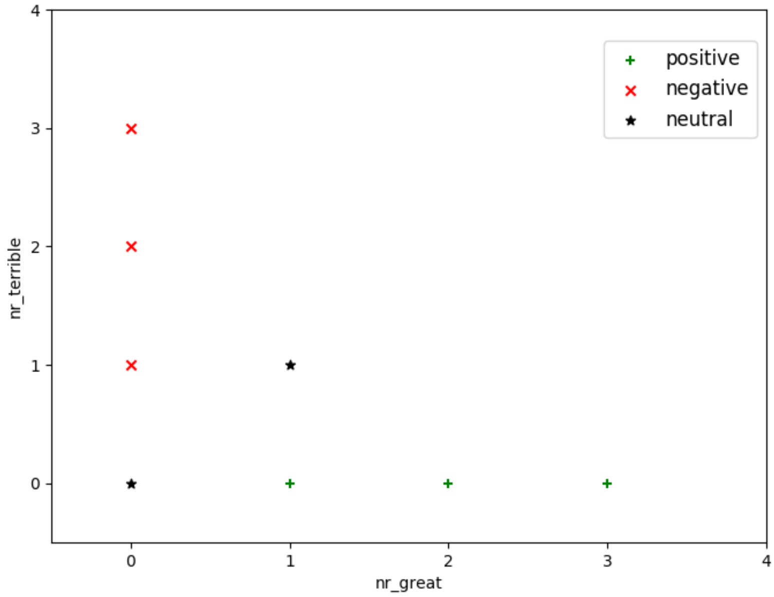


FIGURE 1.1: Sentences from Table 1.1, projected on nr_great and nr_terrible

In the most common scenario, the classes are taken to be disjoint, so that each input is assigned to one and only one class. That way, the input space is divided into decision regions whose boundaries are called *decision boundaries* or *decision surfaces*. As in (Bishop, 2006), for simplification and without a loss of generality, linear models are considered for classification in this section. This means that the *decision surfaces* are linear functions of the input vector $\mathbf{X}$ and hence are defined by $(n-1)$-dimensional hyper-planes within the n-dimensional input space. Data samples belonging to different classes that can be perfectly separated by linear decision surfaces are said to be *linearly separable*. Following is a general definition:

Definition 1. Let $\mathbf{D} = \{(\mathbf{X_1}, y_1), (\mathbf{X_2}, y_2), \ldots, (\mathbf{X_N}, y_N)\}$ be a dataset comprising of N ordered pairs of feature vectors $\mathbf{X_i} \in \mathbb{R}^n$ and their corresponding class labels $y_i \in \mathcal{C} = \{c_1, c_2, \ldots, c_K\}$, $i \in \{1, 2, \ldots, N\}$, where $\mathbb{R}$ is a set of real numbers, n is a vector's dimension (i.e. a number of features used for representation), and $\mathcal{C}$ is a set of K classes for the task. The classification task represents approximation of an intermediate function $f : \mathbb{R}^n \longrightarrow \mathbb{R}, y(\mathbf{X}) = f(\mathbf{w}^T\mathbf{X} + w_0)$, where $\mathbf{w}^T$ is a *transposed weight vector*, and w_0 is a bias, which is later used to construct *decision function* $c : \mathbb{R}^n \longrightarrow \mathcal{C}$, that maps an arbitrary feature vector $\mathbf{X} \in \mathbb{R}^n$ to its class label $c(\mathbf{X}) = \hat{y}$, so that the predicted class equals to the exact one, namely $\hat{y} = y$.

In the case of linear models, discriminant linear models are employed for classification in this thesis. For such models, the representation of a linear discriminant function $f(\cdot)$ can be determined by taking a linear function of the input vector so that $y(\mathbf{X}) = \mathbf{w}^T\mathbf{X} + w_0$.

A special case is the case of *binary classification*, where $K = 2$. These class labels can be encoded as $\mathcal{C} = \{1, -1\}$. Class encoded with 1 is called the *positive class*, while the other one is referred to as the *negative class*.

An input vector $\mathbf{X}$ is assigned to the positive class if $y(\mathbf{X}) \geq 0$ and to the negative class otherwise. The corresponding decision boundary is therefore defined by the relation $y(\mathbf{X}) = 0$, which corresponds to a $(n-1)$-dimensional hyper-plane within the n-dimensional input space.

Consider two points $\mathbf{X_A}$ and $\mathbf{X_B}$ both of which lie on the decision boundary. Because $y(\mathbf{X_A}) = y(\mathbf{X_B}) = 0$, we have $\mathbf{w}^T\mathbf{X_A} + w_0 = \mathbf{w}^T\mathbf{X_B} + w_0$, that is $\mathbf{w}^T(\mathbf{X_A} - \mathbf{X_B}) = 0$, and hence the vector $\mathbf{w}$ is orthogonal to every vector lying within the decision boundary, and so $\mathbf{w}$ determines the orientation of the decision boundary. Similarly, if $\mathbf{X}$ is a point on the decision surface, then $y(\mathbf{X}) = 0$, and so the normal distance from the origin to the decision boundary is given by $\frac{\mathbf{w}^T\mathbf{X}}{||\mathbf{X}||} = -\frac{w_0}{||\mathbf{w}||}$. It can therefore be seen that the bias parameter w_0 determines the location of the decision boundary.

Any binary classifier can be generalised to a multi-class classifier, by decomposing the prediction into multiple binary decisions. Following are the common techniques for the extension of linear discriminators to $K > 2$ classes.

One-versus-the-rest This technique uses $K-1$ classifiers each of which solves a two-class problem of separating points in a particular class $c_i \in \mathbf{C}$ from points not in that class. This approach may lead to regions of input space that are ambiguously classified.

One-vs-one An alternative is to introduce $K(K-1)/2$ binary discriminant functions, one for every possible pair of classes. This is known as a *one-versus-one* classifier. Each point is then classified according to the majority vote among the discriminant functions. However, this also runs into the problem of ambiguous regions.

Often, machine learning algorithms have to minimise a loss function. The loss function estimates how good a prediction model performs in terms of being able to predict the expected outcome. For the sake of minimisation, the most commonly used method of finding the minimum point of function is a gradient descent (Bottou, 2010).

One of the most commonly used loss functions is Mean Square Error (MSE), which is measured as the average of squared difference between predicted value $\hat{y}_i$ and actual value y_i, for $i = 1 \dots N$ (Equation 1.1).

$$MSE = \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{N} \tag{1.1}$$

1.1.1 Classification Methods

In this subsection, descriptions of the classification methods applied later in the thesis are given.

Naïve Bayes

A Naïve Bayes classifier is a statistical machine learning model that is used for classification tasks. Nowadays it provides a "baseline" for evaluating other learning algorithms. Essentially, the classifier is based on the Bayes theorem

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)} \tag{1.2}$$

where $P(h)$ is a prior probability of hypothesis h, $P(D)$ is a prior probability of training data D, $P(h|D)$ is a probability of h given D, and $P(D|h)$ is a probability of D given h.

Generally, an aim is to determine the most probable hypothesis given the training data. This is done using the *Maximum A Posteriori* (MAP) hypothesis h_{MAP}:

$$h_{MAP} = \arg\max_{h \in H} P(h|D) = \arg\max_{h \in H} \frac{P(D|h)P(h)}{P(D)} \propto \arg\max_{h \in H} P(D|h)P(h) \tag{1.3}$$

where $\propto$ denotes proportionality.

The goal of the NB classifier is to estimate a probability of a sample $\mathbf{X}$ occurring in each of the classes, i.e:

$$p(c_i|\mathbf{X}) = p(c_i|x_1, x_2, \ldots, x_n) \tag{1.4}$$

for each hypothesis c_i, $i = 1, \ldots, K$.

After replacing h and D from the Equation 1.2 with c_i and $\mathbf{X}$, respectively, the Equation 1.4 reads $p(c_i|x_1, x_2, \ldots, x_n) = \frac{p(x_1, x_2, \ldots, x_n|c_i)p(c_i)}{p(x_1, x_2, \ldots, x_n)}$. According to the simplification given in Equation 1.3, $p(c_i|x_1, x_2, \ldots, x_n)$ can be estimated as $p(c_i|x_1, x_2, \ldots, x_n) \propto p(x_1, x_2, \ldots, x_n|c_i)p(c_i)$.

The probability $p(c_i)$ is usually estimated as the ratio of number of samples in the labelled dataset belonging to the class c_i, to the total number of samples. In order to compute $p(x_1, x_2, \ldots, x_n|c_i)$, next assumption has to be applicable: $x_1, x_2, \ldots x_n$ are conditionally independent given a class c_i. If each feature x_i is conditionally independent of every other feature x_j, for $j \neq i$, given the class label c_k, then $p(x_i|x_{i+1}, \ldots, x_n, c_k) = p(x_i|c_K)$ holds. The probability $p(x_1, x_2, \ldots, x_n|c_i)$ can be therefore determined as $p(x_1, x_2, \ldots, x_n|c_i) = p(x_1|c_i)p(x_2|c_i) \ldots p(x_n|c_i)$, finally, yielding $p(c_i|x_1, x_2, \ldots, x_n) \propto p(x_1|c_i)p(x_2|c_i) \ldots p(x_n|c_i)p(c_i), = p(c_i) \prod_{j=1}^{n} p(x_j|c_i)$.

Estimation of $p(x_j|c_i)$ depends on the task given. Generally, it is a normalised count of samples to the total number of samples having the j^{th} feature with value x_j that belong to the class c_i. After determining $p(\mathbf{X}|c_i)$, for each $i = 1, \ldots, K$, a class label is assigned to the sample $\mathbf{X}$ using MAP hypothesis given in Equation 1.3: $y = \arg\max_{c_i} p(c_i) \prod_{j=1}^{n} p(x_j|c_i)$.

Logit Classification

Logistic Regression (LR) is part of the category of statistical models called generalised linear models. An overview of generalised linear models is given in (Agresti, 1996).

A LR algorithm uses a linear equation with independent variables (predictors) to predict a value. The predicted value can take values between $(-\infty, +\infty)$, i.e. any value from the set of real numbers $\mathbb{R}$. The output of a classifier is a value from a discrete set of values. Therefore, the first step is to map the output of the linear equation into $(0, 1)$ (which are later interpreted as probabilities).

LR is named after the *logistic function* used at the core of the method. Also known as the *sigmoid function*, it takes a real-valued argument and maps it into $(0, 1)$ (see Figure 1.2), namely:

$$f(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{1 + e^x}$$

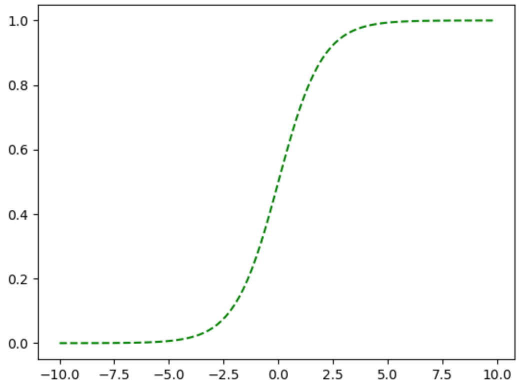


FIGURE 1.2: The Sigmoid function

LR uses an equation as the model representation. This makes it interpretable, similarly as is the case with the NB classifier. The goal is to estimate an equation:

$$f(X) = f(x_1, x_2, \ldots, x_n) = \frac{e^{(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots \beta_n x_n)}}{1 + e^{(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots \beta_n x_n)}} \tag{1.5}$$

where β_0 is *the bias* or *the intercept term*, and β_i are weights or coefficients for each feature estimated from the training samples. Since this function takes values between 0 and 1, it is not wrong to assume that the LR predicts *probability* for its argument. Put precisely, the equation

$$p(y = 1|X) = \frac{e^{(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots \beta_n x_n)}}{1 + e^{(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots \beta_n x_n)}} \tag{1.6}$$

estimates the probability of sample X belonging to the positive (or default) class. Formulated like this, LR model can only be applied to binary classification tasks. The probability of the negative class is determined as $p(y = -1|X) = 1 - p(y = 1|X)$.

The generalisation of the LR model is called *Maximum Entropy*, as described thoroughly by Bishop (2006). The class label is predicted using MAP rule, as Friedman, Hastie, and Tibshirani (2001) explained.

k-Nearest Neighbours

The predicted class label $\hat{y}$ for feature vector $\mathbf{X}$ is the most common value of the class labels y among the k training examples nearest to X, as described in (Bishop, 2006).

The nearest neighbours of a feature vector $\mathbf{X} \in \mathbb{R}^n$ are defined in terms of the standard Euclidean distance. More precisely, the distance between two feature vectors $\mathbf{X_i} = (x_1^i, x_2^i, \dots, x_n^i), \mathbf{X_j} = (x_1^j, x_2^j, \dots, x_n^j)$ is defined to be $d(\mathbf{X_i}, \mathbf{X_j})$, where $d(\mathbf{X_i}, \mathbf{X_j}) = \sqrt{\sum_{r=1}^n (x_r^i - x_r^j)^2}$.

For different values of k, the most common value of y among the k-nearest training examples is assigned to a sample. This means that the choice of parameter k has a strong influence on the outcome of the classification, which is demonstrated in Figure 1.3. For $k = 3$, the assigned class label is the one represented using rectangles, and for $k = 7$, the predicted class changes to the one represented using circles.

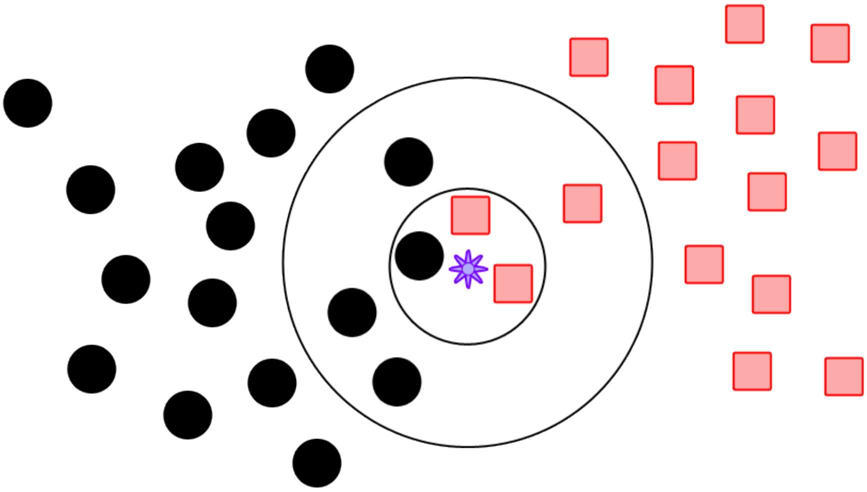


FIGURE 1.3: k-Nearest Neighbours

Decision Trees

Decision tree (DT) classifiers can be used to visually and explicitly represent decisions and the decision making process. A condition based on which the DT splits into branches (edges) is contained in internal nodes. The end of the branch that does not split anymore is the final decision (terminal node, i.e. leaf). In the case of classification, terminal nodes contain class labels. A tree grows by deciding which features to choose and what conditions to use for splitting, along with knowing when to stop. A common technique used for splitting is *Recursive Partitioning*. In this procedure all features are considered and different split points are tried and tested using a *cost function*. This function represents a single, overall measure of loss as a consequence of taking any of the available decisions or actions. The goal is then to minimise the total loss incurred so the split with the lowest cost is selected. For example, a cost function can be defined as a classification error (e.g. percent of samples inaccurately classified). Another cost function is *a Gini function*, defined as: $G = \sum_{j=1}^{n}(p_j(1 - p_j))$. A Gini measures how good a split is by determining how "pure" the response classes are, i.e. whether they contain predominately inputs from the same class. Here, p_j is proportion of the same class inputs present in a particular group. A perfect *class purity* occurs when a group contains all inputs from the same class, in which case p_j is either 1 or 0 and $G = 0$, whereas a node having a $50 - 50$ split of classes in a group has the worst purity, so for a binary classification, it will have p_j and G both equal 0.5.

In respect to the recursive nature of the algorithm, the formed groups can be sub-divided using the same strategy. Due to this procedure, this algorithm is also known as the *greedy algorithm*, since there is an excessive desire of lowering the cost. If the number of splits are not controlled, it can lead to over-fitting to the training data. This can be avoided by setting a minimum number of training inputs to use on each leaf, or by setting maximum depth of the tree model (i.e. to restrict the length of the longest path from a root to a leaf).

Pruning allows further improvement of the performance. This way, the complexity of the tree is reduced, and thus its predictive power is increased by reducing over-fitting. Pruning can start at either root or the leaves. The simplest pruning technique starts at the leaves and removes each node with the most popular contained class, as long as it does not degrade accuracy. It is also called the *Reduced Error Pruning*. Decision tree learning can be improved by using techniques of *bootstrapping*, *bagging* and *boosting*. More details about DTs can be found in (Michie, Spiegelhalter, and Taylor, 1999; Mitchell, 1997; Friedman, Hastie, and Tibshirani, 2001; Bishop, 2006; Segaran, 2007).

Gradient Boosting

Boosting refers to a group of algorithms that utilise weighted averages of the previous iterations, in order to make weak learners become stronger learners. Examples of such classifiers are *AdaBoost* and *Gradient Boosting* (GB). Gradient Boosting is a sequential technique that combines a set of Decision Trees and yields improved prediction accuracy. Trees are added one at a time. When adding new trees, gradient descent procedure is used to minimise the loss. After calculating error or loss, the outcomes predicted correctly are given a lower weight and the ones miss-classified are weighted higher. This is repeated until optimal instance weights are found.

As mentioned earlier, machine learning algorithms tend to define a loss function and minimise it. In case of GB algorithm, Mean Square Error (MSE) is selected as a loss function, given in Equation 1.1. Let y_i be an actual class label of the i^{th} sample, $\mathbf{X_i}$, and $\hat{y}_i$ the predicted class label, for $i = 1 \ldots N$, where N represents the number of samples in a dataset.

Applying gradient descent, we estimate the gradient vector of $\mathbf{\hat{y}_i} = \mathbf{\hat{y}_i} + \alpha \cdot \frac{\delta \sum_{i=1}^{N}(\mathbf{y_i} - \mathbf{\hat{y}_i})^2}{\delta \mathbf{\hat{y}_i}}$, which equals $\mathbf{\hat{y}_i} = \mathbf{\hat{y}_i} - \alpha \cdot 2 \cdot \sum_{i=1}^{N}(\mathbf{y_i} - \mathbf{\hat{y}_i})$, where α is a *learning rate* and $\sum_{i=1}^{N}(\mathbf{y_i} - \mathbf{\hat{y}_i})$ is a sum of *residuals*. Learning rate is a parameter of a gradient descent method that dictates how fast the gradient vector changes in each iteration, while the residuals represent the difference between the predicted and the actual values. An aim is to continue updating the predictions, so that the sum of the residuals is close to 0 (or minimum) and predicted values are sufficiently close to actual values.

Support Vector Machines

With Support Vector Machines, classification is performed by finding the plane in high-dimensional space that separates samples from different classes with the highest possible margin, as displayed in Figure 1.4. In the case when samples are linearly separable, i.e. where it is possible to find a hyper-plane that physically separates training samples belonging to one class from the training samples belonging to one or more other classes, SVM is linear. If samples are not linearly separable, kernel trick should be applied. This means mapping all samples into another, higher-dimensional space, where the separating hyper-plane can be determined.

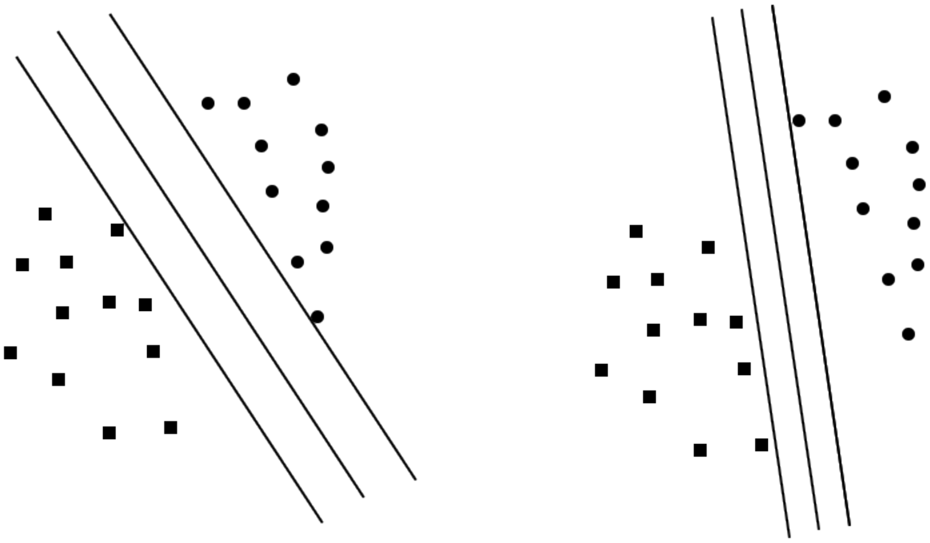


FIGURE 1.4: Hyper-planes with different margins

Definition 2. Let $\mathbf{S_0}$ and $\mathbf{S_1}$ be two sets of points in an n-dimensional Euclidean space. Then $\mathbf{S_0}$ and $\mathbf{S_1}$ are linearly separable if there exist $n + 1$ real numbers $w_1, w_2, .., w_n, b$ such that every point $\mathbf{X_0} \in \mathbf{S_0}$ satisfies

$$\sum_{i=1}^{n} w_i x_i^0 > b$$

and every point $\mathbf{X_1} \in \mathbf{S_1}$ satisfies

$$\sum_{i=1}^{n} w_i x_i^1 < b$$

where x_i^k is the i-th component of $\mathbf{X_k}$.

In the following text, the problem of binary classification is considered. As mentioned earlier, any problem of binary classification can be generalised to a multi-class classification. Let $\mathcal{D} = (\mathbf{X_1}, y_1), (\mathbf{X_2}, y_2), \ldots, (\mathbf{X_N}, y_N)$ be a dataset of N feature vectors $\mathbf{X_i} \in \mathbb{R}^n, i \in 1 \ldots N$, where n is the number of features, and their corresponding class labels $y \in \{-1, 1\}, i \in 1 \ldots N$.

Any hyper-plane in an n-dimensional space can be written as: $w_1 x_1 + w_2 x_2 + \ldots + w_n x_n - b = 0$. If the weights are written as a weight vector $\mathbf{w} = (w_1, w_2, \ldots, w_n)$, then this equation can be rewritten as $\mathbf{w} \cdot \mathbf{X} - b = 0$. This vector $\mathbf{w}$ is orthogonal to the hyper-plane. According to the estimated hyper-plane (i.e. the classification function), the decision function is estimated as $c(\mathbf{X}) = sgn(\mathbf{w} \cdot \mathbf{X} - b)$, where sgn is a sign function.

If the training data are linearly separable, two parallel hyper-planes separating the two classes of data can be selected. The selection criterion is that the distance between them is as large as possible. The region bounded by these two hyper-planes is called the *margin*, and the *maximum-margin hyper-plane* is the surface that lies halfway between them. These hyper-planes can be described by the equations $\mathbf{w} \cdot \mathbf{X} - b = 1$ (anything on or above this boundary is of one class, with label 1) and $\mathbf{w} \cdot \mathbf{X} - b = -1$ (anything on or below this boundary is of the other class, with label -1).

Geometrically, the distance between these two hyper-planes is $\frac{2}{\|\mathbf{w}\|}$, so to maximise the distance between the planes, the goal is to minimise $\|\mathbf{w}\|$. The distance is computed using the equation that determines distance from a point to a plane. Data points also have to be prevented from falling into the margin, so the following constraint is added: for each i either $\mathbf{w} \cdot \mathbf{X_i} - b \geq 1$, if $y_i = 1$ or $\mathbf{w} \cdot \mathbf{X_i} - b \leq -1$, if $y_i = -1$ applies.

These constraints state that each data point must lie on the correct side of the margin. This is called the *hard-margin SVM*, and it can be rewritten as

$$y_i(\mathbf{w} \cdot \mathbf{X}_i - b) \geq 1, \quad \text{for all } 1 \leq i \leq N.$$

Put this together to get the optimisation problem

$$\min \|\mathbf{w}\|$$

subject to:

$$y_i(\mathbf{w} \cdot \mathbf{X}_i - b) \geq 1 \text{ for } i = 1, \ldots, N.$$

An important consequence of this geometric description is that the max-margin hyper-plane is completely determined by those $\mathbf{X_i}$ vectors that lie nearest to it. These $\mathbf{X_i}$ vectors are called the *support vectors*.

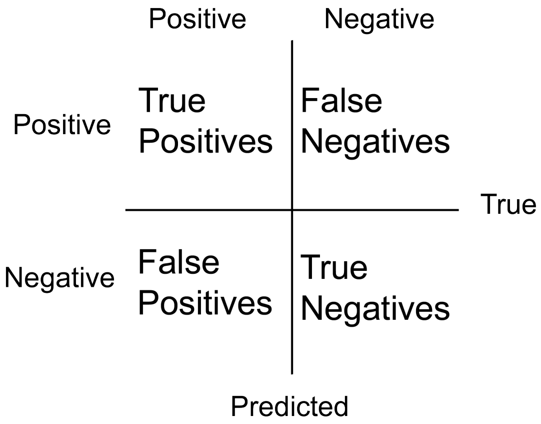
One way to overcome the problem of linearly inseparable vectors is to allow certain error. This is called *soft margin* classification (Cortes and Vapnik, 1995). Boser, Guyon, and Vapnik (1992) suggested a way to create a nonlinear SVM classifier by applying the *kernel trick* to maximum-margin hyper-planes. The resulting algorithm differs in the way that every dot product is replaced by a *nonlinear kernel function*. This allows the algorithm to fit the maximum-margin hyper-plane in a transformed feature space. This kernel trick applies when training samples are not linearly separable in the original feature space: they are then mapped into higher-order feature space, where they are linearly separable.

1.1.2 Evaluation

After mapping samples to a feature space, implementing a model and getting some output in forms of a probability or a class, the next step is to find out how effective the model is based on some metric using a test set. Different performance metrics, originating from Information Retrieval, are used to evaluate different ML Algorithms. The metric chosen for evaluation influences on how the performance of ML algorithms is measured and compared. In the following text, some evaluation metrics for the classification tasks are listed. More about these evaluation metrics can be found in (Fawcett, 2006; Powers, 2011; Sammut and Webb, 2017).

A Confusion Matrix (CM) is a common metric used for finding the correctness and accuracy of the model. It is used either for binary or for multi-class classification tasks. Next terms are associated with CM: True Positives (abbrev. TP) represent the number of cases when the predicted class is a positive class, and the actual class of these samples is also positive, True Negatives (abbrev. TN) represent the number of cases when the predicted class is a negative class, and the actual class of these samples is also negative, False Positives (abbrev. FP) represent the number of cases when the predicted class is a positive class, and the actual class of these samples is negative, False Negatives (abbrev. FN) represent the number of cases when the predicted class is a negative class, and the actual class of these samples is positive. These terms are visualised in Figure 1.5 for a binary classification task.

Based on TP, FP, TN and FN, among others, measures from the Table 1.2 can be determined.



		Predicted	
		Positive	Negative
True	Positive	True Positives	False Negatives
	Negative	False Positives	True Negatives

FIGURE 1.5: Confusion Matrix

TABLE 1.2: Evaluation metrics

Metric	Equation	Description
Precision	$\frac{TP}{TP+FP}$	Ratio of correctly predicted samples to a total number of samples predicted to belong to a certain class
Recall	$\frac{TP}{TP+FN}$	Ratio of samples which are predicted to belong to the certain class to all samples that actually belong to that class
Accuracy	$\frac{TP+TN}{TP+FP+TN+FN}$	Ratio of well predicted samples to the total number of samples
F_1	$\frac{1}{\frac{1}{P}+\frac{1}{R}}$	Harmonic mean of precision and recall

In order to evaluate a model's performance, i.e. to predict how well it will generalise to an unseen data set, there are several evaluation approaches. The most commonly used ones are:

Holdout Method A part of the dataset is removed and used afterwards to get predictions from the model trained on the rest of the data, i.e. on *the training set*. The error estimation then tells how the model is doing on unseen data i.e. on *the test set*. This method suffers from a high variance, because of the uncertainty of which samples will end up in the test set and the result might be entirely different for different selection of sets. The diagram is displayed in Figure 1.6.

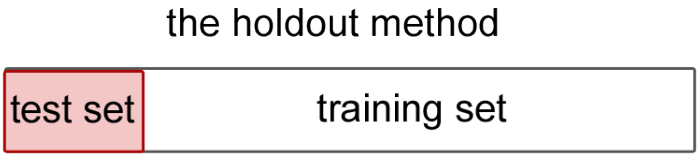


FIGURE 1.6: Splitting dataset into a training and a test set

***k*-Fold Cross Validation** The dataset is firstly sub-divided into k subsets. Then the holdout method is repeatedly applied k times, so that each time, one of the k obtained subsets is used as a test set, and the other $k - 1$ subsets are put together to form a training set. The overall error is finally estimated as an average value of the k error values. Therefore, every sample gets to be in a validation set exactly once, and gets to be in a training set $k - 1$ times, which reduces bias since all data are used for fitting. It also significantly reduces variance, as all data are also being used in the test set. The scheme for $k = 5$ is depicted in Figure 1.7.

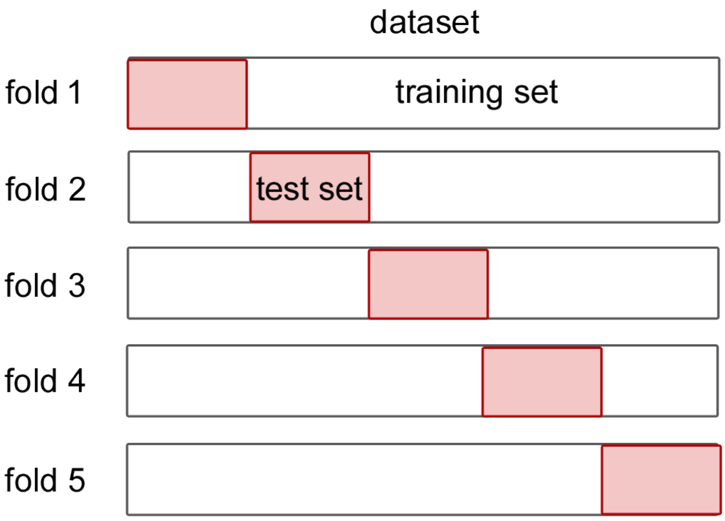


FIGURE 1.7: A 5-fold Cross Validation

1.2 Natural Language Processing

Natural Language Processing is an interdisciplinary field concerned with the processing of human languages by computers. Terms that are often used interchangeable, from theoretical to more application-orientated are: Computational Linguistics (CL), Natural Language Engineering, (Human) Language Technology and Speech and Language Processing.

Computational approaches to language processing are aimed at automating the analysis of the linguistic structure of language and developing applications such as machine translation, speech recognition, and speech synthesis. These models may be “knowledge-based” (“hand-crafted”) or “data-driven” (“statistical” or “empirical”). Work in CL is in some cases motivated from a scientific perspective where one is trying to provide a computational explanation for a particular linguistic or psycho-linguistic phenomenon; and in other cases, the motivation may be more technological, where the goal is to develop a practical component of a speech or natural language system (ACL, 2005).

During the first several decades of work in CL, scientists attempted to write down the vocabularies and rules of human languages for computers. The problems of variability, ambiguity, and context-dependent interpretation of human languages (Hirschberg and Manning, 2015) is nowadays usually modeled using ANNs. This means that today scientists attempt to replicate human way of learning a language. The following subsections briefly describe common tasks related to computational language processing and understanding.

1.2.1 Text Pre-Processing

There are many NLP applications in practical scenarios. For many of them the initial step of *data pre-processing* is needed. This step includes several tasks.

Tokenisation

Electronic text is a linear sequence of symbols (characters or words or phrases). Before any text processing, the input text needs to be segmented into linguistic units such as words, numbers, sentences, punctuation, etc.

Different text segments, depending on their semantic or syntactic role, are (Trost, 2005; Manning, Raghavan, and Schütze, 2008; Mikheev, 2021):

- *A morpheme* is a meaningful morphological unit of a language that cannot be further divided (e.g. *out*, *reach*, *-ing*, forming outreaching).
- *A token* is a sequence of characters that are grouped together as a useful semantic unit for processing.
- *A type* is a class of all tokens that consist of the same character sequence.

Given a character sequence, *tokenisation* is the task of breaking up a given text into tokens. Following is an example of tokenisation:
Example 1.2.1. The original sentence:[4]

It’s a beautiful day, don’t let it get away!

The sentence after tokenisation:

$$\boxed{\text{It's}}\ \boxed{\text{a}}\ \boxed{\text{beautiful}}\ \boxed{\text{day}}\ \boxed{\text{,}}\ \boxed{\text{don't}}\ \boxed{\text{let}}\ \boxed{\text{it}}\ \boxed{\text{get}}\ \boxed{\text{away}}\ \boxed{\text{!}}$$

[4] This sentences is from a U2 band’s song “Beautiful day”.

Challenges in tokenisation depend on the type of language (Manning, Raghavan, and Schütze, 2008). Languages such as English and Spanish belong to the group of *space-delimited* languages, as most of the words are separated from each other by spaces. On the other hand, Thai and Chinese are referred to as *unsegmented* as words do not have clear boundaries. Tokenising of un-segmented language sentences requires additional lexical and morphological information. The writing system and the typographical structure of words also affect tokenisation.

Noisy Entities Removal

In some cases, such as categorisation of a textual document according to its content, any piece of text which is not relevant to the content of the data and the end-output can be specified as the noise. *A stop-word* is a word that has significant syntactic value in sentence formation, but carries minimal semantic value. Such are words in English like: a, an, and, are, as, at, be, by, it, its, of, on, the, to, etc. A negative effect of these words' presence is that they extend the total vocabulary of words that are present in a certain text, but are not the key to the content understanding. For example, URLs or links, social media entities (mentions, hashtags) and punctuation can also be considered noise in the case of document categorisation.

The most simple approach for noise removal is to prepare in advance a dictionary of noisy entities, for example, in the case of stop-words. Similarly, a useful technique is to apply regular expressions for noisy entities such as URLs, punctuation etc. Another method is to assign importance to words. All these approaches origin from Information Retrieval. A common technique for weighing importance of words present in a given text is by using the *TF-IDF* measure. Stop-words are the ones with the least importance. This measure is, as the name suggests, a product of two scores: Term Frequency (TF) and Inverse Document Frequency (IDF) (Salton and Buckley, 1988). The importance of a word is normally determined in relation to a collection of textual documents $\mathcal{D}$. Here, *terms* usually represent tokens obtained after previously performed tokenisation and optional preprocessing steps, such as lemmatisation, stemming and punctuation elimination, which are explained later in the text.

Term Frequency (TF) For a given term t in a textual document D, TF is simply a ratio of number of occurrences of the term n_t in the document, to the total number of terms present in the document $|D|$: $tf_t = \frac{n_t}{|D|}$.

Inverse Document Frequency (IDF) Let the number of documents in the collection $\mathcal{D}$ that contain a term t be denoted as df_t. IDF is determined as: $idf_t = \log \frac{|\mathcal{D}|}{df_t}$.

Where $|\mathcal{D}|$ represents the number of documents in a collection.

TF-IDF weighting The TF-IDF weighting scheme assigns a weight to term t in document D given by:

$$tfidf_t = tf_t \cdot idf_t = \frac{n_t}{|D|} \cdot \log \frac{|\mathcal{D}|}{df_t} \tag{1.7}$$

Overall, this measure assigns a weight to a term t in a document D that is:

- the highest when the term occurs many times within a small number of documents (which enables high discriminating power to those documents);
- lower when the the term occurs fewer times in a document, or occurs in many documents;
- the lowest when the term occurs in almost all documents (which means that the term is not discriminatory for any document in particular).

Hence, it yields values close to 0 for stop-words, making it suitable for the step of dropping.

Lexicon Normalisation

It is common that an arbitrary text contains different forms of the same word, e.g. *play*, *played*, *plays*, *playing*, etc. It is often useful to *normalise* a given text by reducing inflectional forms or derivationally related forms of a word to a common base form. Morphology is the study of the way words are built up from smaller meaning-bearing units, *morphemes*. For example, word *language* consists of a single morpheme (the morpheme language) while the word *languages* consists of two: the morpheme *language* and the morpheme *-s*. As this example suggests, it is often useful to distinguish two broad classes of morphemes: *stems* and *affixes*. The exact details of the distinction vary from language to language. Generally, the stem is the "main" morpheme of the word, which supplies the main meaning. Intuitively, the affixes add "additional" meanings to stems.

Affixes are further divided into *prefixes, suffixes, infixes*, and *circumfixes*. Prefixes precede the stem, suffixes follow the stem, circumfixes do both, and infixes are inserted inside the stem.

Stemming usually refers to a crude heuristic process that chops off the beginnings and ends of words, and often includes the removal of derivational affixes. An example is given in Table 1.3.

TABLE 1.3: An example of words and their stems

Form	Suffix	Stem
flies	-es	fli
traditional	-ional	tradit
skiing	-ing	ski

The most common algorithms for stemming English are Porter's algorithm (Porter, 1980), the Lovins stemmer (Lovins, 1968) and the Paice/Husk stemmer (Paice, 1990).

Lemmatisation usually refers to reduction based on the vocabulary and the morphological analysis of words, normally with an aim to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the *lemma*. An example is given in Table 1.4.

TABLE 1.4: An example of lemmatisation

Form	Morphological information	Lemma
flies	Plural of a noun **fly**	fly
traditional	Adjective derived from a noun **tradition**	traditional
skiing	Gerund of a verb **to ski**	ski

To summarise:

A stem is the base form of a word without any suffixes; the stem can be the same for the inflectional forms of different lemmas;

A stemmer is the process that strips off affixes and leaves a stem (Manning and Schütze, 1999), i.e. the process of collapsing the morphological variants of a word together (Jurafsky and Martin, 2019);

A lemma is a lexicon headword or, more simply, the base form of a word. It is a dictionary-matched base form, unlike the stem obtained by removing/replacing the suffixes; the same lemma can correspond to forms with different stems;

Lemmatisation is a process of replacing words with their lemmas.

For a sample text,[5] behaviour of different stemmers, the implementation of which is available in the NLTK Python module (Bird, Klein, and Loper, 2009), is shown in Table 1.5.

For inflectional languages, better results are often obtained when a full morphological analysis is applied to accurately identify the lemma for each word.

1.2.2 Mapping Texts into a Feature Space

Any text data (novel, news article, review, SMS message, etc.) can be considered a *document*. After pre-processing of the input text (cleaning, lexicon reduction etc.), as described earlier, the next step in TC is to find the most appropriate mapping to the feature space. In other words, one of the most important tasks with TC is to find the best vector representation of text documents. Mapping of text to the vector space of features can be performed using various techniques.

ML algorithms cannot work with raw text directly, so the text has to be converted into a numerical representation. As demonstrated earlier in the example given in Table 1.1, text

[5] Lyrics from the song "El Condor Pasa" by Simon & Garfunkel

TABLE 1.5: A sample text, Porter and Snowball stemmers comparison

I’d rather be a sparrow than a snail, yes I would, if I could, I surely would. Away, I’d rather sail away like a swan that’s here and gone. A man gets tied up to the ground, he gives the world its saddest sound, its saddest sound. I’d rather be a hammer than a nail, yes I would, if I only could, I surely would.
i’d rather be a sparrow than a snail, $\boxed{\text{ye}}$ I would, if I could, I $\boxed{\text{sure}}$ would. away, i’d rather sail away like a swan $\boxed{\text{that'}}$ here and gone. A man $\boxed{\text{get}}$ $\boxed{\text{tie}}$ up to the ground, he $\boxed{\text{give}}$ the world $\boxed{\text{it}}$ saddest sound, $\boxed{\text{it}}$ saddest sound. i’d rather be a hammer than a nail, $\boxed{\text{ye}}$ I would, if I $\boxed{\text{onli}}$ could, I $\boxed{\text{sure}}$ would.
i’d rather be a sparrow than a snail, yes i would, if i could, i $\boxed{\text{sure}}$ would. away, i’d rather sail away like a swan $\boxed{\text{that}}$ here and gone. a man $\boxed{\text{get}}$ $\boxed{\text{tie}}$ up to the ground, he $\boxed{\text{give}}$ the world $\boxed{\text{it}}$ saddest sound, it saddest sound. i’d rather be a hammer than a nail, yes i would, if i $\boxed{\text{onli}}$ could, i $\boxed{\text{sure}}$ would.

documents are usually represented as tabular data. For example, if there are N documents in a collection that should be classified in a certain manner, each document represents a single sample. Each sample should be mapped into a feature space, i.e. a n-dimensional vector for each text document should be obtained. A schematic representation is presented in Table 1.6.

TABLE 1.6: Document-term matrix, a common document representation

sample (document)	$feature_1$	$feature_2$	...	$feature_n$
D_1	x_1^1	x_2^1	...	x_n^1
D_2	x_1^2	x_2^2	...	x_n^2
...			...	
D_N	x_1^N	x_2^N	...	x_n^N

A common and a simple method of feature extraction with text data and the description of the occurrence of words within a document is called the *Bag-of-Words* (BOW) model. It is called so due to the fact that the information about the order or structure of words in the document is discarded. The intuition is that documents are similar if they contain similar sets of words. Hence, as represented by individual words, something about the meaning of the document itself can be learned. BOW model is a representation of text that involves: 1) a union of words present in all text documents and 2) a measure of importance for these words. Each word represents a single feature.

Another common feature space for text representation is a space of linguistic features.

Linguistic Features

Various researchers use the so-called “linguistic features” for different purposes and in different ways. In Table 1.7, some of the linguistic features used by Cristani, Roffo, Segalin, Bazzani, Vinciarelli, and Murino (2012); Roffo, Giorgetta, Ferrario, and Cristani (2014) and Repar and Pollak (2017) are listed. Similar features are used in many existing works. These are selected and shown here, since they are applied in the most similar way to the one later suggested in this thesis. Beside the description of each feature, it is also indicated how each author referred to it. In (Ebert, 2017), the author classifies tweets and SMS messages according to the sentiment they carry: positive, negative or neutral. This was done firstly by mapping messages and tweets into a space of linguistic features. The author distinguishes between two groups: 1) word-based and 2) sentence based linguistic features.

Some authors use very similar features, but differently termed. Cristani, Roffo, Segalin, Bazzani, Vinciarelli, and Murino (2012) use this group of features for identification of a message author in instant messaging. Yet, they do not refer to them as “linguistic” but rather as “stylistic”, and divide them into five major groups: 1) lexical, 2) syntactic, 3) structural, 4) content-specific and 5) idiosyncratic, exactly as Abbasi and Chen (2008). Roffo, Giorgetta, Ferrario, and Cristani (2014) focus on the writing style of individuals, analysing how an individual can be recognised given a portion of chat. In order to examine how personality traits manifest in chats, they extract and analyse various “stylometric” features, which they divide into “lexical” and “syntactic”.

TABLE 1.7: Linguistic features throughout the literature (x - not used; Num - numerical, Nom - nominal, Bin - binary; Lex - lexical, Synt - syntactic, Idios - idiosyncratic, Styl - stylometric/stylistic, Struct - structural, Cont - content specific, w - word, c - char, m - message)

		Cristani, Roffo, Segalin, Bazzani, Vinciarelli, and Murino (2012) [Stylistic]	Roffo, Giorgetta, Ferrario, and Cristani (2014) [Stylometric]	Repar and Pollak (2017) [Linguistic]
# all words	Lex	w-level	Lex	x
# diff. words			x	x
# chars		c-level	Lex	Num
# upp. chars				x
# low. chars				x
# digits			x	Num
# spec. chars			Synt	Num
avg. w len		histogram	Lex	x
hapax legomena		vocabulary richness	x	x
# stop words	Synt	function words	x	Num
# punctuation		punctuation	Synt	Num
# emoticons		emoticons	Synt	x
Contains email, abbreviation...	Struct	m - level	x	x
Bag-of-Words	Cont	w n-grams	x	x
Exclamations			Synt	Num
Abbreviations			x	x
Slang words			x	x
Contains typo	Idios	misspelled word	x	x
# pronouns	x	x	x	Num
# capitalised words	x	x	x	Num
Sentence begins with uppercase and ends with punctuation	x	x	x	Bin
POS of the 1st word	x	x	x	Nom

Part-of-Speech Tagging

As explained in (Kaplan, 2005; Tufis and Ion, 2021), *tagging* is a process of automatic assignment of descriptors, i.e. *tags*, to input tokens. Part-of-Speech information facilitates higher-level analysis, such as recognising noun phrases and other patterns in text (Cutting, Kupiec, Pedersen, and Sibun, 1992). *Part-of-Speech taggers* are computer programs that assign contextually appropriate grammatical descriptors to tokens in text (e.g. noun, verb, adjective, etc.). In the Example 1.2.2, a sample text annotated with POS categories is given.

Example 1.2.2. Can_MD you_PRP water_VB the_DT plants_NNS well_RB using_VBG the_DT water_NN from_IN the_DT well_NN

The explanation of corresponding tags from the Example 1.2.2 can be found in Table 1.8. This example demonstrates homonymy using words *water* (as a verb and as a noun) and *well* (as an adverb and as a noun). Good POS-taggers need to be able to assign appropriate tag to the word, depending on its context in the sentence. For example, good language generation of speech-recognisers relies on POS-tag of a word. A further explanation of the tags used in the example can be found in (Bird, Klein, and Loper, 2009).

TABLE 1.8: Description of POS-tags from the Example 1.2.2

POS-tag	Explanation
MD	Modal verb
PRP	Personal pronoun
VB	Verb, base form
DT	Singular determiner/quantifier
NNS	noun plural
RB	Adverb
VBG	Verb, gerund or present participle
IN	Preposition/subordinating conjunction
NN	Singular or mass noun

POS-tags categories can be differently defined for different languages, and with different level of granularity. Tagsets do not have to be unique per language and they can be redefined for different purposes and text genres. The Universal Tagset[6] defines tags that mark the core part-of-speech categories. Among the first taggers are the CLAWS tagger (Garside, 1987) and the Brill's tagger (Brill, 1992). A state-of-the art POS-tagger is Stanford Part-of-Speech tagger,[7] which can be trained for many languages. For spaCy,[8] various POS-taggers are available for many languages.[9]

POS-taggers are used for many purposes:

- large automatically tagged corpora permit more sophisticated linguistic research;
- POS-tags are also useful for improvement of stop-words removal, since POS-taggers can be trained on corpora that contains tagged stop-words;
- POS-tagging is commonly used as a pre-processing step, since higher levels of analysis benefit from reliable low-level information. POS-tags improve lemmatisation process for converting a word to its base form (lemma);

[6] Universal POS tags, https://universaldependencies.org/u/pos/
[7] Stanford POS-tagger, https://nlp.stanford.edu/software/tagger.shtml
[8] spaCy is Python library for the advanced NLP tasks, https://spacy.io/
[9] POS-taggers in spaCy, https://spacy.io/usage/models

- POS-tags can be used in combination with frequency features, or even alone. Examples of such features are: POS-tag of the word on a certain position in a document, number of nouns, ratio of verbs and nouns, percent of adjectives, etc.
- among many other applications, text indexing and information retrieval systems benefit from POS information (e.g. nouns are better index terms than adverbs or pronouns);
- as it was explained in Example 1.2.2, homographs are words that are spelled equally, but have a different meaning. Word sense disambiguation benefits from correctly assigned POS-tag to a word (e.g. if "well" is a noun or an adverb).

Word Embeddings

A word embedding is a form of representing texts using a vector representation. Based on words that commonly surround a word when it is used, the position of the word within the vector space can be learned from the text. This allows words with similar meaning to have similar representations and to be physically closer in the high-dimensional spaces.

A basic approach for representing words as vectors is using *one-hot* encoding. Essentially, each word is represented as a vector with only one element being 1 and the others being 0. The length of the vector is equal to the size of the vocabulary, as in the BOW model. Conventionally, these unique words are encoded in alphabetical order. Example implementation of one-hot encoding is displayed in Table 1.9:

TABLE 1.9: One-hot vector representation of a token

token	vector
natural	$[1,0,0,0,\ldots]$
language	$[0,1,0,0,\ldots]$
processing	$[0,0,1,0,\ldots]$

Despite its simplicity, the relationship between two words cannot be easily inferred from this words representation. In addition, sparsity is another issue, as with BOW model. Word embeddings offer solution for these issues, and some of them are:

Word2Vec This embedding leverages the context of the words. There are two types of Word2Vec. The first one, Skip-gram, was introduced by Mikolov, Chen, Corrado, and Dean (2013). The main aim of the Skip-gram model is to find word representations that can be used for predicting the surrounding words in a text. Several extensions of the original Skip-gram model are presented in (Mikolov, Sutskever, Chen, Corrado, and Dean, 2013). Mikolov, Yih, and Zweig (2013) introduced a Continuous Bag of Words (CBOW) model. CBOW is very similar to Skip-gram, except that it swaps the input and output. Based on a context, CBOW tries to predict the word which is most likely to appear in it.

FastText Bojanowski, Grave, Joulin, and Mikolov (2017) proposed an approach where each word is represented as a set of character n-grams (Joulin, Grave, Bojanowski, Douze, Jégou, and Mikolov, 2016). In this approach, a vector representation is associated to each character n-gram. Vector representations of words are obtained as the sum of these n-gram vector representations. The proposed method is faster than the original Skip-gram, allowing to train models on large corpora quickly and to compute word representations for words that did not appear in the training data, which was the drawback of the original method. FastText breaks words into several n-grams. For instance, the 3-grams for the word *language* are *lan*, *ang*, *ngu*, *gua*, *uag* and *age*. The character-level word embedding vector for *language* is the sum of all these n-gram vectors. During the training phase, word embeddings for all the n-grams given the training dataset are calculated. This way, even rare words can be properly represented since it is highly likely that some of their n-grams also appear in other words (Bhattacharjee, 2018).

The Appendix A contains Tables A.1, A.2 and A.3 which list and describe stylistic, lexical and syntactical linguistic features, respectively, used for tasks to which this thesis is dedicated. The column *Short name* contains a mnemonic name which will later be used for referring to a certain feature, while the column *Description* contains brief explanation about the feature. Column *G* (group) contains a sub-group of a feature: for lexical features (Table A.2), char-based (c), token-based (t) and sentence-based (s), and for syntactic features (Table A.3) emoticons (e), POS-tag based (p) and other (o). Within the *T* column (type), numerical features are denoted as N, while categorical features are denoted as C. Columns 2, 3, 4 and 5 indicate which feature was used for each of the tasks later described in Chapters 2, 3, 4 and 5 respectively.

1.2.3 Overview of NLP applications and tasks

Below, some tasks and applications of NLP are listed and briefly described.

Question Answering

Semantic parsing of questions is an NLP task that deals with mapping natural language questions to information retrieval queries. It represents a fundamental component for any knowledge-base supported Question Answering (QA) (Yih, He, and Meek, 2014). QA can be divided into semantic parsing-based and unstructured QA. In the first case, a question is translated into a logical form that is executed against a knowledge-base. Since the background knowledge has previously been compiled into a knowledge-base, the challenge is in interpreting the question. For the case of unstructured QA, an answer to a question is offered directly from some relevant text. The model here focuses on matching the question against the document and extracting the answer from some local context, such as a sentence or a paragraph (Talmor, Geva, and Berant, 2017). More about this topic can be read in (Prager, 2021).

Text Summarisation

Paraphrase detection is the task of examining two sentences and determining whether they have the same meaning (Socher, Huang, Pennin, Manning, and Ng, 2011). The ability to detect similar sentences written in natural language is crucial for several applications, such as Text Mining, Text Summarisation, Plagiarism Detection, Authorship Authentication and Question Answering (Agarwal, Ramampiaro, Langseth, and Ruocco, 2018). Text summarisation is the NLP application that deals with creating a short, accurate, and fluent summary of a longer text document. There are two main approaches for text summarisation:

(i) Extraction-based This approach involves pulling key phrases from the text document and combining them to make a summary. The extraction is made according to the defined metric without making any changes to the texts. For example, for the original text:

Jorge *took a day off to* ***participate*** *at the* ***International dance festival*** *held in* ***Madrid****.*
With the sounds of a wonderful Spanish guitar,
Jorge fell in love *with a gorgeous girl named* ***Isabella****.*

A summary based on key phrases extraction would be:

Jorge participate International dance festival Madrid.
Jorge fell in love Isabella.

The words in bold have been extracted and joined to create a summary, which can often produce syntactically incorrect sentences.

(ii) Abstraction-based This approach entails paraphrasing and shortening parts of the source document. It creates new phrases and sentences that relay the most useful information from the original text. It often performs better than the extraction approach, but they are more computationally expensive and challenging to develop.

Following is an example an abstractive summary of the previously given text:

Jorge participated at the International dance festival in Madrid, where he fell in love with Isabella.

More about this topic can be read in (Hovy, 2021).

Machine Translation

Machine Translation (MT) is present in every day usage by many people nowadays. MT systems are designed to help, rather than to replace, professional human translators. These systems are applied in many real word scenarios: professional translators can first obtain a machine output for very large documents, and then post-edit the MT output, they can help with obtaining translations between several languages in real time, etc.

According to their operation and underlying methodology, MT can be classed into the following types (Hutchins, 2005; Specia and Wilks, 2021; Bowker and Pastor, 2021):

(i) Rule-based MT These systems, dated from the 1970s, consisted of: 1) a bilingual dictionary and 2) a set of linguistic rules for each language (e.g., in German, nouns ending in certain suffixes such as -heit, -keit, -ung are feminine, etc); They can be divided into:

• **Direct MT** These systems divide the text into words, obtain translation of each word separately, slightly correct the morphology, and adjust the syntax. An example of such translation is:

Original text: Peter | ate | a | tasty | hot dog

Translated text: Peter | aß | einen | leckeren | heißen Hund

• **Transfer-based MT** Here, an input sentence is firstly parsed. After obtaining grammatical structure of the sentence, rules of the so-called transfer grammar are applied. Afterwards, whole constructions are translated, and not just words separately, as in the direct approach. An example of such translation of previous example in English/German pair is:

Translated text: Peter | aß | einen | leckeren | Hot | dog

• **Interlingual MT** In this method, the aim is to transform the source text to the intermediate representation, which is unified for all the world's languages, i.e. to the *interlingua*. Yet, it turned out that it is impossible to create such universal language representation. On the other hand, it is possible to have different levels of representations, such as morphological, syntactic, and even semantic.

(ii) Example-based MT The basic idea behind these systems is to consult ready-made lists of phrases. If there is a certain sentence that should be translated, and a similar sentence was already translated, then the task is to find the words that differentiate these two sentences and translate them separately.

(iii) Statistical MT The motivation behind these systems, that date from the 1990s, is to analyse *parallel texts* in two languages and to observe the patterns. Two texts are said to be parallel if they have the same content in two different languages. The main disadvantage of this statistical approach is that it requires large amounts of parallel multi-lingual data. The application is simple: the system looks up the queried phrase in the table, and returns its most frequent translation in the training corpus. More about this topic can be found in (Koehn, 2009). There are two main approaches:

• **Word-Based SMT** One sentence translated in two languages is split into words, which are being matched afterwards. This operation is repeated, until tables of all translating words and counts of these occurrences are obtained. The word order is not taken into account. A problem with this approach is when a single word in one language translates into multiple words in another language, or vice versa. There are certain improvements of the approach that try to overcome the issue with order of words;

• **Phrase-Based SMT** This method is based on statistics, reordering, and lexical rules. Instead of words, a text is split into n-grams, i.e., into contiguous sequence of n tokens in a row. In this context, aligned "phrases" are not necessarily phrases from the linguistic point of view. Hence, in this particular context, these n-grams are usually referred as "chunks".

(iv) Neural MT Contrary to the Statistical MT, which is based on various count-based models, Neural Machine Translation is based on a single Artificial Neural Network. Essentially, this ANN is a black box that extracts patterns from a text in one language, encodes these patterns, and decodes them to another language. The idea is somewhat similar to Interlingual MT, but the difference is that the extracted patterns are not known (Koehn, 2020).

Speech Recognition and Text-to-Speech Systems

Speech Recognition (SR) systems merge interdisciplinary technologies from Signal Processing, Pattern Recognition, Natural Language Processing and Linguistics into a unified statistical framework (Lamel and Gauvain, 2021; Dale, 2021). First SR systems with modest vocabularies were limited to a single speaker. Modern SR systems can recognise speech from multiple speakers and large vocabularies in numerous languages. Speech is first converted from physical sound to an electrical signal using microphone as input medium, and then to digital data with an analog-to-digital converter. Most modern speech recognition systems rely on a Hidden Markov Model (HMM) (Juang and Rabiner, 1991). This approach works on the assumption that a speech signal, when viewed on a e.g. ten milliseconds time scale, can be approximated as a stationary process, i.e., a process in which statistical properties do not change over time. Text-to-Speech systems (TTS) convert text into spoken words. This is done by combining NLP techniques and the technology of signal processing. Various IT giants have launched their SR and TTS solutions, that implement various question answering techniques: Apple launched *Siri* assistant,[10] Microsoft developed *Cortana* personal digital assistant,[11] Amazon created *Alexa*[12] etc.

Since this thesis deals with several instances of text classification, a detailed overview of this task and commonly employed techniques is given in the next Section.

1.2.4 NLP Applications Modelled as Text Classification Problems

The classification problem is one of the most fundamental problems in the ML. Almost all well known techniques for classification such as DTs, NB, k-NN, SVM and ANNs have been extended to the case of text data. In recent years, the advancement of web and social network technologies have lead to a tremendous interest in the classification of text documents. The problem of text classification finds applications in a wide variety of domains in text mining. More detailed overview of text classification applications can be

[10] Siri, https://www.apple.com/siri/

[11] Cortana, https://www.microsoft.com/en-us/cortana

[12] Alexa, https://alexa.amazon.com/

found in (Joachims, 2002; Srivastava and Sahami, 2009; Jarvis and Crossley, 2012; Joulin, Grave, Bojanowski, and Mikolov, 2017).

There are various cases of document classification:

News Filtering and Organisation Electronic services of news articles are created on the daily basis in large volumes. In such cases, it is difficult to organise the news articles manually. Therefore, automated methods can be very useful for news categorisation.

Document Organisation and Retrieval A variety of supervised methods may be used for textual documents organisation in many domains. These include large digital libraries of documents, web collections, scientific literature, or even social feeds. Hierarchically organised document collections can be particularly useful for browsing and retrieval.

E-mail Classification and Spam Filtering It is often desirable to classify email in order to determine either the subject or to determine spam email in an automated way. This is also referred to as *spam filtering* or *email filtering*.

Named Entity Recognition Named Entity Recognition (NER) is the task of identifying named entities like personal names, locations, time expressions, etc. in a text (Nadeau and Sekine, 2007; Jarvis and Crossley, 2012). NER systems are often used as the first step in Question Answering, Information Retrieval, Anaphora Resolution, Topic Modeling, etc. Some NER systems are rule-based (Krstev, Obradović, Utvić, and Vitas, 2014; Jaćimović, Krstev, and Jelovac, 2015), while in some cases, NER is considered a TC task.

Classification of Valid Bi-Texts A bi-text is a merged document composed of both source- and target-language versions of a given text. Automatic validation of bi-texts consists of automatically determining if two segments of texts in two languages are valid translations of each other. One of the applications is aimed at detecting translation errors in bilingual texts (Macklovitch, 1994). Another application is evaluating and improving the quality of a Machine Translation system.

Classification of Good Dictionary EXamples In many cases it is important that a dictionary features customisable content, namely that it is able to adapt to specific user needs and to relate dictionary entries to representative sentences that illustrate the meaning of a specific word by showing its usage in a context. If a dictionary entry includes an example which is a good match for the context in which the user has encountered a word, or for the context in which they want to use it, then the user generally gets what they want straightforwardly and in real-time. Thus there is a case for including lots of examples, for lots of different contexts. One of the first extraction tools for representative sentences was Good Dictionary EXamples - GDEX (Kilgarriff, Husák, McAdam, Rundell, and Rychlỳ, 2008; Gorjanc, Gantar, Kosem, and Krek, 2017; Kosem, Koppel, Zingano Kuhn, Michelfeit, and Tiberius, 2019), nowadays used not only by lexicographers, but also in language teaching and learning.

Authorship Identification Authorship Identification (AI), the challenge of inferring characteristics of the author from the characteristics of documents written by that author, is a problem with a long history and a wide range of applications (Juola, 2008; Oakes, 2014; Oakes, 2021). This important topic in the field of NLP enables identification of the most likely author of articles, news or messages. It can be applied to tasks such as identifying an anonymous author, detecting plagiarism or finding a ghost writer.

Sentiment Analysis and Opinion Mining Opinions and its related concepts such as sentiments, evaluations, attitudes, and emotions are the subjects of study of Sentiment Analysis and Opinion Mining (Breck and Cardie, 2021). Availability of plethora of online review sites and personal blogs allows people to actively use information technologies to seek out and understand the opinions of others. The area of Opinion Mining and Sentiment Analysis deals with the computational treatment of opinion, sentiment, and subjectivity in text (Pang and Lee, 2008). Twitter has been providing an inspiring data to many researchers (Pak and Paroubek, 2010; Agarwal, Xie, Vovsha, Rambow, and Passonneau, 2011). Numerous research papers testify that this problem has been very popular for many years now (Pang and Lee, 2004; Wilson, Wiebe, and Hoffmann, 2005; Baccianella, Esuli, and Sebastiani, 2010; Taboada, Brooke, Tofiloski, Voll, and Stede, 2011; Maas, Daly, Pham, Huang, Ng, and Potts, 2011; Liu, 2012).

Hate Speech Detection The increasing propagation of hate speech in the recent years raised the urgent need for effective counter-measures on social media. The anonymity that the Internet offers its users has made it a medium for aggressive communication, as well. As the amount of online hate speech is increasing, methods that automatically detect hate speech are of great value. A large number of methods have been developed for automated hate speech detection on social media (Warner and Hirschberg, 2012; Gitari, Zuping, Damien, and Long, 2015; Nobata, Tetreault, Thomas, Mehdad, and Chang, 2016; Mitrović, Birkeneder, and Granitzer, 2019; Birkeneder, Mitrović, Niemeier, Teubert, and Handschuh, 2019). This aims to classify textual content into non-hate or hate speech, in which case the method may also identify the targeting characteristics (i.e., types of hate, such as race, and religion) in the hate speech (Zhang and Luo, 2019).

1.2.5 Natural Language Processing for Serbian

In this Section, a brief overview of the current state of the language technology support for the Serbian language is given. In (Vitas, Popović, Krstev, Obradović, Pavlović-Lažetić, and Stanojević, 2012), a detailed analysis of the state of the resources and technologies for Serbian is presented. The systematisation proposed in (Vitas, Popović, Krstev, Obradović, Pavlović-Lažetić, and Stanojević, 2012:pp. 71) is partially followed in this overview, enriched with descriptions of recent progress made in this field.

Language Resources

This Subsection describes the existing lexical resources, corpora and knowledge bases developed for Serbian are described.

Text corpora Vitas and Krstev (2012) provided a thorough description of the manually constructed resources for Serbian, namely, the system of morphological electronic dictionaries and semantic networks. This description was followed by a review of some of the Serbian language corpora.

Utvić (2014) dealt with construction of a corpus of contemporary Serbian (SrpKor) as a reference language resource. This electronic corpus of contemporary Serbian was constructed with annotated bibliographical information (corpus texts) and morphological information (part-of-speech and lemma of corpus tokens).

There are various existing domain-specific corpora. Vujičić Stanković and Pajić (2012) developed and described a process of extracting information from meteorological texts in Serbian. In (Krstev, Vujičić-Stanković, and Vitas, 2014), among other contributions, authors compiled a corpus of culinary recipes. Miličević (2015) analysed methods for semi-automatic construction of genre-orientated corpora from the web, comparing four different methods for the case of cooking recipes on the web. Pajić, Vujičić Stanković, Stanković, and Pajić (2018) presented a methodology that contributes to the development of terminology lexica in different areas. Domain concepts were extracted from the agricultural engineering corpus as a case study. The subject of the research presented in (Vasiljević, 2015) concerns specific structural and language rules in legislative corpora in Serbian.

Anđelković, Seničić, and Stanković (2019) presented an aligned parallel English-Serbian corpus for the domain of management, Andonovski, Šandrih, and Kitanović (2019) described the structure of an aligned parallel Serbian-German literary corpus, whilst Tomašević, Stanković, Utvić, Obradović, and Kolonja (2018) developed a mining corpus.

Lexical resources An electronic dictionary is a dictionary developed for automatic text processing. This means that it contains the information which makes it able to solve the problems related to the segmentation, morphological and higher-levels of text processing. For instance, Serbian morphological electronic dictionaries are used for generating all inflected forms of query keywords in Serbian. The electronic dictionary model that proved to be useful for Serbian is based on the finite state automata theory (Vitas and Krstev, 2012).

Krstev (2008) summarised results obtained in developing electronic dictionaries for Serbian. The system of e-dictionaries containing general Serbian lexica (both for Cyrillic and Latin script) consists of a dictionary of simple word forms, a dictionary of multi-word units, and the set of finite-state transducers for the recognition of unknown words that are not recorded in the dictionary.

Development of morphological dictionaries of MWUs is a time-consuming task. This especially applies to the case of Serbian and other languages that have complex morphological structures. Stanković, Obradović, Krstev, and Vitas (2011) strived towards a procedure for an automated production of MWU dictionary. Today, these Serbian e-dictionaries are incorporated into a database (Stanković, Krstev, Lazić, and Škorić, 2018) that also offers services for other NLP applications.

Tree-banks are usually defined as corpora coded with syntactic information. Đorđević (2014) made an important step towards building the Serbian Tree-bank, which would enable an inducement of a rich formal grammar of Serbian to be used for parsing of Serbian texts. The author's focus was on the morphological annotation of sentences.

Mladenović and Mitrović (2013) aimed at construction of formal domain ontology of rhetorical figures for Serbian that can be used for sentiment analysis and opinion mining.

WordNet[13] (Miller, 1995) is a large lexical database of English. It has been recognised as one of the most important resources for the development of NLP applications (information extraction, information retrieval, question answering applications etc.). Due to its promising contribution to NLP-related tasks posed for English language, many researchers started to develop and use WordNet for NLP-related tasks in different languages. Krstev, Pavlović-Lažetić, Vitas, and Obradović (2004) presented two techniques for using textual and lexical resources, such as corpora and dictionaries, in order to develop, validate and re-fine Serbian WordNet (SWN) (Krstev, 2014). In the means of resources, a big step towards development of applications in the culinary domain was made by Vujičić Stanković, Krstev, and Vitas (2014). Mladenović, Mitrović, and Krstev (2014) developed a set of tools that can help developers with easier expansion and maintenance of WordNet.

Grammars Krstev, Stanković, Obradović, and Lazić (2015) dedicated special attention to automatic inflectional class prediction for simple adjectives and nouns and the use of syntactic graphs for extraction of Multi-Word Unit candidates for e-dictionaries, their lemmatisation and assignment of inflectional classes semi-automatically on the basis of lexical resources and local grammars developed for Serbian.

Formal grammar, as a grammar expressed by means of mathematics and logic, presents an integral part of formal language theory. The construction of a formal grammar of Serbian was proposed and described by Đorđević (2017).

Language Technologies

In this Subsection, the developed tools, technologies and applications for Serbian NLP are described.

Basic Tools Krstev, Stanković, and Vitas (2018) presented a procedure for the restoration of diacritics in Serbian texts written using the degraded Latin alphabet. The procedure relies on the the morphological electronic dictionaries, the Corpus of Contemporary Serbian and the local grammars.

Kešelj and Šipka (2008) proposed a general suffix-based method for construction of stemmers and lemmatisers for highly inflectional languages with only sparse resources. The technique was evaluated on a construction of a stemmer for the Serbian language.

[13] WordNet, https://wordnet.princeton.edu/

Resources for language-identification were developed by Zečević and Vujičić-Stanković (2013). The authors tested several top-level language identification tools on a created corpora comprising newspaper articles, literary works written by Serbian authors and the translations of many widely-circulated novels.

In the initial work, Popović (2010) provided a comparative overview of existing morphological taggers and ML methods on which they are based, with practical tests and results about different taggers applied on texts in Serbian. Afterwards, Utvić (2011) trained the TreeTagger (Schmid, 1999) for Serbian and performed and described stages in annotation of the Corpus of Contemporary Serbian (SrpKor), on several levels of annotation. The author also compiled a part-of-speech (PoS) tagset based on the electronic morphological dictionary of Serbian.

Batanović and Nikolić (2017) assessed the impact of lemmatisation and stemming for Serbian on classifiers trained and evaluated on a dataset of film reviews (Batanović, Nikolić, and Milosavljević, 2016).

The existence of large-scale lexical resources for Serbian, e-dictionaries in particular, coupled with local grammars in the form of finite-state transducers, enabled the development of a comprehensive tool for named entity recognition and tagging. Krstev, Obradović, Utvić, and Vitas (2014) targeted some specific types of name, temporal and numerical expressions. This system was later used by Šandrih, Krstev, and Stanković (2019) for the preparation of a gold standard annotated with personal names. It was further used to prepare training sets for four different levels of annotation, on which two additional ML-based NE recognisers were trained and evaluated.

Based on this NER system, Jaćimović, Krstev, and Jelovac (2015) developed an automatic de-identification system for Serbian. Built on a finite-state methodology and lexical resources, the system was designed to detect and replace all explicit personal protected health information present in medical narrative texts.

Acquisition of new terminology from specific domains and its adequate description within terminological dictionaries is a complex task, especially for languages that are morphologically complex such as Serbian (Krstev, Stanković, Obradović, and Lazić, 2015). In their work, authors presented a semi-automatic procedure for terminology acquisition in Serbian on the basis of existing lexical resources and local grammars, i.e. local rules. This approach was afterwards applied to the extraction of multi-word terms in texts belonging to various domain texts. Stanković, Krstev, Obradović, Lazić, and Trtovac (2016) presented a rule-based method for multi-word term extraction that relies on extensive lexical resources in the form of electronic dictionaries and finite-state transducers for modelling various syntactic structures of multi-word terms. The same technology was used for lemmatisation of extracted multi-word terms. The authors demonstrated their approach on the corpus of Serbian texts from the mining domain, with the aim to export the obtained terms to terminological e-dictionaries and databases. Pajić, Vujičić Stanković, Stanković, and Pajić (2018) proposed a domain and language independent hybrid approach, which combines linguistic and statistical information to semi-automatically extract multi-word term candidates from texts. Its performance was evaluated on texts from the agricultural engineering domain.

Importance of terminological resources for specific domains in electronic format is growing with the rapidly expanding availability of various texts on the web. The Termi application[14] supports the development of terminological dictionaries in various fields (mathematics, computer science, mining, library science, computational linguistics, etc.) Termi currently supports the processing and presentation of terms in Serbian, English and recently in German. As mentioned earlier, Andonovski, Šandrih, and Kitanović (2019) enriched the lexical database Termi with a bilingual list of German-Serbian translated pairs of lexical units. This is a product of an enhancement of bilingual search queries in a full-text search of aligned SrpNemKor collection, based on the usage of existing lexical resources such as Serbian morphological e-dictionaries.

Semantic Analysis For automated categorisation of text documents, Graovac (2014) proposed a technique based on byte-level n-grams. The technique provided an effective way of classifying documents using the k-Nearest Neighbours classifier, having experimental data in Serbian, Chinese and English.

For the application of Authorship Attribution, Zečević (2011) proposed a language independent n-gram approach that tries to determine a set of optimal values for number n for specific task of classification of newspaper articles written in Serbian according to authorship.

For the application of Sentiment Analysis (SA), Mladenović, Mitrović, Krstev, and Vitas (2016) described a process of building a Sentiment Analysis Framework for Serbian (SAFOS). This work proposes a hybrid method that uses a sentiment lexicon and SWN synsets assigned with sentiment polarity scores in the process of feature selection.

With a view to improving SA for Serbian, Mladenović, Mitrović, and Krstev (2016) proposed a language-independent process of creating a new semantic relation between adjectives and nouns in WordNets. On a related topic, Ljajić and Marovac (2019) inspected how negation impacts the sentiment of tweets in the Serbian language, whilst Škorić (2017) explored the possibility of using emoticon-riddled text from the web in language-independent SA.

Graovac, Mladenović, and Tanasijević (2019) proposed n-gram based language-independent text representation models for the development of the optimal model for the representation of text documents in various languages, in order to solve the task of classifying texts according to their "positive" or "negative" orientation. The experiments were conducted on film reviews in Serbian, but also for Arabic, Czech, French, and Turkish.

On the topic of irony detection, Mladenović, Krstev, Mitrović, and Stanković (2017) introduced a language dependent model for classification of statements as ironic or non-ironic, based on different language resources and various linguistic features. The evaluation was performed on two collections of tweets that had been manually annotated according to irony.

[14] Termi, http://termi.rgf.bg.ac.rs/

Other Tools University of Belgrade Human Language Technology (HLT) group has produced an integrated and easily adjustable tool, a workstation for language resources, labelled LeXimir (Stanković and Krstev, 2016). This tool augmented the potential of simultaneous manipulation of resources. Leximir has already been successfully used for various language processing related tasks. Similarly, Vebran[15] is a Serbian linguistic web-based service that offers different linguistic capabilities for semantic and morphological extension of a given phrase.

Stanković, Krstev, Vitas, Vulović, and Kitanović (2017) outlined the main features of Bibliša, a tool that offers various possibilities of enhancing queries submitted to large collections of documents generated from aligned parallel articles residing in multilingual digital libraries of e-journals. The tool supports semantically and morphologically expandable keyword queries, full-text and metadata search, extraction of concordance sentence pairs for translation and a support for work with terminologies.

Pajić, Vitas, Pavlović-Lažetić, and Pajić (2013) developed a software system called WebMonitoring, designed for improving information search on the web. The architecture of the WebMonitoring system relies upon finite state machines.

The first experiments in Machine Translation were conducted in the doiman of E-learning. Course translation is a very special service that requires specific subject matter expertise and high technical skills. The tools for translation of e-learning courses and translation support were analysed by Obradović, Dalibor, Ranka, Nikola, and Miladin (2016). Beside presenting the current state of research in course translation, the authors outlined that the translation of electronic courses itself is an ongoing activity at the Faculty of Mining and Geology of the University of Belgrade.

Šimić (2019) proposed a technological solution that combines different technologies and improves e-government search services, making them also accessible to people with disabilities. This solution is based on the previously-mentioned Vebran service.

Jovanović, Šimić, Čabarkapa, Ranđelović, Nikolić, Nedeljković, and Čisar (2019) presented SEFRA, a web-based framework for searching Web content. The system supports indexing, searching and displaying search results adjusted to Serbian. The implemented system is based on several advanced Serbian language services accessible over the Web.

As a support for the developed Named Entity Recogniser, Šandrih, Krstev, and Stanković (2019) joined various existing and developed several new tools and combined them into a Web platform NER&Beyond.[16] The platform features different Named Entity Recognisers for Serbian, which can also be used for NER and visualisation on-line.[17]

At the moment, the Serbian language does not have any resources for natural language generation. Starting from the summer of 2019, Human Language Technology group (HLT) joined the European COST action: Multi3Generation: Multi-task, Multilingual,

[15] Vebran, `hlt.rgf.bg.ac.rs/VeBran`

[16] NER&Beyond, `http://nerbeyond.jerteh.rs/`

[17] Visualisation of NER models for Serbian, `http://ner.jerteh.rs/`

Multi-modal Language Generation,[18] with an intention to improve the status of the Serbian language in this field.

The following chapters are organised as follows. In Chapter 2, the task of automatic validation of bilingual domain-specific terminology pairs is presented. Next, the task of automatically assigning a good usage example for a given dictionary entry is described and a text-classification-based solution is proposed in Chapter 3. Contributions of automatic authorship attribution of short texts are discussed in Chapter 4. The challenges of sentiment analysis in short messages are outlined in Chapter 5. For all of the four cases, new solutions are proposed and evaluated. Finally, Chapter 6 offers conclusions and outlines future work.

[18] CA18231 — Multi3Generation: Multi-task, Multilingual, Multi-modal Language Generation, https://www.cost.eu/actions/CA18231

2 Extraction and Validation of Bilingual Terminology Pairs

This chapter proposes an approach for automatic bilingual terminology extraction and validation. Despite the technique being language-independent, its effectiveness is demonstrated on an English-Serbian language pair, having English as source and Serbian as target language. The original methodology is presented and evaluated in this chapter which also reports the development of a classifier which separates correct pairs returned by the implemented system.

A deeper insight to the problem is given in Section 2.1, along with an overview of previous related work in Section 2.2. The proposed approach for terminology extraction is explained in detail in Section 2.3. Next, a method for semi-automatic validation of the obtained results from the previous extraction is presented in Section 2.4, which is based on classification of the obtained bilingual terminology pairs. The adaptation of the proposed methods for terminology extraction in the case of English/Serbian language pair is described in Section 2.5. The results are discussed Section 2.6. Finally, the concluding remarks and directions for future work are listed in Section 2.7.

2.1 Introduction

In science, industry and many research fields, terminology is developing fast. It is a challenge to deliver and maintain up-to-date terminological resources, especially for languages that are in need of many Natural Language Processing resources and tools. Such is the case for Serbian, for which terminological resources in many domains, if available, tend to be outdated. Given the world supremacy of the English language, domain terms are first coined in English and only after that translated into other languages. It does not happen rarely that a certain term is translated either as a short explanation of its meaning, or the translation is adapted directly so it "sounds" like a word in a target language. An example that demonstrate both cases is an English word "a compiler", from the computer science. In Serbian, this term is either translated as *program koji prevodi kôd iz jednog programskog jezika u drugi* (namely, *a computer program that translates code written in one programming language to another*) or as a "kompajler" (i.e, the word is borrowed). It is common that even experts from a certain field have difficulties while translating texts that contain domain terminology. As in the example with a "compiler", the original word is borrowed for everyday use in IT domain.

In this Chapter, an approach for extracting bilingual terminology pairs automatically is described. Its effectiveness is demonstrated for English/Serbian language pair. The research question addressed in this study is the following: is it possible, on the basis of bilingual, aligned, domain-specific textual resources, a terminological list and/or a term extraction tool for the source language, as well as a system for the extraction of terminology-specific noun phrases in a target language, to compile a bilingual aligned terminological list?

Multi-Word Expressions (MWEs) are lexical units composed of two or more words. What is common for these words is that they are semantically, pragmatically, syntactically, and/or statistically idiosyncratic (Baldwin and Kim, 2010). Words that form MWEs are usually referred to as *components*. As Monti, Seretan, Pastor, and Mitkov (2018) explain, at least one component of the MWE is restricted by linguistic conventions in the sense that it is not freely chosen.

One big challenge about MWEs is that the semantics of the whole is most often not related to the meanings of the individual containing words. Further, in most cases it does not make sense to replace containing words with their synonyms. Finally, component words of MWE can be contained in the same order as in the original MWE, but only as a simple sub-string: for example, MWE 'by and large' (e.g., *by and large we agree* versus *he walked by and large tractors passed him*) (Constant, Eryiğit, Monti, Van Der Plas, Ramisch, Rosner, and Todirascu, 2017). In spite of these challenges, as Mitkov (2021) noticed, research in NLP has made significant progress in the computational treatment of MWEs.

Baldwin and Kim (2010); Schneider and Smith (2015); Constant and Nivre (2016); Constant, Eryiğit, Monti, Van Der Plas, Ramisch, Rosner, and Todirascu (2017); Monti, Seretan, Pastor, and Mitkov (2018); Mitkov (2021) describe Multi-Word Expressions in more details.

This chapter is aimed at MWEs since terminology consists mainly of Multi-Word Terms (MWTs), which are domain-specific MWEs. Terms consisting of a single word are mainly referred to as Single-Word Terms (SMTs). Therefore, the previous question can be reformulated as follows: on the basis of bilingual, aligned, domain-specific textual resources, a terminological list and/or a term extraction tool in a source language, and a system for the extraction of *terminology-specific Multi-Words Terms* in a target language, it is possible to compile a bilingual aligned terminological list?

2.2 Related Work

Over the past years, in order to compile bilingual lexica, researchers used various techniques for MWT extraction and alignment that differ in methodology, resources used, languages involved and purpose for which they were built.

Language pairs for which bilingual lexica were developed include: English/Romanian (Pinnis, Ljubešic, Stefanescu, Skadina, Tadic, and Gornostay, 2012; Kontonatsios, Mihăilă, Korkontzelos, Thompson, and Ananiadou, 2014), Bengali/Hindi/Tamil/Telugu (Irvine and Callison-Burch, 2016), English/French (Bouamor, Semmar, and Zweigenbaum, 2012; Hamon and Grabar, 2018; Hazem and Morin, 2016; Hakami, and Bollegala, 2017; Semmar, 2018), English/Slovene (Vintar and Fišer, 2008), English/Croatian, Latvian and Lithuanian (Pinnis, Ljubešic, Stefanescu, Skadina, Tadic, and Gornostay, 2012), English/Spanish (Oliver, 2017; Ha, Mitkov, and Corpas, 2008; Mitkov, 2016), English/Arabic (Lahbib, Bounhas, and Elayeb, 2014; Sabtan, 2016; Hewavitharana and Vogel, 2016), English/Chinese (Xu, Chen, Wei, Ananiadou, Fan, Qian, Eric, Chang, and Tsujii, 2015), English/Hebrew (Tsvetkov and Wintner, 2010), English/Urdu (Hewavitharana and Vogel, 2016), English/Italian and English/German (Arcan, Turchi, Tonelli, and Buitelaar, 2017), English/Ukrainian (Hamon and Grabar, 2018), English/Greek (Kontonatsios, Mihăilă, Korkontzelos, Thompson, and Ananiadou, 2014), Slovak/Bulgarian (Garabík and Dimitrova, 2015), Italian/Arabic (Fawi and Delmonte, 2015) and English-Italian (Taslimipoor, Desantis, Cherchi, Mitkov, and Monti, 2016).

Bilingual lists of MWTs were in several cases compiled with an aim to improve statistical machine translation (SMT) of an existing machine translation system (Bouamor, Semmar, and Zweigenbaum, 2012; Tsvetkov and Wintner, 2010; Sabtan, 2016; Irvine and Callison-Burch, 2016; Semmar, 2018; Hewavitharana and Vogel, 2016; Arcan, Turchi, Tonelli, and Buitelaar, 2017; Oliver, 2017), for the development of an existing language resource in a target language on the basis of a corresponding resource in a source language (e.g. Vintar and Fišer (2008) improved Slovenian WordNet based on English WordNet), or for the presentation of bilingual correspondences between two languages (e.g. correspondences between Slovak-Bulgarian parallel corpus (Garabík and Dimitrova, 2015)).

In some cases, seed lexicon was used (Tsvetkov and Wintner, 2010; Xu, Chen, Wei, Ananiadou, Fan, Qian, Eric, Chang, and Tsujii, 2015; Semmar, 2018) or existing translation memories and phrase tables (Oliver, 2017). Beside the input corpus, additional resources were not required in several other cases (Sabtan, 2016; Arcan, Turchi, Tonelli, and Buitelaar, 2017; Garabík and Dimitrova, 2015; Hewavitharana and Vogel, 2016; Bouamor, Semmar, and Zweigenbaum, 2012).

Parallel sentence-aligned data was requested in some techniques (Arcan, Turchi, Tonelli, and Buitelaar, 2017; Garabík and Dimitrova, 2015; Bouamor, Semmar, and Zweigenbaum, 2012; Semmar, 2018), while other techniques performed the extraction on comparable corpora (Xu, Chen, Wei, Ananiadou, Fan, Qian, Eric, Chang, and Tsujii, 2015; Hazem and Morin, 2016; Hewavitharana and Vogel, 2016; Pinnis, Ljubešic, Stefanescu, Skadina, Tadic, and Gornostay, 2012). Sabtan (2016) used groups of aligned sentences (verses). Irvine and Callison-Burch (2016) performed two experiments: the first one relying on the existence of a bilingual dictionary with no parallel texts, and the second one requiring only the existence of a small amount of parallel data. Bilingual sentence-aligned data is essential for other MT-related applications, as well. For example, the concept of translation memory tools is based on the idea that a translator should benefit from existing bilingual sentence-aligned data (Mitkov, 2020) as much as possible.

Some authors apply machine learning and mathematical optimisation techniques for the extraction and validation of terminology. In (Hakami, and Bollegala, 2017), the problem of determining whether a target term was the correct translation of a given source term is treated as a problem of binary classification. With bilingual domain glossary as a training set, the authors created a binary classifier that is able to tell if two terms are translations between the two languages. Hakami, and Bollegala (2017) represented a term in a language using character *n*-gram features extracted from a term (able to capture useful properties about a term such as its inflection), and contextual features (words that appear within a certain window surrounding the term under consideration). They used mean average precision and *k*-top translation accuracy as evaluation metrics.

Bouamor, Semmar, and Zweigenbaum (2012) proposed an approach for generating MWE pairs that extracts MWEs from the source language (English) by using predefined morpho-syntactic patterns, and aligns them with their translations in the target language (French) using the vector-space model.

Vintar and Fišer (2008) used English MWEs from the Princeton WordNet (Miller, 1995) in order to extract Slovene MWE equivalents from a monolingual corpus by using three different techniques. For the first translation-based approach, they extracted a list of MWEs from PWN with an aim to match these MWEs with their Slovene counterparts in the corresponding corpus. With the second seed-word-based approach, authors extracted a list of MWEs from the corpus, which was then filtered using a list of seed terms. In the last approach, they used lexico-syntactic patterns to extract words from the corpus that are in a hypernym-hyponym relation, which were then mapped to WordNet.

Pinnis, Ljubešic, Stefanescu, Skadina, Tadic, and Gornostay (2012) presented methods for term extraction and bilingual mapping, as well as term tagging in comparable documents, based on existing term extraction techniques. Bilingual term mapping was applied and evaluated for English, on one side, and Croatian, Latvian, Lithuanian and Romanian, on the other. In order to find possible translation equivalents of terms tagged in bilingual comparable corpora, authors developed a term mapping tool TERMINOLOGYALIGNER. Given bilingual document pairs with tagged terms, the developed tool extracts two lists of terms, and assigns scores to candidate pairs.

Garabík and Dimitrova (2015) used hunalign software (Varga, Halácsy, Kornai, Nagy, Németh, and Trón, 2005) to align Bulgarian and Slovak input texts on the sentence level. This sentence-aligned corpus was processed with MOSES (Koehn, Hoang, Birch, Callison-Burch, Federico, Bertoldi, Cowan, Shen, Moran, and Zens, 2007), following the word-level alignment using GIZA++ (Och and Ney, 2000), with the *grow-diag-final* heuristic (Koehn, Och, and Marcu, 2003). More about GIZA++ can be found in Section 2.5. Translation pairs were selected using a score function resulting from a combination of four phrase translation scores, i.e. probabilities stored in GIZA++'s resulting phrase-table. The pairs with a score below a specified threshold were discarded.

Another system for automatic terminology extraction and automatic detection of translations in the target language using translation phrase-table obtained by GIZA++ is presented by Oliver (2017). The system is intended to be used alongside a computer assisted translation tool, which provides term candidates and their translations within an observed text segment. The system is based on the text from the segment being translated, the translation memories assigned to the project and the GIZA++ phrase-table. It also uses a terminological database assigned to the project in order to avoid presenting already known terms.

In (Semmar, 2018), the problem of MWE extraction and alignment from parallel corpora was perceived as an NP hard problem of integer linear programming. In order to find the best alignment of MWEs in the target and source languages, an appropriate scoring function was defined. The result of this step was a list of alignment pair candidates. In the second step, filtering of the valid pairs from this list was performed using corresponding morpho-syntactic patterns for the construction of bilingual lexica of MWEs.

Arcan, Turchi, Tonelli, and Buitelaar (2017) performed monolingual extraction of domain-specific terms from a small parallel corpus. For this purpose, they used three different terminology extractors: KX TOOLKIT (Pianta and Tonelli, 2010), TWSC (Pinnis, Ljubešic, Stefanescu, Skadina, Tadic, and Gornostay, 2012) and ALCHEMYAPI.[1] Once they obtained these lists of automatically extracted monolingual terms for the source and target languages, authors performed bilingual terminology alignment. Given a source term and the parallel sentence pair in which it appears, a set of possible translations is found by either translating the term with an existing SMT system trained on the same corpus, or by applying a word aligner. Having a set of possible translations for each term, the correct one is retrieved if a target translation from the candidate list matches a span of words in the target sentence (sentence lookup), or if it has also been identified as a term in the target sentence by the monolingual term extractor (term lookup). They compared their approach with the existing systems for terminology alignment: TERMALIGNER (Aker, Paramita, and Gaizauskas, 2013) and PHRASETABLE2GLOSSARY (Thurmair and Aleksić, 2012).

2.3 Proposed Method for Extraction of Terminology Pairs

The proposed method was initially published in coauthor work with (Krstev, Šandrih, Stanković, and Mladenović, 2018). The proposed technique was later improved and the extended approach was published in (Šandrih, Krstev, and Stanković, 2020). This proposed method relies on the existence of a parallel sentence-aligned corpus. First, lexica of monolingual domain terminology is obtained on both, the source and the target side. This is done either by using the existing lists or by exploiting terminology extractors. After the resulting monolingual terminology lists are available, corresponding pairs are automatically aligned, yielding a bilingual list of candidate translation pairs of domain terms. These pairs are then ready for manual evaluation. Pairs that are evaluated as valid translations represent good pairs for domain terminology lists. The proposed approach is illustrated as workflow in Figure 2.1.

[1] AlchemyAPI,
http://www.alchemyapi.com/products/features/keyword-extraction/

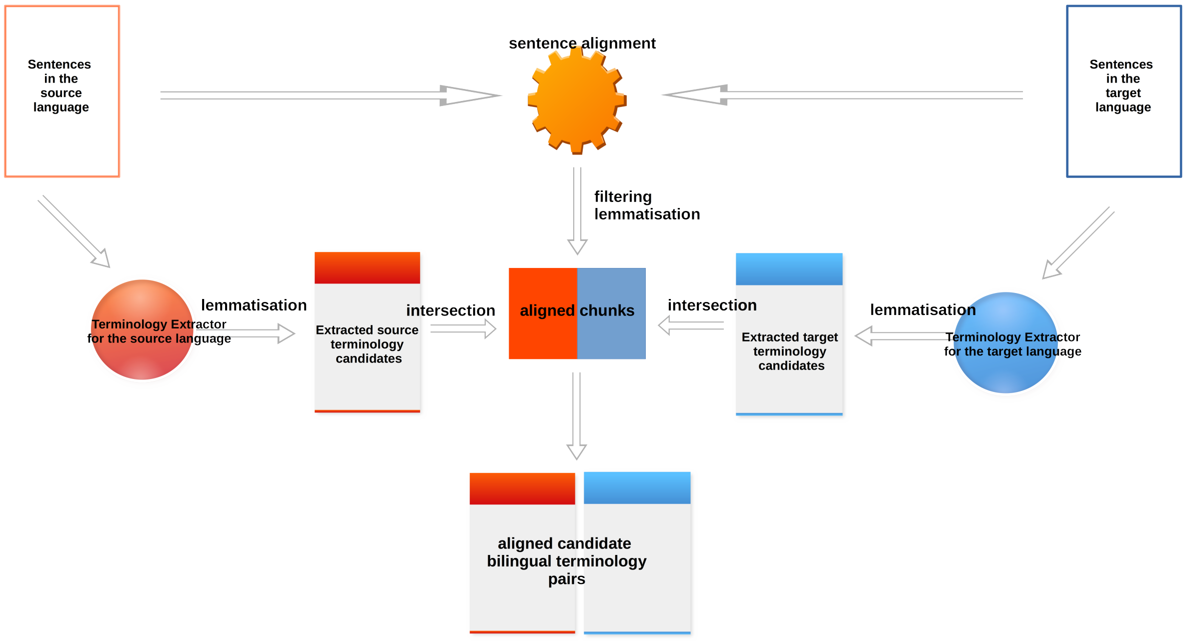

Sentences in the source language
sentence alignment
Sentences in the target language
filtering
lemmatisation
lemmatisation
Terminology Extractor for the source language
Extracted source terminology candidates
intersection
aligned chunks
intersection
Extracted target terminology candidates
lemmatisation
Terminology Extractor for the target language
aligned candidate bilingual terminology pairs

FIGURE 2.1: Workflow of the proposed approach for terminology extraction

The conceptual design of the proposed system for terminology extraction is as follows:

1. Input:

   i A sentence-aligned domain-specific corpus involving a source and a target language. An entry in this corpus is denoted by $S(text.align) \leftrightarrow T(text.align)$. Let the $T(text.lalign)$ later represents $T(text.align)$ lemmatised word-by-word;

   ii A list of terms in the source language. This list can be either an external resource from the same domain or extracted from the text. An entry in this list is denoted by $S(term)$;

   iii A list of terms in the target language. This list can be either an external resource from the same domain or obtained from the text. An entry in this list is denoted by $T(term)$.

2. Processing:

   i Aligning bilingual chunks (possible translation equivalents) from the aligned corpus. Aligned chunks are denoted by $S(align.chunk) \leftrightarrow T(align.chunk)$; Let the $T(align.lchunk)$ later represents $T(align.chunk)$ lemmatised word-by-word;

   ii Initial filtering of the chunks so that only chunks in which the source part of the chunk matches a term in the list of domain terms in the source language remain: $S(align.lchunk) \sim S(term)$, where the symbol $\sim$ denotes the relation "match" (explained later);

iii Subsequent filtering of the chunks that remained after the initial filtering so that only chunks in which the target part of the chunk matches a term in the list of extracted MWTs in the target language remain: $T(align.chunk) \sim T(term)$;

3. The result:

   - A list of matching source and target terms $S(term) \leftrightarrow T(term)$, obtained from the aligned chunks:

$$(S(term) \sim S(align.chunk)) \wedge (T(term) \sim T(align.lchunk)) \\ \wedge \ (S(align.chunk) \leftrightarrow T(align.lchunk))$$

The relation "match" ($\sim$) is defined as follows: if a chunk is represented by an unordered set of distinct words obtained from the chunk after removal of stop words, the two chunks match if they are represented by the same set. For example, if one chunk is "information dissemination" and another one is "dissemination of information", their corresponding set representations are {information, dissemination} and {dissemination, information}, respectively ('of' should be discarded as a functional word). Since these two sets are equal, these two chunks match. This applies for both sides of the aligned corpus.

Let two candidate pair chunks be "u digitalnoj biblioteci" (in digital library) and "digitalne biblioteke" (digital libraries). For a person who understands both Serbian and English these two chunks should match. Yet, if "match" relation is defined in the presented manner, they do not. If observed as unordered set of distinct words, these chunks can be written as {digitalnoj, biblioteci} and {digitalne, biblioteke}, respectively ("u" is a preposition, meaning *in*, and should be discarded as a functional word). For the best matching possible, chunks have to be normalised. This especially applies to highly inflectional languages, such as Serbian. In this specific case, simple-word lemmatisation within MWTs is needed. This means that each simple word from a MWT has to be replaced by a corresponding lemma. In this case, lemmas were obtained from the morphological e-dictionaries (Krstev, 2008). For example, in the chunk "u digitalnoj biblioteci", a word "digitalnoj" is an adjective, in the feminine gender, in singular and in the genitive case. A lemma for an adjective is in singular, in the masculine gender and in the nominative case, namely "digitalni" for this adjective. In the second chunk "digitalne biblioteke", a word "digitalne" is also an adjective, but in the plural number, in the feminine gender, and in the nominative or the accusative case (it can also be the form of the same adjective in the single number, the feminine gender and the genitive case). The same applies for the forms "biblioteci" and "biblioteke". After single-word lemmatisation, this word is replaced with its lemma "digitalan". After simple-word lemmatisation, previously mentioned sets become equal {digitalni, biblioteka} and {digitalni, biblioteka} (in English {digital, library}), and they, therefore, match.

For word-level alignment of a sentence-aligned corpus, GIZA++ (Och and Ney, 2000) was used.

In order to compile a bilingual lexicon for a specific domain, several settings were combined and compared. Besides using only a parallel sentence-aligned corpus, an experiment was conducted where sentences from the corpus were extended with a bilingual list of inflected word forms from a general-purpose dictionary. Tsvetkov and Wintner (2010) also used a bilingual list from a general-purpose dictionary, albeit for a different purpose, namely as an enhancement in the step where they identify terminology candidates, and not in order to extend the corpus. Hazem and Morin (2016) used an existing bilingual general-purpose dictionary for improving the vector space model representation for each term.

With different configurations for the extraction of domain terminology on both, source and target sides, was experimented. For the source side, two cases were compared. In the first case, an existing bilingual domain dictionary was used. The aim is to obtain translations of existing source terms on the target side, and evaluate these obtained translations against the existing target terms from the dictionary. A similar approach was proposed by Vintar and Fišer (2008). Their goal was to translate English terms from the Princeton WordNet (Miller, 1995) to Slovene in order to enrich the Slovene WordNet. Hakami, and Bollegala (2017) also used an existing list of source terms from the biomedical domain, which they later mapped to an existing list of terms on the target side. Kontonatsios, Mihăilă, Korkontzelos, Thompson, and Ananiadou (2014) used the biomedical meta-thesaurus for the evaluation of their proposed method.

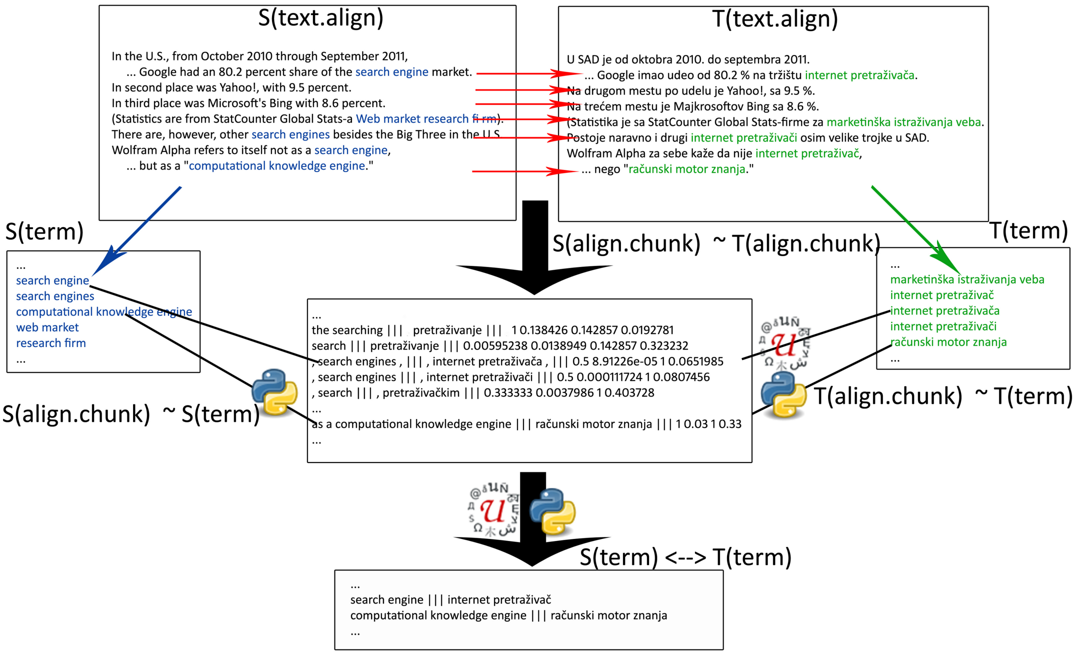
In the second case, source terminology was obtained using an existing term extractor. For this purpose, several existing extraction tools for English were compared, similarly to some other authors (Pinnis, Ljubešic, Stefanescu, Skadina, Tadic, and Gornostay, 2012; Hamon and Grabar, 2018; Oliver, 2017; Arcan, Turchi, Tonelli, and Buitelaar, 2017). For the extraction of terminology on the target side, morphological analysis was applied. A similar approach for other languages was applied by different authors (Bouamor, Semmar, and Zweigenbaum, 2012; Lahbib, Bounhas, and Elayeb, 2014; Fawi and Delmonte, 2015; Hamon and Grabar, 2018; Sabtan, 2016; Semmar, 2018).

Example

Figure 2.2 illustrates how the proposed approach works. On the source side, the following MWTs can be observed: *search engine*, *web market research firm*, *search engines*, *computational knowledge engine*. Ideally, ENG-TE would detect all these MWTs and store them in the resulting $S(term)$ list. Simultaneously, the following MWTs should SERB-TE detect: “internet pretraživača” (search engine, accusative case), “marketinška istraživanja veba” (web market research), “internet pretraživači” (search engines), “internet pretraživač” (search engine, nominative case), “računski motor znanja” (computational knowledge engine). These are then stored in $T(term)$ list. In practice, as there is no extractor that works perfectly, it is not surprising that some MWTs were not correctly recognised (e.g. *web market* and *research firm* are recognised as two terms).

The parallel corpus aligned at sentence level was processed by GIZA++. The result is a list of bilingual aligned chunks (dubbed as $S(align.chunk) \leftrightarrow T(align.chunk)$) and their corresponding translation probabilities (explained later in Subsection 2.5). The paired chunks with low translation probabilites are removed. Some pre-processing steps follow on $S(term)$ and $T(term)$: lemmatisation and duplicates removal. After lemmatisation, *search engine* and *search engines* represent the same term. After removing duplicates, all three occurrences of *search engine* are reduced to a single occurrence.

The first processing step is to intersect $S(align.chunk)$ with $S(term)$. Afterwards, $S(align.chunk) \leftrightarrow T(align.chunk)$ is reduced, because only the candidate chunks whose source chunk was present in the $S(term)$ list was kept. Next, the $T(align.chunk)$ is intersected with $T(term)$. This yields the refined list of bilingual aligned chunks, namely, in this example, *search engine* is matched with *internet pretraživača* and *computational knowledge engine* is matched with *računski motor znanja*, which is correct.



In the U.S., from October 2010 through September 2011,
... Google had an 80.2 percent share of the search engine market.
In second place was Yahoo!, with 9.5 percent.
In third place was Microsoft's Bing with 8.6 percent.
(Statistics are from StatCounter Global Stats-a Web market research fi rm).
There are, however, other search engines besides the Big Three in the U.S.
Wolfram Alpha refers to itself not as a search engine,
... but as a "computational knowledge engine."

U SAD je od oktobra 2010. do septembra 2011.
... Google imao udeo od 80.2 % na tržištu internet pretraživača.
Na drugom mestu po udelu je Yahoo!, sa 9.5 %.
Na trećem mestu je Majkrosoftov Bing sa 8.6 %.
(Statistika je sa StatCounter Global Stats-firme za marketinška istraživanja veba.
Postoje naravno i drugi internet pretraživači osim velike trojke u SAD.
Wolfram Alpha za sebe kaže da nije internet pretraživač,
... nego "računski motor znanja."

```
...
search engine
search engines
computational knowledge engine
web market
research firm
...
```

```
...
marketinška istraživanja veba
internet pretraživač
internet pretraživača
internet pretraživači
računski motor znanja
...
```

```
...
the searching |||  pretraživanje |||  1 0.138426 0.142857 0.0192781
search ||| pretraživanje ||| 0.00595238 0.0138949 0.142857 0.323232
, search engines , ||| , internet pretraživača , ||| 0.5 8.91226e-05 1 0.0651985
, search engines ||| , internet pretraživači ||| 0.5 0.000111724 1 0.0807456
, search ||| , pretraživačkim ||| 0.333333 0.0037986 1 0.403728
...
as a computational knowledge engine ||| računski motor znanja ||| 1 0.03 1 0.33
...
```

```
...
search engine ||| internet pretraživač
computational knowledge engine ||| računski motor znanja
...
```

FIGURE 2.2: An example of terminology extraction

2.4 Proposed Method for Validation of Bilingual Pairs

The knowledge acquired in the previous step is a useful resource in automating the process of terminology lexica generation for the same domain. After manual evaluation of the compiled list of bilingual MWTs, automatic validation of candidate pairs was considered. The idea was to develop a sequence of steps to be added at the end of the previously described procedure, which would separate correct from incorrect translations. To that end, a Radial Basis Function Support Vector Machine (RBF SVM) classifier is proposed, with the next classes: OK for pairs that represent correct translations (positive class), and NOK for the pairs that do not (negative class).

The proposed approach is illustrated in Figure 2.3.

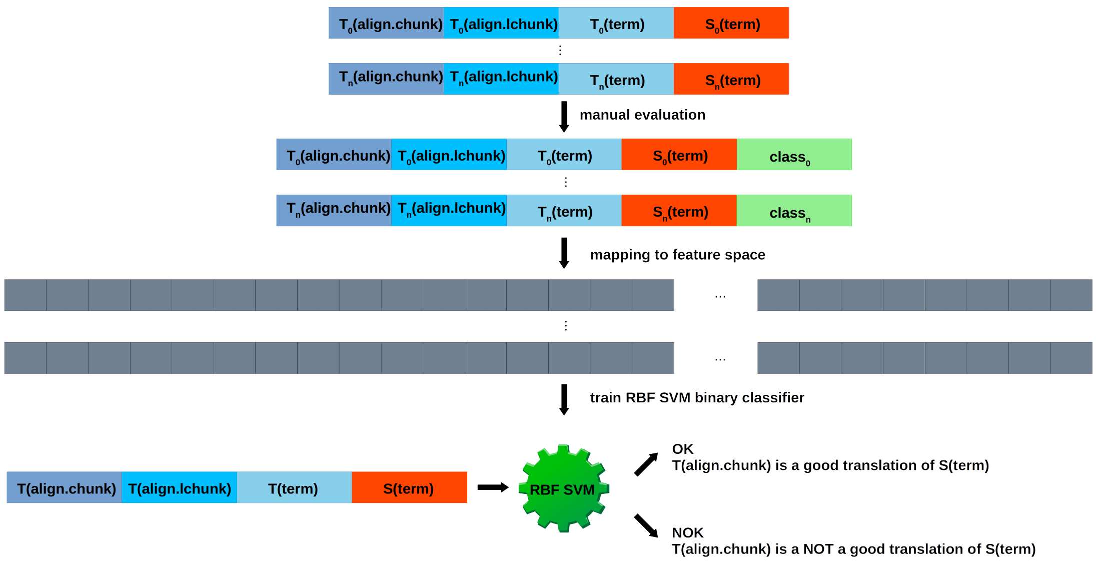


FIGURE 2.3: Diagram of the proposed approach for validation of bilingual pairs

After manual evaluation, to each quadruple ($T(align.chunk)$, $T(align.lchunk)$, $T(term)$, $S(term)$) class label is assigned: if $T(align.chunk)$ translates as $S(term)$, the label is 'OK', otherwise the label is 'NOK'. Afterwards, different lexical and syntactic linguistic features are extracted from the quadruple. A similar approach to classification of terms was proposed by Hakami, and Bollegala (2017), where the authors represented each term using character n-grams features, extracted from a term, and contextual features. In the proposed approach, feature-represented samples are then fed into the RBF SVM classifier. The model is trained to predict, for the quadruple not seen in the training set, whether the corresponding translation pair is valid or not.

The pre-processing step required for the proposed approach is a Part-of-Speech tagging of each component of the quadruple. An example of such pre-processed quadruple is shown in Table 2.1.

TABLE 2.1: An example of the input quadruple

$T(align.chunk)$	$T(align.lchunk)$	$T(term)$	$S(term)$	class
_A_projektne	_A_projektni	_A_projektni	_NOUN_project	OK
_N_dokumentacije	_N_dokumentacija	_N_dokumentacija	_NOUN_documentation	
_PRO_koje				

For this specific application, the Appendix A lists the proposed set of lexical and syntactic linguistic features in Tables A.2 and A.3, respectively (indicated by X in column V). Namely, for each of the 4 components from the quadruple, 31 features are proposed for extraction, yielding a total of 124 features. Components from the quadruple can be observed in pairs (6 different combinations), with 9 features proposed to be extracted from each pair, yielding 54 features in total. These 31 features that can be extracted from single MWTs are in latter referred as "single" features, and the 9 features that are proposed to be extracted from pairs as "joint" ones. Finally, this yields 178 features in total. For example, several feature values from Table A.2, for the quadruple given in the Table 2.1, are listed in Table 2.2.

TABLE 2.2: Values of several lexical single features for the components from the quadruple given in Table 2.1

	num_tokens	num_vocals	avg_token_len	sentence_length
$T(align.chunk)$	3	11	8.67	28
$T(align.lchunk)$	2	9	11	23
$T(term)$	2	8	11	23
$S(term)$	2	8	10	21

For the joint lexical feature *cmn_substr_longer_6* (exists common substring having 6 or more characters), the corresponding feature values extracted for the quadruple from the Table 2.1 are listed in Table 2.3.

TABLE 2.3: Values of the *cmn_substr_longer_6* feature for the combinations of the components from the quadruple given in Table 2.1

$T(align.chunk)$ $T(align.lchunk)$	$T(align.chunk)$ $T(term)$	$T(align.chunk)$ $S(term)$	$T(align.lchunk)$ $T(term)$	$T(align.lchunk)$ $S(term)$	$T(term)$ $S(term)$
True	True	False	True	False	False

After the feature extraction (all categorical features are automatically encoded to consistent numerical values after the feature extraction step), it is desirable investigating how different features influence overall classifier's performance.

2.5 Adaptation of the Proposed Method for Serbian Terminology Extraction

In this section, details of the specific technical system developed for the case of English/Serbian language pair are discussed. For the evaluation of the proposed methodology, the domain of Library and Information Science was selected. Based on the pipeline displayed in Figure 2.1, resources and tools used for the experiments are shown in Figure 2.4, and briefly described in the next paragraphs.

As explained earlier, firstly lexica of monolingual domain terminology are obtained for both source and target languages. This is done either by using the existing terminology lists or by exploiting terminology extractors. After generating monolingual terminology lists, the corresponding pairs are automatically aligned, yielding a bilingual list of candidate translation pairs of domain terms. Under the assumption that the described resources exist, the proposed pipeline can be adapted for other languages.

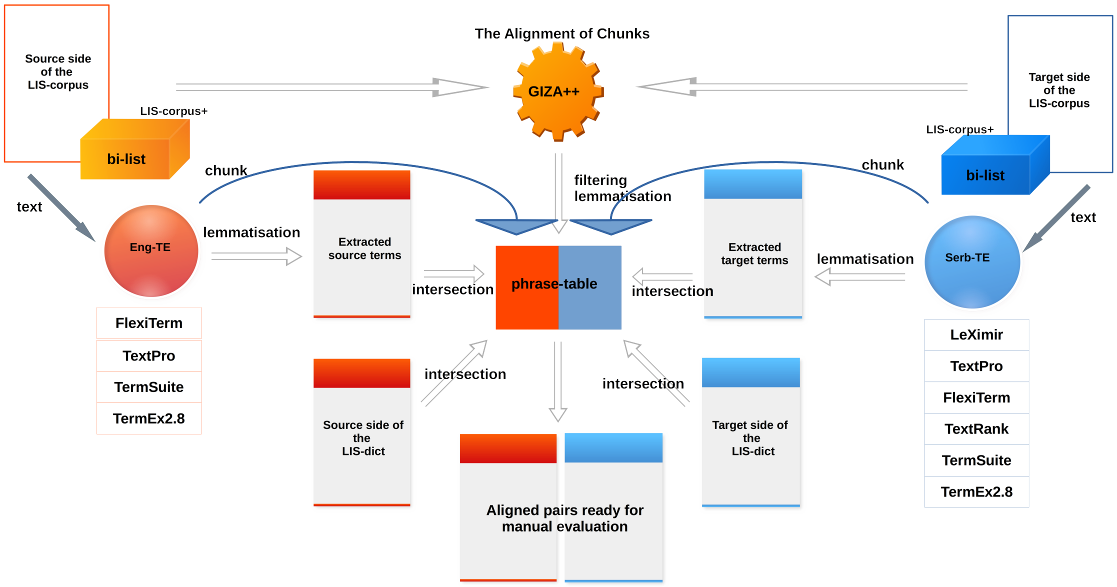


FIGURE 2.4: Diagram of the resources and tools used for terminology extraction for the case of Serbian/English

Aligned domain corpus – LIS-CORPUS

The English/Serbian aligned sentence were derived from the Journal for Digital Humanities *INFOtheca*.[2] Twelve issues with a total of 84 papers were included in the corpus. The papers were either written originally in Serbian and translated to English (61 articles, 73%) or vice versa (23 articles, 27%). Translations were done either by authors themselves, who were experts in the LIS field but not trained translators, or by professional translators who had no specific expertise in LIS. All papers in both languages were proofread.

The main topics of papers published in this journal are represented in Figure 2.5 using the TreeCloud tool for visualisation (Gambette and Véronis, 2010).

The selected papers were aligned at the sentence level resulting in 14,710 aligned segments (Stanković, Krstev, Vitas, Vulović, and Kitanović, 2017).[3] The Serbian part has 301,818 simple word forms (41,153 different), while the English part has 335,965 simple word forms (21,272 different). This means that the average frequency of a word form in the Serbian part is 7, while the average word form frequency in the English part is 15. The major reason for this difference is high inflection, which is a characteristic feature of Serbian language, and which produces many different forms for each lemma. In this work to this resource is referred to as LIS-CORPUS.

[2] INFOtheca, http://infoteka.bg.ac.rs/index.php/en
[3] Available for searching at *Biblisha* site http://jerteh.rs/biblisha/Default.aspx

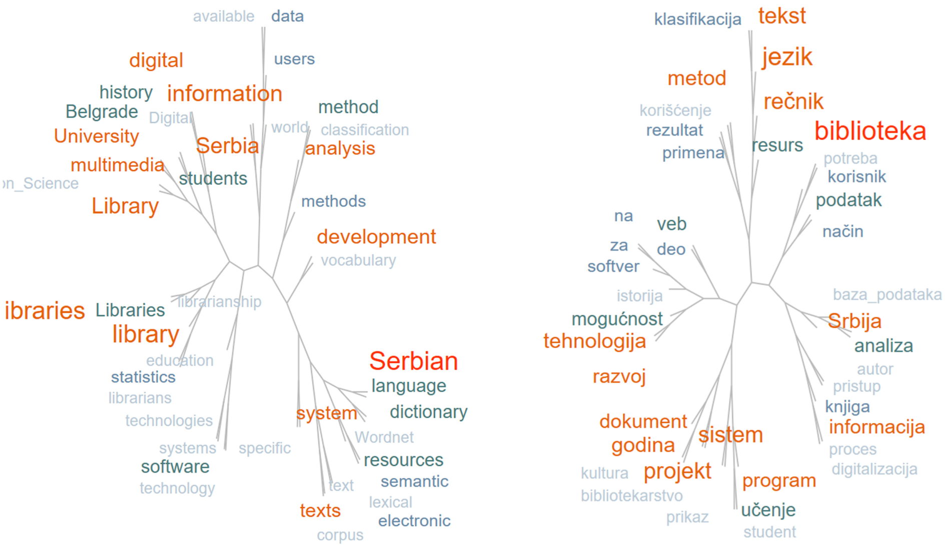


FIGURE 2.5: Main corpus topics represented as Tree Clouds produced from article titles, keywords and abstracts.

Dictionary of Library and Information Science – LIS-DICT

The Dictionary of Librarianship: English-Serbian and Serbian-English (in this text referred to as LIS-DICT) (Kovačević, Injac Malbaša, and Begenišić, 2017) was developed by a group of researchers from the National Library of Serbia. The version of the dictionary that was used for this experiment has 12,592 different Serbian terms (out of which 9,376, or 74%, were MWTs), 11,857 different English terms (8,575, or 72% MWTs), generating a total of 17,872 distinct pairs.[4]

Among distinct pairs, both terms were MWTs in 10,574 (60%) cases, while in 1,923 (11%) cases a Serbian MWT had a SWT equivalent in English, and in 1,070 (6%) cases an English MWT had a SWT equivalent in Serbian. Both terms in a pair were SWTs in 4,305 cases (24%). Among Serbian SWTs, 1,378 (43%) were components of a MWT, while the same was true for 1,245 (38%) English SWTs.

The Extraction of English Terms – ENG-TE

For the extraction of English MWTs, an open-source software tool, FlexiTerm (Spasić, Greenwood, Preece, Francis, and Elwyn, 2013) was selected. It automatically identifies MWTs from a domain-specific corpus, based on their structure, frequency and collocations.

FlexiTerm performs term recognition in two steps: linguistic filtering is used to select term candidates followed by the calculation of a termhood, a frequency-based measure used

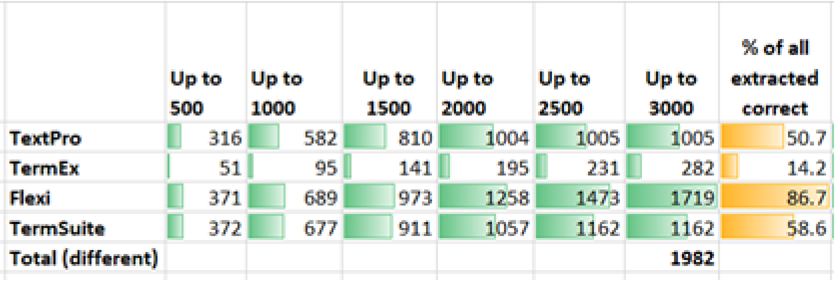
[4] A more enhanced version of this dictionary, presented on the Web, (http://rbi.nb.rs/en/home.html) contains 40.000 entries (approx. 14.000 in Serbian, 12.400 in English and 14.000 in German).

as an evidence that qualifies a candidate as a term. In order to improve the quality of termhood calculation, which may be affected by the term variation phenomena, FlexiTerm uses a range of methods to reduce variation in terms. It deals with the syntactic variations by processing candidates using a Bag-of-Words representation. The system handles orthographic and morphological variations by applying a stemmer in combination with lexical and phonetic similarity measures.

This tool was originally evaluated on biomedical corpora, but in this case it was used for an MWT extraction in the domain of Library and Information Sciences. It was run with default settings and without additional dictionaries.

Three other MWT extractors were also considered for obtaining English MWTs: TextPro[5] (Pianta, Girardi, and Zanoli, 2008), TermSuite[6] (Cram and Daille, 2016) and TermEx2.8.[7]

The results are shown in Figure 2.6. Evaluation performed on the list of terms extracted by all four extractors and evaluated as potential MWU terms showed that FlexiTerm outperformed the other three. Namely, out of 3,000 top ranked MWTs, FlexiTerm recognised 1,719, TextPro 1,005, TermSuite 1,162 and TermEx 289.



	Up to 500	Up to 1000	Up to 1500	Up to 2000	Up to 2500	Up to 3000	% of all extracted correct
TextPro	316	582	810	1004	1005	1005	50.7
TermEx	51	95	141	195	231	282	14.2
Flexi	371	689	973	1258	1473	1719	86.7
TermSuite	372	677	911	1057	1162	1162	58.6
Total (different)						1982	

FIGURE 2.6: Evaluation of different term extractors for English

The positive list of terms was composed for the purpose of evaluation in the following manner: the list initially composed of all manually evaluated extracted terms with addition of all terms from the LIS dictionary, the simple word terms were filtered out, all terms were lemmatised (word by word) and then frequencies were recalculated. The final list for evaluation of all tools consisted of 3000 top ranked terms (by frequency).

In this work, the tool that extracts MWT in English text is referred to as ENG-TE.

The Extraction of Serbian MWTs – SERB-TE

The only system developed specifically for the extraction of MWTs from Serbian texts is a part of LEXIMIR (Stanković and Krstev, 2016), a tool for management of lexical resources. Extraction module of the LEXIMIR system is based on rules, and it relies on e-dictionaries

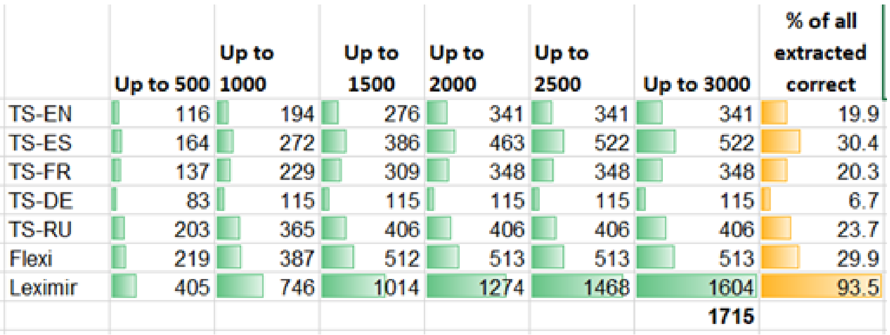
[5] TextPro, textpro.fbk.eu

[6] TermSuite is the Open Source and UIMA-based application drawn out from the European project TTC Terminology Extraction, http://termsuite.github.io

[7] TermEx, allgo.inria.fr/app/termex

and local grammars that are implemented as finite-state transducers (FST) (Stanković, Krstev, Obradović, Lazić, and Trtovac, 2016).

In order to experiment with term extraction for Serbian with some existing tools, the following free and open tools for terminology extraction were examined: TextPro (Pianta, Girardi, and Zanoli, 2008), FlexiTerm (Spasić, Greenwood, Preece, Francis, and Elwyn, 2013), TextRank (Zhang, Petrak, and Maynard, 2018), TermSuite (Cram and Daille, 2016) and TermEx2.8. The results of the evaluation are given in Figure 2.7.



	Up to 500	Up to 1000	Up to 1500	Up to 2000	Up to 2500	Up to 3000	% of all extracted correct
TS-EN	116	194	276	341	341	341	19.9
TS-ES	164	272	386	463	522	522	30.4
TS-FR	137	229	309	348	348	348	20.3
TS-DE	83	115	115	115	115	115	6.7
TS-RU	203	365	406	406	406	406	23.7
Flexi	219	387	512	513	513	513	29.9
Leximir	405	746	1014	1274	1468	1604	93.5
						1715	

FIGURE 2.7: Evaluation of different term extractors for Serbian

FlexiTerm is designed specifically for English, while for TermSuite several modules for different languages were developed: English, Spanish, German, French and Russian. FlexiTerm and TermSuite were compared with LeXimir, the extractor specifically built for Serbian terminology. Evaluation performed on the list of terms extracted by all extractors and evaluated as potential MWU terms showed that LEXIMIR outperformed the other two.

The positive list of terms for evaluation was produced in the same way as for English terms. Out of 3,000 top ranked MWTs, LEXIMIR recognised 1,604, TermSuite for Spanish 522 and FlexiTerm 513 while other TermSuite modules gave poor results. LEXIMIR recognised 93.5% of terms extracted by all extractors and positively evaluated – 1,715.

In the later text, the system for the extraction of MWT in Serbian texts is referred to as SERB-TE.

The Bilingual List of Inflected Word Forms – BI-LIST

With a view to improving the quality of the statistical machine alignment of chunks, a set of aligned and inflected English/Serbian single and multi-unit word forms was used. For this purpose, the following bilingual lexical resources were prepared: 1) Serbian Wordnet (SWN) (Krstev, 2014),[8] which is aligned to the Princeton WordNet (Miller, 1995), and 2) a bilingual list containing general lexica with 10,551 English/Serbian entries. These resources were processed with the tool LEXIMIR.

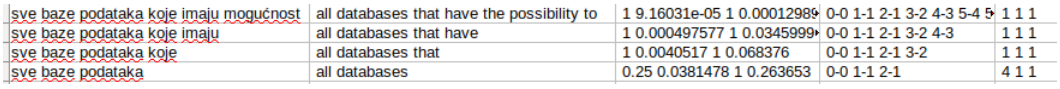
[8] Serbian WordNet can be browsed at http://sm.jerteh.rs/.

The obtained bilingual list of inflected forms contained 426,357 entries. This resource is henceforth referred to as BI-LIST.

The Alignment of Chunks – ALIGN

The first stage of the alignment of chunks was using MOSES as a pre-processing tool (Koehn, Hoang, Birch, Callison-Burch, Federico, Bertoldi, Cowan, Shen, Moran, and Zens, 2007) to perform tokenisation, truecasing and cleaning. In the next step a 3-gram translation model was developed using KenLM (Heafield, 2011), followed by the training of this translation model. For the purpose of word-alignment, phrase extraction, phrase scoring and creation of lexicalised reordering tables, GIZA++[9] (Och and Ney, 2000) were deployed in this order, together with the *grow-diag-final* symmetrisation heuristic (Koehn, Och, and Marcu, 2003).

Aligning with GIZA++ results in the so called "phrase-table". An excerpt from the phrase-table is shown in Figure 2.8.



sve baze podataka koje imaju mogućnost	all databases that have the possibility to	1 9.16031e-05 1 0.00012989	0-0 1-1 2-1 3-2 4-3 5-4 5	1 1 1
sve baze podataka koje imaju	all databases that have	1 0.000497577 1 0.0345999	0-0 1-1 2-1 3-2 4-3	1 1 1
sve baze podataka koje	all databases that	1 0.0040517 1 0.068376	0-0 1-1 2-1 3-2	1 1 1
sve baze podataka	all databases	0.25 0.0381478 1 0.263653	0-0 1-1 2-1	4 1 1

FIGURE 2.8: An excerpt from the phrase-table

Each pair of aligned chunks from the phrase-table contains, among other, information about inverse and direct phrase translation probabilities (the 1^{st} and the 3^{rd} value in the third column in Figure 2.8). These values are later used for the first filtering step, which will be explained subsequently.[10] These numbers are obtained in the following way. GIZA++ reads two input texts in parallel. Whenever two bilingual chunks appear together, their co-occurrence is written into text file (called *f_phrases*). Afterwards, *f_phrases* is sorted in two ways, producing two tables.

Let e be a target chunk, and f source chunk. First, pairs of chunks are sorted so that all source translations of a certain target phrase e are next to each other, as shown in example given in Table 2.4. Therefore, the count of e ("analiza i") is 17.

TABLE 2.4: The first sorting of phrase-table by target chunks

analiza i	analysis and (× 13)
analiza i	and
analiza i	evaluation and
analiza i	the analysis and
analiza i	through evaluation and

Afterwards, pairs of chunks are sorted so that all source language translations of a certain source phrase f are next to each other, as shown in example given in Table 2.5. Therefore, the count of f ("analysis and") is 14.

[9] Statistical Machine Translation toolkit, https://github.com/moses-smt/giza-pp

[10] See more details on how all values are determined at http://www.statmt.org/moses/?n=FactoredTraining.ScorePhrases

TABLE 2.5: The second sorting of phrase-table by source chunks

analysis and	analizirano i
analysis and	analiza i (× 13)

Finally, common occurrences of e and f equal 13. Direct phrase translation probability (the 1st value of the third column, given in Figure 2.8) is calculated as the number of common occurrences of e and f divided by the number of occurrences of f, thus giving $13/14 = 0.9285$ in this case.

Inverse phrase translation probability (the 3rd value of the third column, given in Figure 2.8) is calculated as the number of common occurrences of e and f divided by the number of occurrences of e, thus giving $13/17 \sim 0.7647$ in this case.

Along with these probabilities, original and lemmatised forms of chunks and their frequencies in the original phrase-table, an information about the alignment of content words was preserved, as well. Let's consider the 3$^{\text{rd}}$ example from the Figure 2.8, namely the aligned pair (*all databases that*, *sve baze podataka koje*). An additional (last) column for this pair contains the entry "0-0 1-1 2-1 3-2", which means that the first component (indexed as 0) from the English chunk "all" is mapped to the first component from the Serbian chunk "sve' (indexed as 0), and the next English component "databases" is mapped to the second and third components in the Serbian chunk (since *databases* translates as *baze podataka*), and finally the component "koje" (indexed as 3) is mapped to "that" (indexed as 2). If a mapping contains only a "0-0" value, it means that the English SWT is translated as a SWT in Serbian and vice versa. This information helped in creating a backup of Serbian SWTs that are translated as English SWTs (dubbed SWT-CHUNK). These pairs were eliminated from this list, but kept in a separate file in order to be used for filtering of final results.

The Table 2.6 illustrates one row from the phrase-table for this example.

TABLE 2.6: Direct and inverse probabilities from the phrase-table

e	f	$p(e\|f)$	_	$p(f\|e)$	_	token align	freqs
analiza i	analysis and	0.9285	_	0.7647	_	0-0 1-1	17 14 13

In order to discard aligned pairs that were not correct mutual translations to the greatest possible extent, two filtering steps were added to the proposed approach. In the first filtering step, based on the available information from the phrase-table, aligned chunks that did not have at least one of these probabilities greater than 0.85 were discarded, simultaneously eliminating punctuation marks. Chunks that consisted of punctuation marks and digits only were also discarded.

For the second step, a Bag-of-Words (BoW) representation for English terms from the LIS-DICT was provided, i.e. from ENG-TE, and stop words were removed from it, producing a list mainly populated with content words. Then lemmatised each token from the BoW was lemmatised using the Natural Language Toolkit (nltk) Python library and its WordNet interface.[11] The same simple-word lemmatisation was applied to the English parts of the aligned chunks. Aligned chunks in which the English part did not have at least one lemmatised content word from the BoW list were eliminated.

Henceforth, this suit of tools is referred to as ALIGN.

2.6 Experimental Results and Discussion

Different resources and tools were used in the experiments. In the conducted experiments, each of the three following parameters were combined, all related to the preparation of the input, with the other two, thus obtaining 8 different experimental settings:

1. The input domain aligned corpus (Input i, Section 2.3) consists of:
   - (a) the aligned corpus **LIS-CORPUS**;
   - (b) the aligned corpus LIS-CORPUS extended with the bilingual aligned pairs BI-LIST (**LIS-CORPUS+**);
2. The list of domain terms for the source language (Input ii, Section 2.3) is
   - (a) the source language part of **LIS-DICT** including SWTs;
   - (b) the output of the extractor **ENG-TE** applied to the source language part of the aligned input corpus;
3. The extraction of the set of MWTs in the target language by SERB-TE (Input iii, Section 2.3) was done:
   - (a) on the target language part of the aligned chunks (**CHUNK**);
   - (b) on the target language part of the aligned input corpus (**TEXT**).

As the aligned corpus (Input i, Section 2.3), LIS-CORPUS was used either alone, either augmented with bilingual pairs from the BI-LIST (LIS-CORPUS+). For the extraction of English terms (Input ii, Section 2.3), English side of the dictionary LIS-DICT was used in one series of experiments, and term extractor ENG-TE in the other, while the extraction of Serbian terms (Input iii, Section 2.3) was performed using SERB-TE. The alignment of bilingual chunks (Processing i, Section 2.3) was done by ALIGN.

The input preparation steps as well as processing consists of several components developed in C# and Python that are interconnected to work in a pipeline. It relies on existing tools for the extraction of English MWTs (ENG-TE) and Serbian MWEs (SERB-TE) implemented in LEXIMIR (Stanković and Krstev, 2016) and on GIZA++ for word alignment, while all other components are newly developed.

[11] WordNet interface in Python, `https://www.nltk.org/howto/wordnet.html`

2.6.1 Terminology Extraction

The summary of results obtained by the system for 8 experiment settings is given in Table 2.7.

TABLE 2.7: Results of the proposed term extraction system

	Experiment		A	B	C	B ∧ A	I ∧ C	II→III	II→IV	YIELD
						I	II			
LIS-DICT	LIS-CORP	CHUNK	17,889	240,253	26,719	6,646	1,141	647	173	820
		TEXT			49,632		1,531	770	240	1,010
	LIS-CORP+	CHUNK		496,787	45,813	11,740	2,508	1,105	301	1,406
		TEXT			50,644		2,500	1,075	362	1,437
ENG-TE	LIS-CORP	CHUNK	3,169	215,317	35,226	5,063	2,233	x	x	2,053
		TEXT			49,632		2,233	x	x	2,021
	LIS-CORP+	CHUNK		446,979	44,885	8,164	3,333	x	x	**2,856**
		TEXT			50,644		3,310	x	x	**2,855**

The numbers in the columns represent the following results:

- Input and GIZA++ output results

  A Number of entry pairs in LIS-DICT, i.e. English terms extracted by ENG-TE;

  B Number of lines obtained from GIZA++ phrase table, after pre-processing steps;

  C Number of distinct, lemmatised Serbian MWTs extracted from the target language part of the aligned chunks (for CHUNK) or from the target language part of the aligned input corpus (for TEXT).

- Additional filtering of results obtained by GIZA++:

  I Number of the aligned chunks after initial filtering using English terms (Processing ii, Section 2.3): $(S(align.lchunk) \sim S(term))$, where the list of English terms depends whether the English part was taken from the LIS-DICT, or it was obtained from the corpus by using ENG-TE for extraction.

  II Number of aligned chunks after subsequent filtering using Serbian terms (Processing iii, Section 2.3): $(S(term) \sim S(align.chunk)) \wedge (T(term) \sim T(align.lchunk)) \wedge (S(align.chunk) \leftrightarrow T(align.lchunk))$.

  III Number of new term pairs after filtering, namely those that do not already exist in LIS-DICT – these term pairs were obtained by selecting filtered chunks in which the Serbian part of the chunk does not match a term in the Serbian part of LIS-DICT $((T(align.chunk) \nsim T(term.list)))$ (applicable only when LIS-DICT is used in the experiment);

  IV Number of term pairs after filtering already existing in LIS-DICT – these term pairs were obtained by selecting filtered chunks in which the Serbian part of the chunk matches a term in the Serbian part of $(T(align.chunk) \sim T(term.list))$ (also applicable only for (LIS-DICT) experiments);

YIELD Data in this column represent the number of candidate bilingual term pairs obtained by the respective experiments and prepared for evaluation. Before manual evaluation of these candidates, additional filtering was done.

In order to assess the efficiency of the proposed approach, all extracted pairs were first evaluated manually. To each pair, a label from a set of labels was assigned, which were different for different source language extraction methods:

- For extraction from the English part of LIS-DICT, the following labels were used: LIS – if the extracted pair is correct, that is, the Serbian part of the pair is a MWT that is the translation equivalent of the English term, e.g. *automated service* ≡ *automatizovan servis*; NOK – if the extracted pair is not correct, e.g. *bibliographic description* ≢ *bibliografska obrada* (it should be *bibliographic processing*).
- For the extraction by the English term extractor ENG-TE, the following labels were used: LIS – if the extracted pair is correct and the extracted terms belong to the LIS domain, e.g. *librarianship* ≡ *bibliotečka delatnost*; T – if the extracted pair is correct and the extracted terms belong to some other domain, e.g. *Finite State Machine* ≡ *konačan automat*; OK – if the extracted pair contains translational equivalents, but does not represent a term, e.g. *active sentence* ≡ *aktivna rečenica*; NOK – if the extracted pair does not contain translational equivalents, e.g. *further education* ≢ *u obrazovne svrhe* (it should be *dalje obrazovanje*); X – if ENG-TE extracted neither a term nor a complete noun phrase, e.g. *language white paper*.

When choosing the label LIS for the (LIS-DICT) parameter setting, or LIS, T and OK for (ENG-TE) parameter setting the evaluator took the following approach: the Serbian term in the pair is a correct noun phrase and it is a translation equivalent of the English term in the same pair. It does not necessarily mean that a terminologist, specialist for the domain, would recommend it. For instance, *bibliotečki konzorcijum* would be a preferable term for 'library consortium', but *konzorcijum biblioteka* was labelled as correct, as well.

The evaluation results are summarised in Table 2.8. They show that precision (P) is almost always better when the Serbian term extractor is applied to the target language part of the aligned input corpus (column TEXT vs. column CHUNK, the only exceptions, almost insignificant, can be found in the ENG-TE/LIS-CORP+ setting). The results also show that the use of additional bilingual pairs reduces precision for ENG-TE; however, the number of retrieved pairs as well as the number of acceptable pairs raises significantly (column LIS-CORP vs. column LIS-CORP+). It can be observed that, when LIS-DICT is used for extraction, the ratio of pairs already present in the dictionary (LIS-DICT) and all positively evaluated pairs (LIS-DICT+LIS) is rather stable, ranging from 44.40% for the LIS-CORP+/CHUNK setting to 48.73% for the LIS-CORP/TEXT setting.

Evaluation results showed that a number of new term pairs were retrieved. When LIS-DICT was used for English term extraction, 364 English terms from the dictionary were linked to new Serbian translations yielding 428 new term pairs. One example is the term 'book collection' for which equivalent terms from the dictionary was *kolekcija knjiga*, while

TABLE 2.8: Evaluation results for 8 experiments: number of pairs per labels and the precision measure per combinations of labels

	LIS-DICT				ENG-TE			
	LIS-CORP		LIS-CORP+		LIS-CORP		LIS-CORP+	
	CHUNK	TEXT	CHUNK	TEXT	CHUNK	TEXT	CHUNK	TEXT
LIS-DICT	173	240	301	362				
LIS	182	284	377	413	373	377	430	429
T					611	602	800	799
OK					470	486	728	754
X					76	76	94	90
NOK	465	486	728	662	523	480	804	783
total	820	1,010	1,406	1,437	2,053	2,021	2,856	2,855
P (LIS) %	22.20	28.12	26.81	**28.74**	18.17	18.65	15.06	15.03
P (LIS, T)%					47.93	**48.44**	43.07	43.01
P (LIS, T, OK) %					70.82	**72.49**	68.56	69.42
P (LIS, LIS-DICT) %	43.29	51.88	48.22	**53.93**				

the proposed procedure added *zbirka knjiga*. Likewise, 109 Serbian terms from the dictionary were linked to new English terms, yielding the same number of new translation pairs. For instance, *bibliotečko osoblje* was the translation of 'electronic publication' in the dictionary, while the proposed procedure linked it also to 'electronic edition'.

The next aim was to estimate the recall of the proposed system, since it was not feasible to calculate it exactly. To that end, the overall number of pairs of equivalent terms in the used corpus (the unknown set POSITIVE) were estimated by following these steps:

1. First, English terms from the English part of LIS-CORPUS were extracted, assuming that the English term probably has the Serbian term equivalent in the Serbian part of the corpus. For the experiments that used LIS-DICT for extraction, the English part of the dictionary was used. For the experiments that used the English term extractor ENG-TE, the union of two sets was used. The first set contained all terms extracted from the English part of the used aligned corpus that occurred with a frequency ≥ 3 and that the human evaluator, an expert in the LIS field, evaluated as LIS terms, or terms from some other, close domain. The second set contained all distinct English terms occurring in the evaluated pairs. The union of two sets is denoted by TERM_TEXT and its size by *s_term_text* (the first estimation of the size of the set POSITIVE).

2. Next, an adjustment of the number of SWTs in the set obtained in the previous step had to be made. The adjustment was done only for experiments that used LIS-DICT for extraction, as ENG-TE extracts only MWTs and acronyms. LIS-DICT contains a number of SWTs, of which not many appear in the positively evaluated set, since only pairs in which the Serbian term is a MWT were taken into consideration. The

size of the TERM_TEXT set was reduced, so that the contribution of SWTs in it corresponds to the contribution of English SWTs in the evaluated set of pairs, in terms of their percentage share. Thus, the adjusted size of the set POSITIVE was obtained, denoted by *s_term_text_adj*.

3. Finally, the number of pairs of equivalent terms covered by extracted English terms was calculated, separately for two methods of extraction. To that end, the average number of pairs per one English term in the positively evaluated set was determined.Two adjustment parameters were also calculated, one for LIS-DICT settings and another for ENG-TE settings. These parameters were obtained as the ratio of all different term pairs and all different English terms in all four experiments related to the chosen extraction method.These parameters were applied to the adjusted size *s_term_text_adj* of the POSITIVE sets obtained in previous steps; thus, the estimated size of the sets of all pairs of equivalent terms in the used corpus was obtained $s\_positive = tp + fn$.

In Table 2.9, the results obtained by applying these steps are presented, along with the calculated precision, recall and F_1 score. Recall and F_1 scores were considered as relative since they depend on the source language (English) extraction – the comprehensiveness of LIS-DICT and successfulness of ENG-TE. When calculating these measures, as well as for subsequent results presented in this section, as true positives are considered all those pairs that were marked as LIS, T, and OK. It should be noted that it was sometimes difficult for evaluators to distinguish between LIS and T marks, and that the evaluator of ENG-TE terms and the evaluator of the extracted pairs often disagreed. All of LIS, T, and OK marked pairs are considered as successfully paired terms.[12] For both extraction methods the best results were obtained with LIS-CORP+/TEXT settings – the only exception is the precision which was highest for the ENG-TE/LIS-CORP/TEXT.

Better results for settings using additional bilingual pairs (LIS-CORP+) are expected since significantly more aligned chunks were obtained with their usage (column B in Table 2.7). The application of SRP-TE to the whole Serbian text (TEXT) yielded more extracted terms then its application to Serbian chunks (CHUNK) (column C in Table 2.7) because their application was not impeded by chunk boundaries. The subsequent use of the loose match function produced additional hits among which many were false (see Figure 2.9); nevertheless, the correct hits prevail as the raise in the precision for all experiment settings using TEXT shows.

In Table 2.10, data that offer an insight into the diversity of extracted term pairs, when different parameter settings are used, is presented. Results were grouped by the major parameter, the method of term extraction in the source language LIS-DICT vs. ENG-TE. In these tables only positively evaluated pairs were taken into consideration. The results in Table 2.10 show that the proposed extraction system fared much better when ENG-TE was used for source language term extraction, regardless of the choice of other parameters, $((1206/2248) \cdot 100 = 53.65\%)$, than when LIS-DICT was used $((244/902) \cdot 100 = 27.05\%)$.

[12] The evaluation of ENG-TE terms and of the extracted pairs was done by two different experts. However, the evaluator of the extracted pairs used the results of the evaluation of ENG-TE terms in his work. For that reason the measure of their agreement could not be calculated.

TABLE 2.9: The calculation of the set of equivalent pairs, precision P, recall R and F_1 score

	LIS-DICT				ENG-TE			
s_term_text	2,881				2,185			
s_term_text_adj	1,245							
parameter	1.2794				1.2393			
s_positive	**1,592.89**				**2,707.76**			
	LIS-CORP		LIS-CORP+		LIS-CORP		LIS-CORP+	
	CHUNK	TEXT	CHUNK	TEXT	CHUNK	TEXT	CHUNK	TEXT
TP (true positive)	355	524	678	775	1,454	1,465	1,958	1,982
FN (false positive)	465	486	728	662	599	556	898	873
P %	43.29	51.88	48.22	**53.93**	70.82	**72.49**	68.56	69.42
R %	22.29	32.90	42.56	**48.65**	53.70	54.10	72.31	**73.20**
F_1	29.43	40.27	45.21	**51.15**	61.03	61.96	70.41	**71.26**

Results obtained with different settings of the first parameter (LIS-DICT vs. ENG-TE) were not compared, since comparison of all different term pairs extracted by using these two parameters (902 vs. 2,248, "at least 1" line in Table 2.10) showed very low overlap – only 488 common pairs.

TABLE 2.10: An overlap between results obtained by the use of LIS-DICT and extractor FLEXITERM

	LIS-DICT		ENG-TE	
	pairs	example	pairs	example
exactly 4	244	*bibliografska referenca* 'bibliographic reference'	1,206	*afilijacija autora* 'affiliation of authors'
exactly 3	190	*zaštićeno robno ime* 'trademark'	69	*reč iz korpusa* 'corpus word'
exactly 2	318	*knjižni fond* 'library holding'	855	*kvalitet usluga* 'service quality'
exactly 1	150	*kontrola kvaliteta* 'quality control'	150	*časopis u otvorenom pristupu* 'journal with open access'
at least 3	434		1,275	
at least 2	752		2,130	
at least 1	902		2,248	

Finally, pairs that were marked by the evaluator as NOK were analysed, in order to find the main sources of false pairings, and the results are presented in Figure 2.9. The main reason for rejecting a term pair was that the Serbian term is longer than it should be (label 'long' in the figure legend); for instance, *format elektronskog izvora* ↔ 'electronic resource' (it should be *elektronski izvor*). The other frequently occurring reasons are:

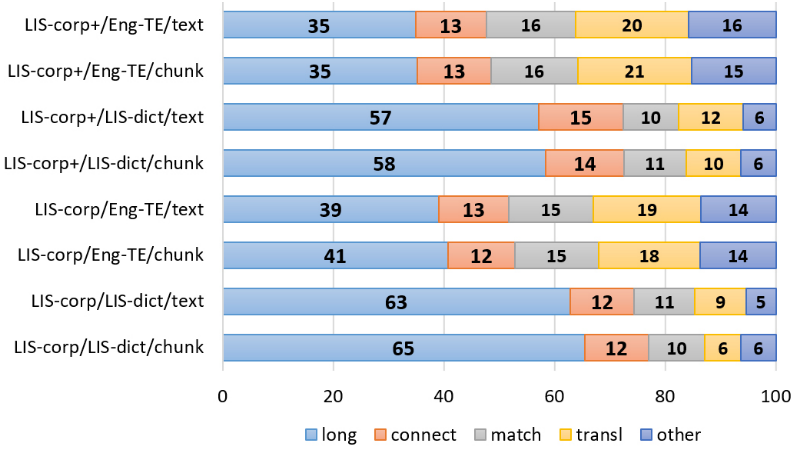


FIGURE 2.9: The distribution of four most important sources of false pairings for various settings

- The match function that was used for comparing extracted terms and aligned chunks was in some cases too loose (label 'match'). For instance *citat u bazi podataka* 'citation in a database' has the same set representation as *baza podataka citata* (the correct translation of 'citation database') and was thus offered as a translation. However, in such cases the correct pairs were often also offered, *citatna baza* and *citatna baza podataka* in this case.
- Translations in texts within LIS-CORPUS were incorrect, imprecise or did not use a corresponding term (label 'transl'). An example of an incorrect translation is 'log file', which was translated in the text as *pristupna datoteka* (a literal translation for 'access file', instead of the proper translation *datoteka izvršenih procesa*): An example of imprecise translation occurred for 'software package', which was translated as *programski paket* (literally 'programming package') instead as *softverski paket*. In the Serbian text '(wide) accessibility' was not translated by a term, but rather by *širi i lakši pristup* (literally, 'wider and easier access').
- In a number of cases chunks were not correctly linked, which resulted in false parings, for instance, *menadžerka kontrole* (literally 'library manager' (woman)) was linked with 'quote' (label 'connect').

The remaining sources of false parings resulted from Serbian terms being shorter than they should be, for instance, *isporuka dokumenata* ↔ 'document supply service' (it should be *servis za isporuku dokumenata*) or when Serbian and English terms overlapped in scope, for instance *objedinjena pretraga* ↔ 'catalog search' (it should be *objedinjena pretraga kataloga* ↔ 'union catalog search'). In a few cases the English term was not a noun phrase and thus could not be captured by SERB-TE, for instance 'not our application'.

2.6.2 Automatic Validation of Bilingual Pairs

The results of the method proposed in Section 2.4 are presented in the following text. The idea was to develop a sequence of steps to be added at the end of the previously described procedure, which would separate correct from incorrect translation pairs. To that end, a RBF SVM classifier is proposed, trained to predict the following two classes: OK for pairs that represent correct translations (positive class), and NOK for the pairs that do not (negative class).

Information on the set of experiments (LIS-DICT or ENG-TE) in which the pair was generated was assigned to each pair, as an additional feature, and after that, all pairs were joined into a single dataset, and all duplicates were eliminated. Eventually, 5,602 pairs were used as samples for the classifier: 2,071 from LIS-DICT (out of which 1,583 were not obtained by ENG-TE), 3,531 from ENG-TE (out of which 3,043 were not obtained by LIS-DICT), and 488 pairs obtained by both LIS-DICT and ENG-TE.

Pairs evaluated as OK, BI, T, as well as term pairs that already existed in LIS-DICT, represented in column IV of Table 2.7, were classified as positive (i.e. they are considered to be good translations), while pairs evaluated as NOK or X were classified as belonging to the negative class. Eventually, this resulted in a dataset with 3,150 positive and 2,452 negative pairs.

As stated earlier, the first pre-processing step is to perform Part-of-Speech tagging of each component of the MWT, regardless of the language. Unitex was used for POS-tagging of Serbian chunks and terms, and the POS-tagger included in the Python's *nltk* module was used for English terms. English was tagged using the universal tagset, while the tagset that Unitex uses for Serbian consists of: N (noun), PRO (pronoun), A (adjective), ADV (adverb), PREP (preposition), CONJ (conjunction), PAR (particle), INT (interjection), NUM (number) and V (verb).

From the pairs with matching POS-tags, a total of 178 features were extracted. The used stylistic, lexical and syntactical linguistic features used for the validation are indicated in Tables A.1, A.2 and A.3, respectively (indicated by X in column V). After feature extraction, all categorical features were automatically encoded to consistent numerical values.

The performance of different supervised classification methods, previously described in Subsection 1.1.1, was compared to the performance of the proposed RBF SVM classifier: Naive Bayes (NB) (Rish, 2001), Logistic Regression (LR) (Hosmer, Lemeshow, and Sturdivant, 2013), Linear Support Vector Machines (SVM) (Joachims, 1998), Random Forests (RF) (Liaw and Wiener, 2002) and Gradient Boosting (GB) (Friedman, 2001). These binary classifiers were trained and evaluated in both 5-fold and 10-fold Cross Validation (CV) settings, using standard classification evaluation metrics: accuracy, F_1, precision and recall. The results for the 5-fold CV setting for different classifiers are displayed in Table 2.11.

As shown in Table 2.11, the best accuracy (78.49%), F_1 score (82.09%) and recall (87.65%) were obtained with RBF SVM classifier, as previously assumed. The motivation behind proposing specifically this classifier was, as shown over the literature, its effectiveness in

TABLE 2.11: Evaluation of different types of classifiers in a 5-CV setting

CLF	ACCURACY %	F_1 SCORE %	PRECISION %	RECALL %
NB	70.05 ± 0.60	73.96 ± 0.51	72.35 ± 0.68	75.65 ± 0.77
LR	78.44 ± 0.60	82.04 ± 0.47	77.16 ± 0.61	87.59 ± 0.55
SVM	61.95 ± 15.24	75.97 ± 5.73	71.19 ± 6.24	64.86 ± 35.17
RBF	**78.49** ± **0.90**	**82.09** ± **0.73**	77.19 ± 0.78	**87.65** ± **0.84**
RF	77.29 ± 0.88	79.96 ± 0.47	**79.73** ± **0.90**	78.25 ± 1.07
GB	78.13 ± 0.79	81.75 ± 0.60	76.98 ± 0.76	87.24 ± 0.84

the higher dimensional spaces, especially in terms of accuracy. Due to a regularisation parameter, over-fitting of these models can be controlled. Finally, with the Radial Basis Function kernel, SVM performs well even if the data is not linearly separable in the original feature space (Vapnik, 1995; Burges, 1998; Joachims, 1998; Vapnik, 1999; Manevitz and Yousef, 2001; Kecman, 2001; Scholkopf and Smola, 2001; Tong and Koller, 2001; Joachims, 2002; Diederich, Kindermann, Leopold, and Paass, 2003).

In Table 2.12, the results for of 5-fold CV for the best classifier, RBF SVM, obtained on pairs from experiments LIS-DICT (2,071 samples) and ENG-TE (3,531 samples), are contrasted.

TABLE 2.12: Comparison of classification metrics for RBF SVM on datasets comprised of pairs obtained from experiments that used LIS-DICT and ENG-TE, separately

EXPERIMENT	ACCURACY %	F_1 SCORE %	PRECISION %	RECALL %
LIS-DICT	**80.15** ± **1.05**	77.92 ± 1.41	75.55 ± 0.94	80.49 ± 2.60
ENG-TE	77.12 ± 1.48	**83.40** ± **1.09**	**77.49** ± **1.10**	**90.30** ± **1.66**

Except for the accuracy, other classification metrics gave better results for the experiment that used ENG-TE. The number of samples in this case was higher, which may explain better precision, recall and F_1 score, since classifiers are prone to over-fitting on smaller datasets (resulting in higher accuracy).

Ten features with highest influence on the classification outcome, according to the Gradient Boosting classifier trained on the whole training set, are displayed in Figure 2.10. Features that have the most influence are the ones related to $S(term)$ (regardless of whether it comes from LIS-DICT or is extracted by ENG-TE): POS-tag of the word at the 2^{nd} position in the MWT, number of different characters, total number of characters and total number of tokens.

Among the top five influential features is also the POS-tag of the word at the 3^{rd} position of the extracted Serbian MWU. The feature that indicates the origin of the pair is ranked as the 109^{th} by relevance for this classifier.

Ten features having the highest Pearson's correlation coefficients with the target label, examined on the whole training set using Weka tool (Witten, Frank, Hall, and Pal, 2016), are displayed in Figure 2.11. The earlier mentioned features that strongly influenced the GB classifier appeared among those most correlated with the target label.

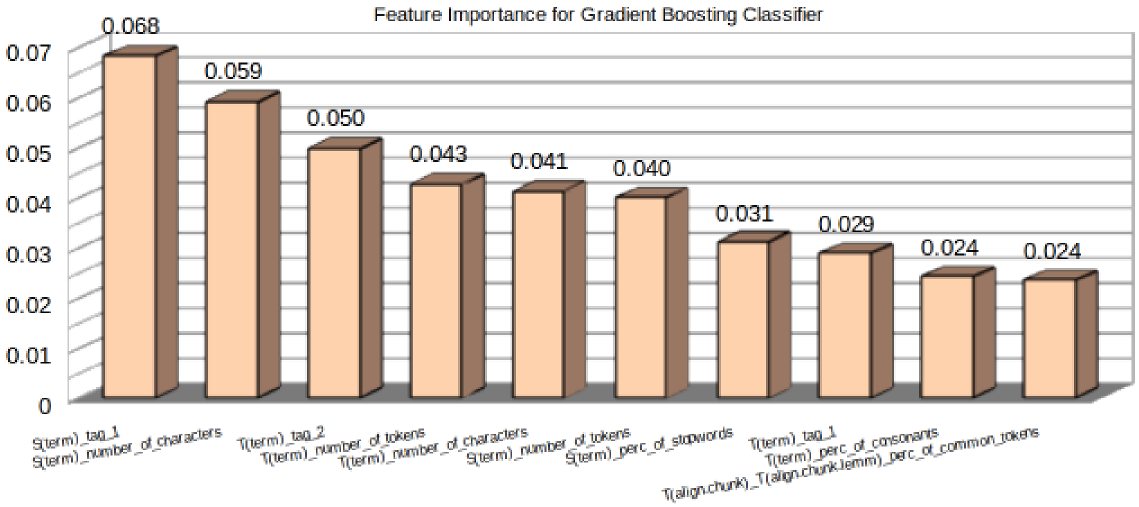


FIGURE 2.10: Feature Importance for GB Classifier in a 5-fold CV setting

It is interesting to take a look at other features that are correlated with the target label, but did not influence the GB classifier, such as the origin feature (source) and percent of lexical diversity of the $S(term)$. Pearson's correlation coefficients of the source feature with other features was determined. The features that had the highest correlations were all related to $S(term)$: number of tokens (0.465), number of characters (0.391), number of different characters (0.389), POS-tag at the 2^{nd} position (0.319) and percent of lexical diversity (-0.312). Since feature selection is a part of GB algorithm, a possible explanation for the fact that the source feature did not show up among the most influential ones is its high correlation with other influential features.

All features were compared, taking into account the origin of the candidate pair. The average, minimum and maximum values of almost all features were consistent, regardless of the origin of the pair. The only exceptions were the maximum number of characters for $T(term)$ (both the original and lemmatised; 62 characters for the pairs obtained after the experiment that used ENG-TE, and 45 for the pairs obtained from the experiment that used LIS-DICT), and the average number of characters in $S(term)$ (12.532 for the case of LIS-DICT, and 17.197 for the case of ENG-TE). It was concluded that ENG-TE yielded somewhat longer English terms, which sometimes resulted in pairing with longer Serbian MWUs.

The goal was to build a language-independent method for the validation of the list of pairs compiled from the presented procedure. Results obtained on this dataset are satisfactory, given that they are obtained on a modest number of samples, and that the method did not use any external language resources (e.g. dictionaries for the validation of translations). It was concluded that it is safe to add the predicted class (positive i.e. OK, negative

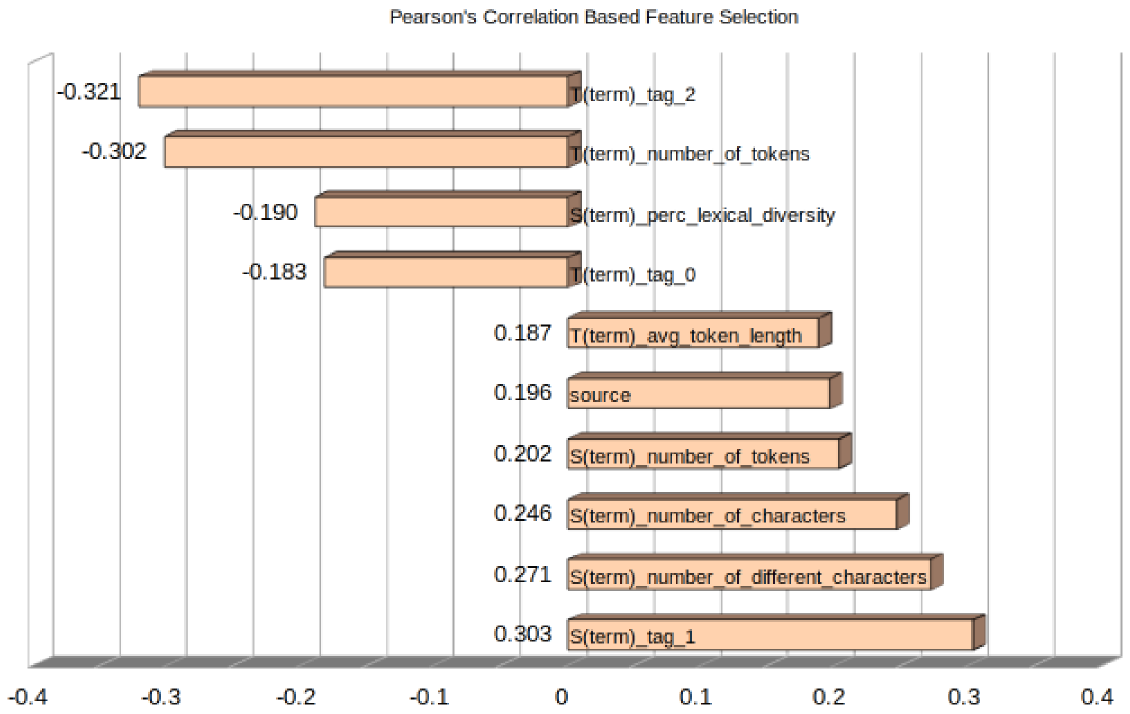


FIGURE 2.11: Pearson's Correlation Based Feature Selection

i.e. NOK) to each pair in the final list, as a suggestion when deciding if a pair should enter a bilingual lexicon or not.

2.7 Concluding Remarks

The obtained results show that for both methods of term extraction from the English part of the aligned corpus the best results were achieved when the corpus was enriched with additional bilingual pairs, and when extraction of Serbian terms was performed on the Serbian part of the aligned corpus, instead of aligned chunks. For the first approach, the F_1 score varies from 29.43% to 51.15%, while for the second it varies from 61.03% to 71.03%. On the basis of the evaluation results, a binary classifier that decides whether a candidate pair, composed of aligned source and target terms, is valid, was built. Different classifiers were trained and evaluated on a list of manually labelled candidate pairs obtained after the implementation of the extraction system. The best results in a 5-fold cross-validation setting were achieved with the proposed Radial Basis Function Support Vector Machine classifier, giving a F1 score of 82.09% and accuracy of 78.49%. As an additional result, the Dictionary of Library and Information Sciences was enriched with $2,474$ term pairs.

With approaches proposed in this work, two goals were initially set: (a) to evaluate the system for the extraction of bilingual multi-word terms by experimenting with different settings; (b) to build a classifier that is going to be able to automatically separate correct term pairs produced by the developed system. The two approaches differ in the way terminology for the source language is obtained: the first relies on an existing domain terminology lexicon, while the second one uses a term extraction tool. For both approaches, four experiments were performed with two parameters being varied.

Finally, the procedure was implemented as a web service, integrated with other applications, and made available as a user friendly interface.[13].

As part of the future work, more experiments with different parameter values are planned. Moreover, the BI-LIST will be enriched with newly produced pairs. The conducted experiments also show that both methods of extraction produce some different pairs of equivalent terms. In the future, not only both methods will be used, when a dictionary for a source language is available, but also terms obtained from several different extractors. In addition, it is intended to introduce more linguistic features, such as multilingual word embeddings (word vectors for the both, source and target side) and some other hand-crafted features tailored for the specific purpose.

Further, development and improvement of the proposed system is intended. The most imminent tasks include: (a) the improvement of the SERB-TE in order to eliminate recognitions that in many cases led to the production of incorrect pairs; (b) experiments with new parameters, such as the recognition of longest matches vs. all matches; (c) experiments with different, more strict, “match” relations between terms and extracted chunks. The most important future research will concentrate on developing methods for reliable distinction between domain specific terms and free noun phrases.

Future research will include application of the same approach to other domains – mining, electrical energy and management – for which aligned domain corpora have already been prepared. Needless to say, the enrichment of sentence-aligned domain-specific corpora, bilingual word lists and monolingual dictionaries of MWTs is the long-term activity.

[13] The corresponding Web application is available at http://bilte.jerteh.rs/, while all the results from different settings can be explored at http://bilteresults.jerteh.rs/. All resources for the classification are also available on-line https://github.com/Branislava/BilTE. For more details about the application, refer to Appendix B

3 Classification of Good Dictionary EXamples for Serbian

This chapter proposes a method that classifies sentences which can serve as the most appropriate examples for common use of a certain dictionary entry. The aim of the presented approach is to support dictionary example selection in order to make the process of composing a dictionary faster and more efficient.

A motivation for proposing a method for automatic dictionary examples selection is explained in Section 3.1, followed by an overview of the related work in Section 3.2. The dataset used as a gold standard, along with other corpora used for the analysis and comparison, is presented in Section 3.3. The same Section contains descriptions of the proposed feature space, justified by a feature distribution analysis of examples from five volumes of dictionary of the Serbian Academy of Sciences and Arts. This analysis is followed by a comparison with distribution in sentence samples extracted from other corpora than the gold one. The research focused on the development of the preliminary model for example selection is presented in Section 3.4. Plans for the future work and some concluding remarks are given in Section 3.5.

3.1 Introduction

Dictionary examples are essential elements in both physical and electronic dictionaries (Gorjanc, Gantar, Kosem, and Krek, 2017). Examples have different roles, some of which are mentioned by Atkins and Rundell (2008):pp. 458–461: they can complement the definition and help a user understand the meaning of a word or a phrase; they should show the typical and natural way of behaviour of a word; and they must be easy to understand – which means that their syntactic structure should be simple and the lexis[1] not too difficult and uncommon. Informativeness and typicality with naturalness are basic criteria for Good Dictionary EXamples (GDEX). Atkins and Rundell (2008):p. 454 also point out that sometimes an entry cannot be understood without the adequate use examples.

However, many experts claim that it is not easy to find good dictionary examples in corpora. Kilgarriff, Husák, McAdam, Rundell, and Rychlý (2008):p. 429 note that reading of concordances is "an advanced linguistic skill", and "the point of reading concordances – to pick up the common patterns a word occurs in – is itself an abstract and high-level task". This task is difficult even for trained lexicographers. In addition, finding good

[1] Lexis refers to the vocabulary of a language.

examples is time-consuming. Nowadays, the electronic corpora can be very big. The number of concordances one gets for a keyword can be so large that it is almost impossible to read all of them. All this was the motivation for the development of GDEX, the tool designed for extraction of good dictionary examples (Kilgarriff, Husák, McAdam, Rundell, and Rychlỳ, 2008), now used not only by lexicographers, but also in language learning and teaching.

A technique that automates the selection process of good example candidates for a dictionary entry can save lexicographers a lot of effort. Tools for identifying good example candidates need criteria when they search for example candidates and these criteria can be based on human input (rules made up by lexicographers), ML input (taught on existing manually produced data), or as a combination of the two. Criteria are determined not only for good example candidates but also for inappropriate ones. It is often easier to describe unwanted features than to define the features that an example should have to be a good candidate for a dictionary example (Kosem, Koppel, Zingano Kuhn, Michelfeit, and Tiberius, 2019).

This chapter proposes a data-driven technique combined with lexicographic expert knowledge, that is aimed at providing support for building different kinds of dictionaries of Serbian language. The motivation was the need for modernisation of dictionary-making process for the Serbian Academy of Sciences and Arts (SASA) dictionary. SASA is still developed traditionally, and its modernisation could help in various directions, such as speeding up the dictionary-making process and development of a lexical database as the source for building new dictionaries of Serbian.

3.2 Related Work

In recent years, various GDEX selection techniques have been developed for several languages. Didakowski, Lemnitzer, and Geyken (2012) implemented a tool for automatic example extraction to assist lexicographers in the development of the Dictionary of Contemporary German. This tool used a rule-based approach, i.e. it searched for examples based on a previously selected set of criteria, such as sentence length, whole sentence form, low sentence complexity, etc., as well as a criterion that all the extracted good examples should exemplify all meanings of the headword. The evaluation showed that a high percentage (95.3%) of extracted examples were deemed acceptable.

Lemnitzer, Pölitz, Didakowski, and Geyken (2015) noted that there were still too many inappropriate examples even in the highest ranked examples per each headword. This study reported on an experiment in which a rule-based approach to good example extraction was combined with ML, using examples once selected by lexicographers as good examples, and the rest as inappropriate ones. The aim was to boost precision at the cost of lower recall by removing as many inappropriate examples as possible (with a risk of losing some good examples as well). Yet, the loss of good examples influenced the final outcome and the results were not that promising. Lemnitzer, Pölitz, Didakowski, and Geyken (2015) identified several possible ways to improve these results, including increasing the number of examples provided by the good example extractor tool, and combining good example extraction with word sense induction in order to limit the good example candidates.

Ljubešić and Peronja (2015) used a supervised Machine Learning approach for extracting good dictionary examples for Croatian on a dataset of 1094 sentences. The examples were annotated as very good, good, inappropriate or very inappropriate. Using this approach, 23 linguistic features were defined and extracted. The evaluation showed approximately 80% precision on the first 10 candidates, and approximately 90% precision on the first three candidates.

Pilán, Volodina, and Johansson (2013); Pilán, Volodina, and Johansson (2014) and Pilán, Vajjala, and Volodina (2016) used Swedish NLP tools and resources to investigate readability and understandability at both document and sentence level based on different linguistic features for language learning purposes. They proposed a rule-based as well as a combination of rule-based and machine learning methods. A set of 61 features, divided into five groups, was used: length-based (e.g. number of tokens and characters), lexical (word-list based), syntactic (dependency relation tags) and semantic features, and features based on Part-of-Speech and morpho-syntactic tags.

When Pilán, Vajjala, and Volodina (2016) classified Swedish texts according to their difficulty level at the document level, lexical features were more dominant (their model obtained the accuracy of 81.3%). When assessing linguistic complexity at sentence level, it was especially useful to use a combination of different features which yielded 7% improvement in classification accuracy. This was confirmed by Pilán, Volodina, and Borin (2016), who focused on language complexity criterion while selecting good examples for language learning exercises.

The Good Dictionary EXamples tool (GDEX) was first implemented as a software module of the Sketch Engine[2] by Kilgarriff, Husák, McAdam, Rundell, and Rychlỳ (2008). Essentially, the GDEX tool is intended to be used for any language and is based on a set of rules that assign a numerical score to each sentence based on its content. The concordances are then sorted descending by their corresponding score. Apart from the sorting, the score can be used to filter out sentences below a certain threshold. This scoring formula constitutes a so-called GDEX configuration. The GDEX configuration usually consists of several classifiers, which judge various features of the sentence, and combines the scores given by these classifiers as a product or weighted sum. Classifiers can be grouped into two categories: hard classifiers are those that include a very high penalty and push the sentence to the bottom of the candidate list, thus acting as a sort of a filter that separates good candidate sentences from the inappropriate ones. Soft classifiers penalise or award bonus points to the candidate sentences, and therefore their importance lies especially in ranking good example candidates. The most common and basic classifiers are universal, i.e. are normally applied regardless of the type of dictionary project (Kosem, Koppel, Zingano Kuhn, Michelfeit, and Tiberius, 2019).

[2] Language corpus management and query system, https://www.sketchengine.eu

3.3 Proposed Method for Classification of Good Dictionary EXamples for Serbian

The proposed method, inspired by previous similar work (Kilgarriff, Husák, McAdam, Rundell, and Rychlý, 2008; Gorjanc, Gantar, Kosem, and Krek, 2017; Kosem, Koppel, Zingano Kuhn, Michelfeit, and Tiberius, 2019), can be described as follows: a set of (headword, sentence, class) triplets should be provided as an input. The following step is the extraction of features. These features are divided into two subgroups: the ones that are dependent of the headword, and the ones that are not. Afterwards, a Decision Tree classifier is trained based on these features. For a new pair of (headword, sentence), the DT decides whether the sentence is a good illustration of use for the given headword or not. This approach is illustrated in Figure 3.1.

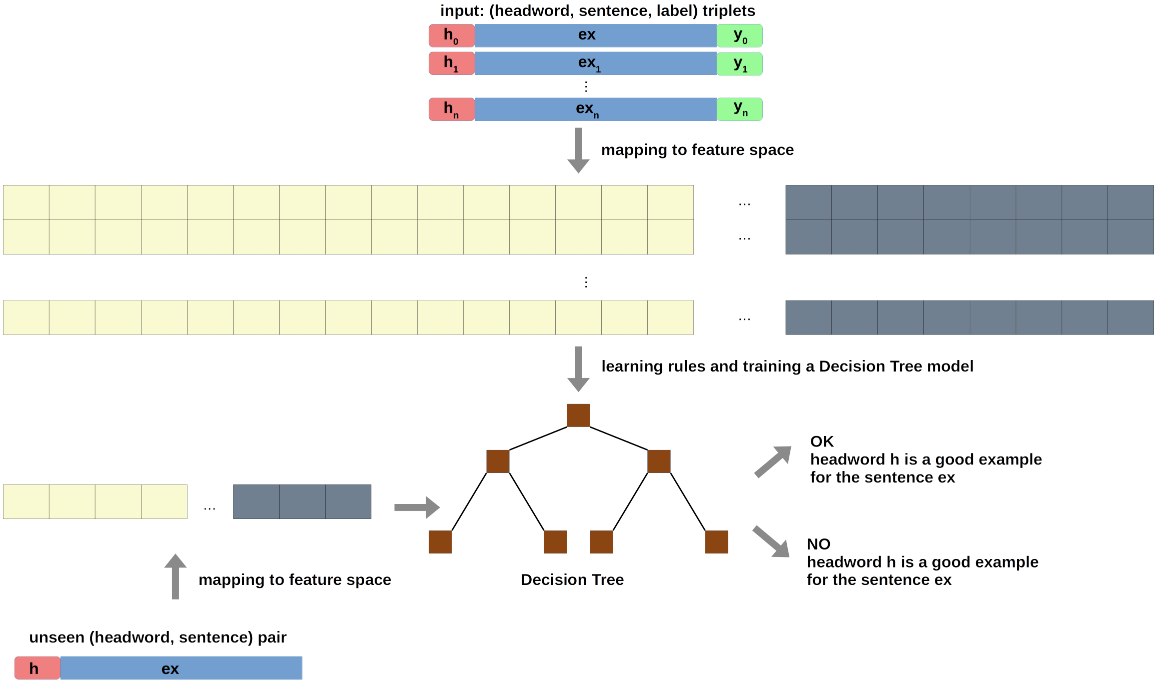

input: (headword, sentence, label) triplets
h_0
ex
y_0
h_1
ex_1
y_1
h_n
ex_n
y_n
mapping to feature space
...
learning rules and training a Decision Tree model
OK
headword h is a good example
for the sentence ex
NO
headword h is a good example
for the sentence ex
mapping to feature space
Decision Tree
unseen (headword, sentence) pair
h
ex

FIGURE 3.1: Proposed GDEX method for Serbian

This method, as initially published by Stanković, Šandrih, Stijović, Krstev, Vitas, and Marković (2019), is based on a thorough analysis of various lexical and syntactic linguistic features in a representative dataset. The feature distribution of examples from this dataset was then analysed and compared with feature distribution of sentence samples extracted from corpora comprising various texts. Afterwards, for further integration into a solution for present and future dictionary production projects, a supervised binary classifier was trained on sentences represented as feature vectors. These sentences contain either standard or non-standard (archaisms, dialects, slang, etc.) Serbian language, which is indicated for each sentence in the SASA dictionary.

The rationale behind proposing exactly Decision Tree classifier is the original solution for the initially set task of Good Dictionary EXamples selection: based on the hand-crafted rules, with thresholds and values observed in the training data. Therefore, it is expected to have implicitly detected patterns and well deduced rules learned by the DT classifier.

The following text describes the steps performed towards the implementation of this method.

3.3.1 Dataset for the Proposed Method

The SASA dictionary (Stijović and Stanković, 2018) covers a high proportion of the vocabulary of Serbian language. It is a combination of the standard- and overall-descriptive dictionary (Zgusta, 1971:p. 212), which means that all marked lexis (dialectal, archaic or dated, jargon, etc.), as well as non-standard phonetic, morphological and syntactic forms are labelled. Each dictionary entry contains (or may contain) several lexical units, along with their descriptive definitions (sometimes definitions by synonyms). Every definition is followed by several illustrative examples (examples are listed chronologically), with precise bibliographic references. The task for lexicographers is to choose 2 to 6 examples from a corpus for each entry, taking into account all previously mentioned criteria for good examples.

Examples may be modified by lexicographers. It is advisable to shorten sentences that are too long, and this kind of intervention is marked by an ellipsis ("…"). It is allowed to omit all irrelevant sentence parts if their presence is not important for the illustration of meaning.

The following is an example[3] from which the irrelevant parts where omitted: "Jednog plavušana …triput sam vraćao u kolonu …", which was originally: "*Jednog plavušana* {, tamo, odnekle s Banije, } *triput sam vraćao natrag u kolonu* {, i opet mi se negdje sakri. Golema ova šikara, zgodno se prikriti. }". This can be translated as: "*One blonde boy* {, somewhere from Bania, } *I returned back to the queue three times* {, and again he managed to hide from me. This forest is huge, convenient for hiding. }".

It is also possible to add insertions: "*Oni* [Talijani] *ti nikako ne vole ove komunce i ove njihove petokrake.*". This can be translated as: "*Them* [Italians] *do not like those communists and their five-stars at all*".

For each example from the five electronic SASA dictionaries, the following list of supporting information was extracted: the volume where it appeared, which headword it explains, headword's Part of Speech (POS), type of editor's intervention on the example (shortening and/or insertion, if any) and a code for the bibliographical source. One example for a headword "peškirče" (eng. 'tiny towel') is given below:

20 | peškirče | N | - | (Petr. E. 4, 82). | Izvadio *peškirče* i obrisao čelo

Which is translated as 'Took out the *tiny towel* and wiped his forehead'.

[3] Example is taken from the book Bašta slezove boje by Branko Ćopić

The size of this gold standard dataset is 133,904 examples, comprising 1,711,231 words or 10,577,723 characters. Within the gold standard dataset, three types of partitioning were used: 1) by published volume (labelled D01, D02, D18, D19 and D20 for volumes 1, 2, 18, 19 and 20, respectively), 2) by type of language (labelled with DSS for standard Serbian and DNS for non-standard Serbian) and 3) by POS of the headword (N – nouns, V – verbs, A – adjectives, ADV – adverbs and X – other).

DSS partition contains sentences in standard language with examples that were not modified by editors. It was assumed that they would be good examples for some future dictionary of contemporary Serbian. DNS contains examples in varieties other than standard Serbian (Church Slavonic, Čakavian, Kajkavian), and lexis marked with labels, such as obsolete, dialect, non-standard, loanwords, slang, etc.

In addition to this gold standard dataset containing dictionary of examples, a control dataset was prepared, derived from various texts, which was used as a sample dataset for dictionary example extraction. The control dataset of example candidates was obtained from the digital library Biblisha[4] (Stanković, Krstev, Vitas, Vulović, and Kitanović, 2017), SrpKor – Corpus of contemporary Serbian (Vitas and Krstev, 2012; Utvić, 2014) and Serbian ELTeC Collection.[5]

For the first collection with contemporary novels (labelled CN), the sentences were extracted from 7 novels written by contemporary Serbian writers and from 7 novels written in German and translated into Serbian (Andonovski, Šandrih, and Kitanović, 2019). In order to represent domain knowledge, two scientific journals (labelled SJ) were used: The Journal for Digital Humanities Infotheca[6] and The Journal of Underground Mining Engineering.[7] The sample labelled DP, with 17 issues of the daily newspaper *Politika* published in 2001–2010 was retrieved from SrpKor (Utvić, 2014). A part of *Serbian ELTeC* was used, which contains 10 complete novels and excerpts from 15 novels that were all published 100 or more years ago (labelled ON for old novels).

The system for Serbian text processing, based on comprehensive e-dictionaries and local grammar in the form of finite-state automata (Krstev, 2008) was used for sentence segmentation. Concordances were extracted using appropriate regular expressions, to serve as candidate examples for corresponding headwords in future volumes. They were bound by sentence delimiters and left/right context of up to 500 characters. The size of the control dataset is 30,104 sentences, comprising 908,980 words or 5,841,700 characters.

3.3.2 Proposed Feature Space

The method proposed in this work is based on the automatic analysis of various linguistic (lexical and syntactic) features of the gold standard examples of use. For the feature

[4] Biblisha, http://jerteh.rs/biblisha/

[5] Distant Reading for European Literary History (COST Action CA16204) https://distantreading.github.io/ELTeC/srp/index.html

[6] Infotheca, http://infoteka.bg.ac.rs/index.php/en/infotheca

[7] The Journal of Underground Mining Engineering, http://www.rgf.rs/publikacije/PodzemniRadovi/

distribution analysis, lexical and syntactic features extracted are listed in Tables A.2 and A.3, respectively (indicated by X in column X). The initial set of features was inspired by Kilgarriff, Husák, McAdam, Rundell, and Rychlỳ (2008); Gorjanc, Gantar, Kosem, and Krek (2017), and guided by the overview of the features performed later by Kosem, Koppel, Zingano Kuhn, Michelfeit, and Tiberius (2019). The motivation for this analysis was to explore different distributions of various parameters and the dataset's homogeneity over various criteria.

Detailed analysis of feature distributions is given in Appendix C. The main results of this analysis are the following. First, examples for adjectives and nouns are longer than those for adverbs and verbs. Then, the sentences in the control dataset partitions are longer than in any volume of the SASA dictionary. Similarly, the dispersion for contemporary novels (CN) is the highest, and the average length of sentences in journals and daily papers is similar. Also, old novels (ON) have shorter sentences than the contemporary ones (CN). Sentence length is an important feature that can tell a lot about clarity of an example. Usually, the shorter ones are less informative, whilst the ones that are long have to be truncated, they are harder to understand, take a lot of space, etc. Further, dictionary examples have less punctuation marks than control dataset, as expected. The average word length is similar for all dictionary volumes, slightly shorter for novels and much longer for daily papers and even more for journals. One reason for this could be due to the use of specific terminology in scientific journals. The texts scraped from Web can be sometimes incorrectly processed, so that words turn up to be "glued" together, which can be one possible explanation for the longer words. Average word length is an important feature, because longer words can negatively impact readability of a sentence.

It can also be seen that sentences in novels contain more pronouns than examples in SASA dictionary. The first two volumes have a very low median, which corresponds to the lexicographers' practice to choose examples with nouns because they are easier to understand. Sentences extracted from daily papers and scientific journals also have very few pronouns, which can be explained by a greater need for precision in scientific and journalistic language.

In order to approximate and predict the ability of a user (with a specific profile) to understand a specific example, a "frequency indicator" was calculated for each example/sentence. This feature (feature *avg_freq_in_corpus* in Table A.2) was determined as an average frequency of each word in the reference corpus. The underlying assumption is that users will more easily understand examples that use more frequent words. Word frequencies were obtained from SrpKorp2013 (Utvić, 2014). It can be seen that novel examples have higher frequency indicators, while these indicators for journal examples are lower. The first two volumes of SASA dictionary have a wider span of frequency indicators than other volumes, as expected, due to the type of the lexis contained in each volume (for example, the majority of the lexis beginning with an *a*, contained in the first volume, is of foreign origin, while the second volume contains lexis mostly labelled as regional, obsolete, ephemeral etc.).

Examples in standard (DSS) and in non-standard (DNS) Serbian in the dictionary have a similar distribution of the number of words in the examples, which means that there is no difference in this respect between good examples illustrating standard or non-standard lexis. On the other hand, the evaluated dataset has a wider distribution for inappropriate examples (DNS (NO)), while similar distribution with those in the dictionary. Results for other features also show that there are no significant differences between examples in DSS and DNS.

Distribution of values for features *sentence_length*, *avg_token_len*, *perc_pronouns* and *avg_freq_in_corpus* from Table A.2, over five partitions of the SASA dictionary, is listed in Table 3.1.

TABLE 3.1: Values distribution for features *sentence_length*, *avg_token_len*, *perc_pronouns* and *avg_freq_in_corpus*

		Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
sentence_length	D01	6	46	66	71	90	88
	D02	6	47	65	70	88	88
	D18	7	55	77	82	103	264
	D19	6	55	77	84	105	292
	D20	6	57	81	88	111	543
avg_token_len	D01	0.0	4.2	4.8	4.9	5.5	14
	D02	0.0	4.2	4.7	4.9	5.4	16
	D18	1.0	4.3	4.8	4.9	5.5	15
	D19	0.0	4.2	4.8	4.9	5.5	16
	D20	2.3	4.3	4.8	4.9	5.4	16
perc_pronouns	D01	0	1	1	1.8	3	15
	D02	0	1	1	1.7	2	15
	D18	0	1	2	2.2	3	17
	D19	0	1	2	2.0	3	15
	D20	0	1	2	2.2	3	23
avg_freq_in_corpus	D01	0	1557	3201	3275	4707	15490
	D02	0	1553	3162	3270	4679	16815
	D18	0	2079	3495	3552	4922	15490
	D19	0	1948	3353	3433	4760	15714
	D20	0	2090	3458	3533	4848	15490

Distribution of values for features *sentence_length* and *avg_token_len* from Table A.2, by Part-of-Speech of the headword, over five partitions of the SASA dictionary is given in

Table 3.2.

Histograms and boxplots were supported by a data summary of calculated features, which offered the guidelines for data cleaning and control dataset preparation. Preprocessing of both used datasets was performed and data summaries were provided. They were analysed by lexicographers, on the basis of which parameters for potential examples cleaning were deduced and threshold values for them were defined.

Table 3.3 presents the data summary from SASA Dictionary for five representative features. The conclusions are in line with the default thresholds proposed by the SketchEngine itself.[8] For example, the optimal sentence length interval proposed is between 10 and 14 characters, which agrees with the findings for Serbian as well (40th and 65th percentiles).

TABLE 3.2: Distribution for features *sentence_length* and *avg_token_len* by Part-of-Speech of the headword

		Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
sentence_length	A	6	55	75	82	101	292
	ADV	10	46	66	72	92	265
	N	6	54	75	81	102	543
	V	7	47	68	74	94	276
	x	6	42	62	68	86	263
avg_token_len	A	1.5	4.2	4.7	4.8	5.4	10.7
	ADV	1.7	3.8	4.4	4.5	5.0	10.5
	N	1.0	3.9	4.4	4.5	5.0	20.0
	V	1.5	3.7	4.2	4.3	4.8	11.0
	x	1.2	3.2	3.8	3.9	4.5	8.2

3.4 Results and Discussion

The earlier Sections described the initial steps toward developing modules for detection of good examples as well as for detecting those that are not appropriate examples for standard language use. Further filtering and ranking of examples is performed using rules obtained from the analysed data (feature vectors) combined into a single score, which is explained more in the text below.

[8] SketchEngine GDEX configuration files,
https://www.sketchengine.eu/syntax-of-gdex-configuration-files/

TABLE 3.3: Data summary from SASA dictionary for selected features

Percentile	Sent. length	# words	Avg. word length	# stop-words	# UCase inside sent.
5^{th}	28	5	3.6	0	0
40^{th}	64	10	4	3	0
Median	73	12	4.8	4	0
65^{th}	87	14	5.2	5	0
95^{th}	150	25	6.6	10	2

The development of the GDEX function is inspired by the state of the art SketchEngine implementation for which offers the following functions: *blacklist()*, *greylist()* and *optimal_interval()*. For each feature the function *optimal_interval* uses four key percentiles from the gold SASA dataset (earlier determined in Table 3.3), where feature values lower than the first and higher than the last are assigned a score of 0.01, in the median scores are 1, and between them a linear interpolation function was used. The four percentiles were computed for different key values. For the *greylist()* function, only two key values were used (5^{th} and 95^{th} percentiles): values lower of the 5^{th} are assigned a score of 1, higher than 95^{th} a score of 0, and between them linear interpolation is used.

In addition to the solution with multiple assessments of features, each feature value was converted to a numerical value from 0 to 100 and a numerical weight (priority) was assigned to it (the sum of all weights being 1), which yielded better results. The precision calculated on evaluation set: it was 0.77 for the first 100 ranked examples, 0.70 for the first 200, 0.65 for the first 400, 0.6 for the first 1000 etc. Besides, it was noted by the evaluators that the results can be improved with additional hand-crafted rules: for example, if the adverb of time or place is not the headword to be illustrated by the example, sentences beginning with these adverbs are not good examples, because they often need the preceding context (such as *Tada je progovorila* which means "Then she started speaking"). Yet, these findings are not regarded as final. The final results will be available after more extensive evaluation.

Sentences were ranked by a GDEX weighted sum of feature score values, which was then mapped to a user friendly final score from 1 (poor, lowest 20%) to 5 (good, highest 20%), representing their suitability to serve as examples. These sentences, represented as feature-vectors, were used as the dataset for different supervised Machine Learning models, which was then used in a GDEX classifier for contemporary Serbian sentences. Since the dataset of examples was unbalanced, with DSS examples twice as much sa DNS examples, 44,808 (out of 89,096) examples with standard lexis from DSS dataset were randomly extracted and labelled as 'OK' (positive class) and the same number of examples (44,808) from DNS set with non-standard lexis (labelled as 'NO' – negative class). Manually evaluated sample, being small, was replicated 5 times, yielding 7,165 'NO' and 6,585 'OK' examples.

The first step before training a classifier is to analyse and select features. The full list of the examined lexical and syntactic linguistic features is indicated with character X in the column G in Tables A.2 and A.3, respectively. Pearson correlation matrix that represents correlation of features to manually assigned class labels was determined and visualised, where green represents a strong positive correlation, red a strong negative correlation, and yellow no correlation. After removing irrelevant features (those that have very low correlation with class label, like avg_word_len, or those that are highly correlated with each other, such as max_word_len and max_token_len), we represented each sample with the shorter feature vector (Figure C.11).

In order to assess the performance of the proposed classifier in comparison to some other classifiers, the gold standard dataset was split into a training and test set (80% and 20% of the dataset, respectively). Several classifiers for the both, positive and negative classes, were examined. The evaluation results (precision, recall, F_1 score in favor of both classes) are given in Table 3.4.

TABLE 3.4: Classification results for different ML models and parameter values

CLF	P+	R+	F+	P-	R-	F-
kNN	.6	.59	.6	.6	.62	.61
DT	.82	.83	**.83**	**.83**	.82	**.83**
RF	.84	.66	.74	.72	.88	.79
Ada	.85	.76	.81	.79	.87	**.83**
NB	.75	.25	.37	.55	**.92**	.69
LR	.84	.68	.75	.73	.87	.79

Given the results displayed in Table 3.4, the Decision Tree classifier, as expected, gives the highest and the most stable values for F_1 score for both classes. Out of 11,056 negative samples in the test set, 9,212 were classified as negative (83%, true negative), and the remaining ones as positive (17%, false positive). From 11,180 positive samples, 9,190 were classified as positive (82%, true positive), and the remaining ones as negative (17%, false negative).

One of the main advantages of the DTs is that they are simple to understand, due to the simple concept behind them. Consequently, they are easy to visualise and interpret. One important aspect is also that the feature selection is implicitly included in the method itself. The nature of the data is irrelevant, as well, since DTs can operate with both, numerical and categorical data.

The feature extractor is freely available,[9] while the GDEX ranking and the trained DT model is available for authorised users. The future system for semi-automatic identification of good dictionary examples implies the development of more modules, e.g. a user interface for feature extraction and for GDEX parameters fine tuning, but the evaluation of first results of the developed core components is encouraging.

[9] For more details, refer to Appendix C

3.5 Concluding Remarks

This chapter proposed a method for selection of Good Dictionary EXamples for Serbian, which is based on a detailed analysis of various lexical and syntactic features. The initial set of features, inspired by a similar approach for other languages, was extracted from a corpus compiled of examples from the dictionary of Serbian Academy of Sciences and Arts. Next, the resulting feature distribution was compared with the feature distribution of sentence samples extracted from other textual sources. Finally, based on these features, a binary Decision Tree classifier was trained to predict whether an example sentence contains standard or non-standard Serbian language.

This work made custom SketchEngine GDEX configuration for Serbian possible (Zingano Kuhn, Dekker, Šandrih, Zviel-Girshin, Arhar Holdt, and Schoonheim, 2019; Dekker, Zingano Kuhn, Šandrih, and Zviel-Girshin, 2019). Improvement of weighted measure of features will follow, with a combination of expert knowledge and data training results.

Another aim is to develop a Web service that will implement a wider set of features and criteria for the flexible selection of GDEX parameters. Full system integration will combine the use of lexical database with corpora exploitation via the developed Web service and software. Since the work on digitisation of other volumes of SASA dictionary is in progress, it is expected that larger data would contribute to more conclusive results.

The trained classification model can be further improved. It makes sense to expand the existing feature set with new features, to add more sample, or to experiment with the state-of-the-art Neural Network architectures. Another future step is the evaluation of a model on a control dataset. The performance of extraction and ranking is going to be evaluated by more expert evaluators, parallel evaluation and inter-rater agreement computed. It is also intended to introduce flexible mapping of computing scores (from worst to best) and to score the examples using them, e.g. based on Linear Regression. Finally, the next major objective is to to expand the proposed approach for the extraction of good examples for bilingual English/Serbian dictionaries, based on parallel corpora.

4 Authorship Identification of Short Messages

This chapter asks the following question: is it possible to tell who is the sender of a short message, by just analysing a writing style of the sender, and not the meaning of the content itself? If possible, how reliable would the judgment be? Are we leaving some kind of a "fingerprint" when we text, and can we tell something about others based just on their writing style?

The motivation for this research is explained in Section 4.1, while Section 4.2 outlines previous related work. The proposed method, including the description of the proposed feature space for Authorship Identification in short messages is presented in Section 4.3. The steps for creating two classifiers and a feature analysis are described in Section 4.4, followed by a discussion of the obtained results. Finally, conclusions and plans for the future work are given in Section 4.5

4.1 Introduction

Exchange of short messages is one of the most popular communication styles in present times. Many researchers focus their work on analysing datasets obtained from Twitter or Facebook. On the contrary, not so many papers have been dedicated to analysis of SMS messages. This is probably due to personal nature of these messages, which makes obtaining a dataset with the size comparable to the size of datasets retrieved from micro blogging services a hard task. This is somehow paradoxical, since SMS messages are one of the oldest and the most used forms of digital communication.

Any analysis of SMS messages is challenged with consequences of some specific circumstances. Firstly, single message is restricted to the length of 160 characters. Therefore, SMS messages often do not contain enough information for the analysis of their meaning. This is in addition to the many spelling errors and typos. Nowadays people mostly use post paid contracts with mobile network operators and therefore can concatenate and send many messages instead of one, but they still tend to write very short messages. One of the potential reasons of this brevity could simply be a consequence of an old habit. For the most part, SMS messages lost their popularity in favour of applications for instant messaging.

Another specificity can be seen when people text using a language with diacritics. For example, Serbian language uses two alphabets (Latin and Cyrillic), and Latin letter contains five diacritics (č, ć, đ, š, ž). A single SMS message can contain maximum 140 Bytes. Standard ASCII characters are coded with 1B, and diacritics with 2B. This also applies to messages in Cyrillic. As a consequence, people usually omit the use of diacritics. Since electronic language tools contain words and their lemmas written in their correct forms, i.e. with diacritics, these tools cannot be applied to SMS datasets, without some previous step of diacritics restoration.

It is nothing unusual that one just sees a message and knows the sender, without even checking the message header. Despite the missing signature, voice, mimics, sound and so many other components that written and oral communication contains, just by usage of emoticons, abbreviations, specific typos, grammar mistakes or specific use of punctuation — one can often correctly guess who is the corresponding sender. This is primarily true for people with a specific writing style. In the case of a very short message, e.g. "Will you be on time?", determination of the sender can become more difficult. The task is not easy at all even for humans, especially when there is no other information such as mobile phone model of the sender, operative system the sender uses, location etc.

In this work, different classifiers were trained and compared in order to predict the message sender, based on SMS representation in the vector space of different linguistic (lexical, stylistic and syntactic) features. All resources used in this research (table of extracted features, Python module for feature extraction and code for model training and evaluation) are available on-line.[1] Since the additional validation dataset was not available, the performance estimation is done by using 5-fold cross validation (CV).

4.2 Related Work

Authorship Identification (AI), regarded as a part of User Profiling (Rangel, Rosso, Koppel, Stamatatos, and Inches, 2013; Rangel, Rosso, Potthast, Stein, and Daelemans, 2015; Rosso and Rangel, 2020), is a prominent research field. Most of the work done so far was related to the semantic analysis of the content (Pennebaker and King, 1999; Mairesse, Walker, Mehl, and Moore, 2007). Concerning AI, another approach in solving the task of automatic recognition of the given text's author is by observing *stylometric* cues (Oakes, 2014). These stylometric features, as Roffo, Giorgetta, Ferrario, and Cristani (2014):p. 33 name them, include *lexical* (counts of words and characters in text) and *syntactic* (punctuation and emoticons) features. After extraction of these features, they are typically used with discriminate classifiers, so that each author represents one class.

[1] Github repositorium, https://github.com/Branislava/sms_fingerprint

A survey about application of AI to Instant Messaging (IM) was conducted by Stamatatos (2009). Zheng, Li, Chen, and Huang (2006) used stylometric features with Support Vector Machine (SVM) and Artificial Neural Network (ANN) classifiers, while Abbasi, Chen, and Nunamaker (2008) applied dimensional reduction for AI on corpora containing E-mails, IM, feedback comments and even program code. Similar work on AI in IMs was also conducted by Abbasi and Chen (2008).

Orebaugh and Allnutt (2009) identified participants within IM conversation by observing sentence structure and usage of special characters, emoticons and abbreviations. The writing style of individuals was the focus of Roffo, Giorgetta, Ferrario, and Cristani (2014). Authors analysed whether special interactional behaviour, as the one present in the live communication, can emerge in chats. They also studied if certain personality traits affected writing style. Authors concluded that some characteristics significantly influence chatting style and that some of them can be very effective with identifying a person among diverse individuals.

Similar research was performed by Eckersley (2010) and Laperdrix, Rudametkin, and Baudry (2016). These authors were more concerned with determining how traceable certain computer configuration was, based on Web browser version, the underlying operating system, the way emojis were displayed within a Web browser, etc.[2]

4.3 Proposed Method for Authorship Identification of Short Messages

The proposed method can be described as follows: as an input, a dataset of (author, SMS) pairs should be provided. The following step is the extraction of features. Afterwards, a Gradient Boosting classifier is trained based on these features. For a short message unseen during the training phase, the GB model decides who is the author of a message from the list of authors seen in the training set. For the initial experiment, the classification task was relaxed in the following way: all messages from one author are selected and marked as the ones belonging to the positive class. The rest of the messages are considered to belong to the negative class. This way, the classifier's task is to determine whether a message is or is not written by a certain author.

This approach is illustrated in Figure 4.1.

The first answers to the research question asked in this Chapter were offered by Šandrih (2018). For this purpose, a dataset of $\sim$ 5,500 SMS messages was extracted from the mobile phone of one person and two gradient boosting classifiers were built: the first one is trying to distinguish whether the message was sent by this exact person (mobile phone owner) or by someone else; the second one was trained to distinguish between messages sent by some public service (e.g. parking service, taxi, bank reports etc.) and messages sent by humans.

[2] The website dedicated to studying the diversity of browser fingerprints and providing developers with data to help them design good defenses, `https://amiunique.org/`

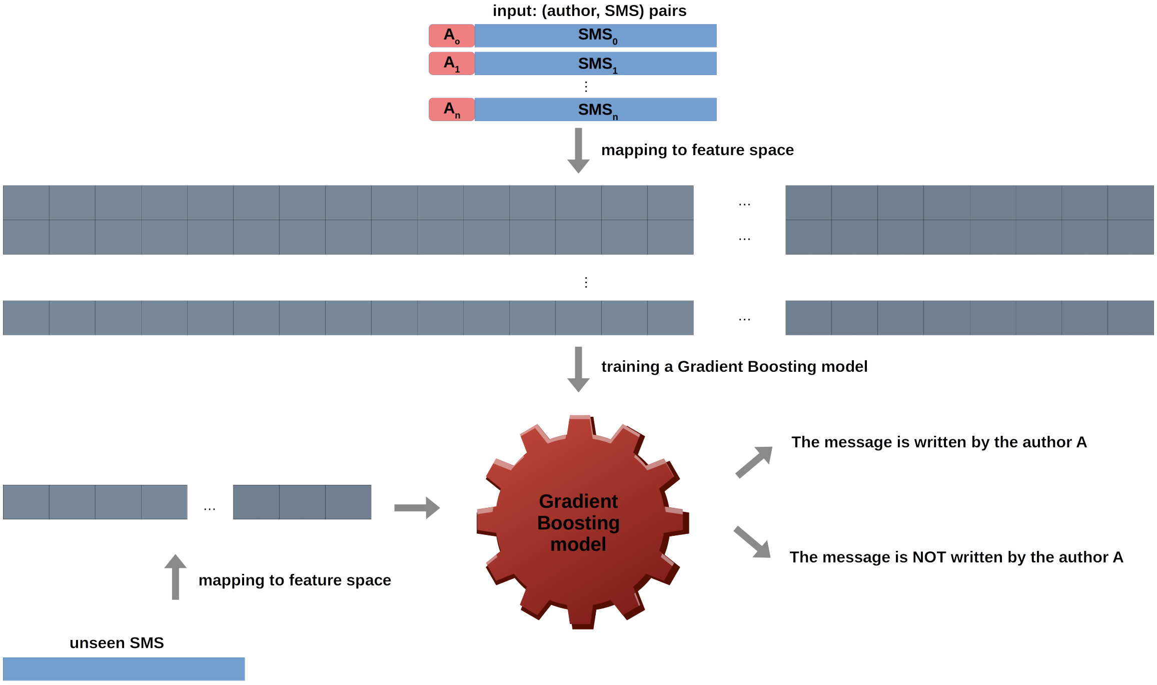

input: (author, SMS) pairs
A_0
SMS_0
A_1
SMS_1
A_n
SMS_n
mapping to feature space
training a Gradient Boosting model
Gradient Boosting model
The message is written by the author A
The message is NOT written by the author A
mapping to feature space
unseen SMS

FIGURE 4.1: Proposed method for identification of authors in short messages

For this approach, a Gradient Boosting algorithm is proposed. In general, boosting refers to a technique that, in each iteration, trains a model based on the output of the model trained in the previous iteration. This way, the trained model improves with each new iteration. Gradient boosting, specifically, is based on Decision Trees. DTs are added one at a time and a gradient descent procedure is used to minimise the loss that occurs when a new DT is added. After calculating error or loss on the dataset, the outcomes predicted correctly are given a lower weight and the ones miss-classified are weighted higher. These steps are repeated until the best instance weights are found.

4.3.1 Proposed Feature Space

The following are two messages from the used dataset, written in different alphabets. Both messages contain the same text, namely “What’s up?”. The first message contains informal dialect-specific greeting that can be observed by use of repeated letters and an emoticon. The second message is written in Cyrillic, normally less used in informal communication. Attribute *type* represents whether the message was sent (value 1) or received (value 0).[3]

*<sms address="+381687457***" type="1" contact_name="Mar***"
readable_date="18.11.2017 5:20:26" body="Sta imaaaaaaaaaaa :-)" />*

[3] Real names and telephone numbers are changed to preserve anonymity.

<sms address="+381600854***" type="0" contact_name="Дав***"
readable_date="14.02.2016 20:24:45" body="Шта има?" />

Various linguistic features were extracted from the *body* attribute of `<sms>` elements. The full list of the proposed set of stylistic, lexical and syntactical linguistic features is given in Tables A.1, A.2 and A.3, respectively (indicated by X in column A).

The list of lexical features (see Table A.2) includes the following character-based features: count of characters, count of Cyrillic characters, count of diacritics, count of umlauts (as minority of messages are in German, later explained), count of uppercase characters, count of lowercase characters, count of digits, count of alphabet characters, count of exclamation marks, count of question marks, count of dots, count of commas and the total count of present punctuation.

Sixteen additional lexical features were added as a ratio of already mentioned feature counts. These are: 1) the ratio of exclamation marks/question marks/dots/commas/-total punctuation/alphabetic characters/diacritics/umlauts/Cyrillic/uppercase/lowercase/digits and the total number of characters; 2) the ratio of upper and lowercase characters; and 3) the ratio of punctuation/Cyrillic/digits and alphabetic characters. A part of the Python code that generates the list of lexical features is given in Listing D.2.

The list of proposed syntactic features (see Table A.3) can be divided into two categories: emoticons and abbreviations. Emoticons have been useful in many research topics, such as sentiment analysis (Read, 2005; Škorić, 2017) or for interpreting short messages (Walther and D'Addario, 2001; Derks, Bos, and Von Grumbkow, 2007). One hundred and two different emoticons were listed and classified into nine groups: 1) emoticons that represent a smile (smiley), 2) emoticons that have a happy face (happy), 3) sad, 4) surprised, 5) kissing, 6) winking, 7) tongue, 8) skeptic, and 9) other facial expressions. The full list of emoticons along with their corresponding regular expressions is given in Listing D.3. In this specific dataset, not all emoticons from Listing D.3 are present. The ones that were missing were discarded during pre-processing phase, resulting in final list of 34 emoticons. They are represented with corresponding regular expressions:

smiley `:-) ;) :) ({2,}: (:`

happy `xD{2,} xD :D{2,} :-D{2,} :D`

sad `:( :-({2,} :-( :({2,} :-’( :-’({2,}`

surprised `:o :-o`

kiss `:* :*{2,} :-* :-*{2,}`

wink `;-) ;){2,} ;-){2,}`

tongue `:-p{2,} :p{2,} :-P{2,} :-P{2,}`

skeptic `:/{2,} :/`

misc `=D =] 8-)`

An absolute count of each emoticon appearance per message was added as a single feature. Afterwards, additional nine features were added as aggregated count of each emoticon type (e.g. total number of smiley emoticons, total count of all happy emoticons in a message etc.). The full list of emoticons is available on-line.[4]

Abbreviations are very common in the context of writing short messages, and therefore a list of total one hundred and thirty five different abbreviations was made. Some of the proposed common abbreviations in texting are: *ae* (hajde - "come on"), *dog* (dogovoreno – "deal"), *dop* (dopisivati – "chat"), *k* (ok – "ok"), *msm* (mislim – "I think"), *mzd* (možda – "perhaps"), *najvrv* (najverovatnije – "most probably"), *nmg* (ne mogu – "I cannot"), *nmvz* (nema veze – "nevermind"), *nnc* (nema na čemu – "you're welcome"), *np* (nema problema – "no problem"), *npm* (nemam pojma – "I have no clue"), *nzm* (ne znam – "I don't know"), *stv* (stvarno – "really"), *ustv* (u stvari – "actually"), *vcs* (večeras – "tonight"), *zvrc* (zovi me –"call me") etc. The full list of abbreviations is available on-line.[5]

The list of previously described lexical and syntactic linguistic features was compiled after a careful manual analysis of the dataset, as it appeared that these features could help with distinguishing message senders that make specific typos and grammatical mistakes, or the ones that write too long or very short messages.

Six features specific to the texting style were also proposed. For example, the minority of senders write in uppercase or in Cyrillic only, and ones that write in German (hence the umlauts count). The motivation for the addition of this so-called stylistic features (see Table A.1) is given in the following text. The motivation for the feature *consecutive_chars* is that the repeated characters, like in a word "heeeeeeeeeej" make an impression of that a person is excited. For the feature *sent_start_lower*, the rationale behind is that if one starts most sentences with lowercase characters, that is probably due to mobile phone operating system, which can be a partially identifying feature; the feature *space_follows_punct* counts the number of times when space existed after punctuation, such as dot or a question mark; finally, *num_double_dot* and *num_double_question* count the number of occurrences of the tokens '..' and '??', respectively. This group of features was extracted with an idea that certain individuals always make similar typing mistakes. For example, some people tend to be informal and/or careless about punctuation (e.g. write two dots instead of one or three), they "join" the sentences together with a dot and no additional blank space, etc. A part of the Python code that generates the list of stylistic features is given in Listing D.1.

4.4 Experimental Results

A dataset of 5,551 short messages, initially published by Šandrih and Vitas (2018), structured as XML was collected from the mobile phone of one person over a period of 4 years.

[4] The full list of emoticons, https://github.com/Branislava/sms_fingerprint/blob/master/features_extraction/emoji.py

[5] Full list of abbreviations, https://github.com/Branislava/sms_fingerprint/blob/master/features_extraction/language_resources.py

Each message contained information about the sender's phone number, the date the message was sent, content of the message and other technical information. The dataset mostly consists of messages in Serbian, typed in both alphabets, Latin and Cyrillic, with some messages in English and German.

For this analysis, two experiments were conducted. In the first case, an aim was to examine if it was possible to automatically identify the owner of the phone. Therefore, class labels were induced from this *type* attribute. There were 2,170 messages written by this specific person (positive class) and 3,381 messages written by someone else (negative class), making this dataset slightly unbalanced.

For the second experiment, the goal was to examine if it was possible to build a classifier which was able to automatically recognise messages written by public services (banks, parking service, taxi, mobile providers and similar). In this case, there were 918 messages sent from public services (positive class) and 4,633 messages sent by humans (negative class).

The First Experiment: Specific Person vs. Others

The classification model was built to tell whether an unseen message was written by the native mobile phone owner (positive class, label 0) or by someone else (negative class, label 1). Class labels were induced from the *type* attribute of `<sms>` element.

List of fifteen features that had the strongest influence on the GB model can be seen in Figure 4.2. Order of these most important features may slightly change during different cross-validation folds, depending on the message instances selected for the training set. The majority of the most influential features are lexical linguistic features: ratio of uppercase characters and message length (significant for persons who write in uppercase), message length alone, ratio of upper and lowercase letters, presence of spaces after punctuation, usage of question marks and dots. The fact that these features showed up as most important was not a surprise, since it was expected that exactly these features are what makes person's SMS writing style distinguishable from other senders'.

The Second Experiment: Human vs. Machine

Despite the dataset being unbalanced in this case, the task was much easier than the previous. List of fifteen features that had the strongest influence on the GB classifier's outcome are shown in Figure 4.3.

Most of these features are related to presence of numbers, which was expected. These reports mainly consist of different digits that represent date and time when the report was sent, amount of money in a bank account, time when the parking card expires, etc. Similarly, length of these messages is also somewhat specific, i.e. reports usually contain more tokens than regular humans' messages. Another common feature is the number of the full-stop characters used in comparison to other characters. Reports are usually longer

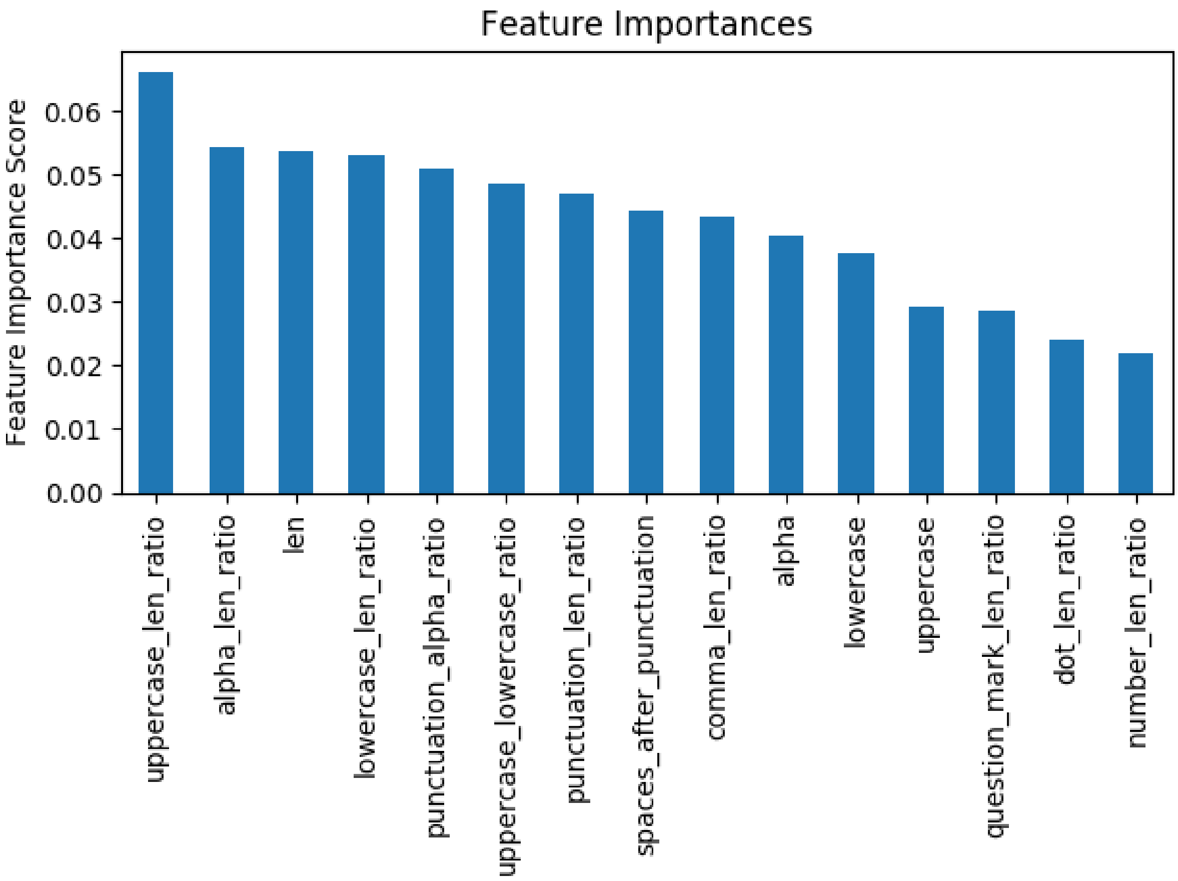


FIGURE 4.2: Most important features for the first model: Specific Person vs. Others

and contain few sentences, each concluded with a full-stop, which could not be guaranteed for informal messages. It can also be noticed that features have stronger influence (higher scores, y-axis) than in the previous experiment.

This method has proven to be well-performing especially in the cases when there is a small number of samples, which are high-dimensional vectors (Santhanam, Saranya, and Kundathil, 2018). As explained earlier, datasets of SMS messages are not easy to gather, so this method was a perfect solution for the problem. In order to test the performance of the proposed GB method, a comparison with various ML classifiers was made. In the following text, values of the parameters that were found to be optimal for these classifiers are given, used with implementations of the classifiers given within *SciKit-Learn*, a Machine Learning module for Python (Pedregosa, Varoquaux, Gramfort, Michel, Thirion, Grisel, Blondel, Prettenhofer, Weiss, and Dubourg, 2011):

SVM Different values of the penalty parameter C for the linear SVM were examined. Other parameters of this classifier are: *gamma* (ignored if kernel is not Radial Basis Function, as in this case), *tol* (tolerance for stopping criterion, default value is 0.001), *class_weight* (if not given, all classes are supposed to have weight 1) and *max_iter*

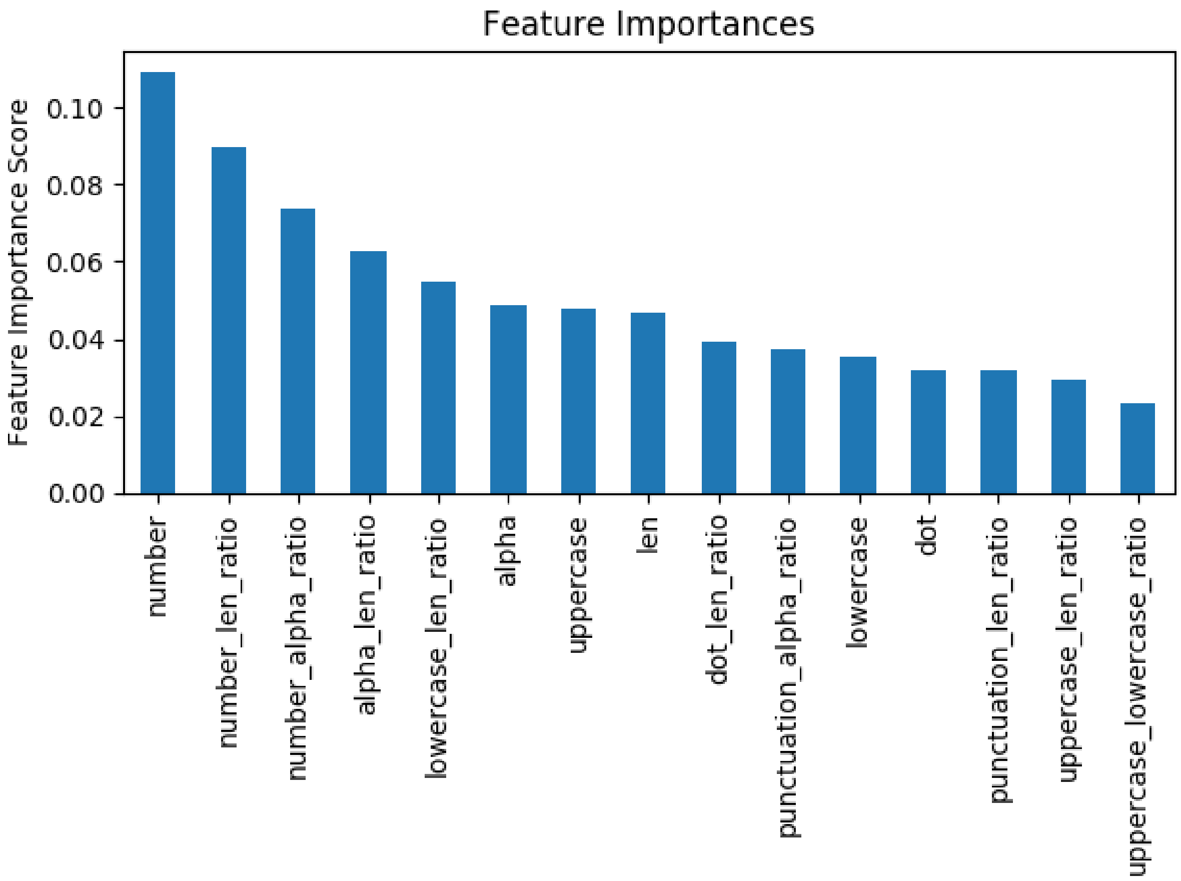


FIGURE 4.3: Most important features for the second model: Human vs. Machine

(maximum number of iterations; by default, this number is unlimited)

MLP As for parameters, except regularisation term *alpha* = 1, the default values were used: *hidden_layer_sizes* = 100, the rectified linear unit function (relu) for *activation* parameter, stochastic gradient-based optimiser (adam) for *solver*, *tol* = 0.0001 as tolerance for optimisation etc.

Gradient Boosting Before building this classifier, grid search was performed in order to find optimal classifier's parameters. At the end, model was tuned with the next parameter values: *learning_rate* = 0.1, *n_estimators* = 160, *min_samples_split* = 10, *min_samples_leaf* = 30, *max_depth* = 9, *max_features* = 11, *subsample* = 0.8 and *random_state* = 10.

The performance of classifiers was evaluated in the 5-fold CV setting using the following basic measures: accuracy, precision, recall and F-score. As a baseline, a classifier that always predicts the majority class in the dataset was used.

Detailed results for the 1^{st} experiment are given in Table 4.1, in favour of both positive and negative classes.

TABLE 4.1: Classification results for the 1^{st} experiment with different algorithms and parameter settings

CLF	ACC	P+	R+	F+	P-	R-	F-
Baseline	.609	.000	.000	.000	.609	1.000	.757
SVM (C=0.025)	.714	.643	.612	.619	.763	.779	.768
SVM (C=1)	.715	.641	.635	.631	.769	.766	.764
RBF	.619	.708	.049	.091	.617	.984	.759
MLP	.686	.656	.485	.528	.723	.815	.757
GB	**.736**	.673	.641	**.653**	.777	.796	**.785**

After this evaluation, GB classifier yielded the highest accuracy and the F_1 score for the both classes, as assumed. Yet, based on the figures from Table 4.1, it can be concluded that the use of emoticons, punctuation or abbreviations is not enough to identify a person. Even for a human, it would be impossible to tell difference among senders who are writing with perfect grammar and without emoticons. But with additional information like the one used by Laperdrix, Rudametkin, and Baudry (2016) and Eckersley (2010), this task might be simpler. For the time being, current results imply that this kind of identification is possible, at least as one of the steps in the AI.

For detailed results of the 2^{nd} experiment in favour of both, positive and negative classes, see Table 4.2.

TABLE 4.2: Classification results for the 2^{nd} experiment with different algorithms and parameter settings

CLF	ACC	P+	R+	F+	P-	R-	F-
Baseline	.835	.000	.000	.000	.835	1.000	.910
SVM (C=.025)	.984	.964	.937	.950	.988	.993	.990
SVM (C=1)	.989	.968	.966	.967	.993	.994	.993
RBF	.947	1.000	.679	.805	.940	1.000	.969
MLP	.982	.939	.953	.946	.991	.987	.989
GB	**.993**	.984	.973	**.978**	.995	.997	**.996**

It can be concluded that results are much better in this case, and that the GB classifier proved to perform best for this similar problem, as well. Yet, the task was also much simpler, due to specific nature of messages written by public services, such as high rate of digits and absence of emoticons.

4.5 Concluding Remarks

The technique proposed in this chapter was developed for solving Authorship Identification classification task on short messages written in Serbian. In order to solve this task in a supervised manner, it is important to have a representative dataset of SMS data and metadata such as the sender's name, phone number, etc. Due to privacy concerns, people have trust issues and are not willing to share their SMS messages. Twitter data might seem to be a good candidate (Twitter corpora are publicly available, there is the same character count threshold and Tweets can be easily retrieved), but Twitter posts and SMS messages are not having the same purpose. An SMS message is addressed to a specific person and most often asks questions or answers them. Tweets mostly contain opinions or comments, referring to other users or topics using hash tags. These hash tags are very common in tweets and can be even a more reliable source than text features. Although the problem itself could be stated on any type of text that is interchanged between two or more sides (Facebook posts, tweets, E-mails, SMS messages, forum posts, Viber/WhatsApp messages etc.), it is expected that, due to the difference in purpose of these different services, different techniques would give the best results for each of them.

In the future, the intention is to generalise the problem so Facebook and Twitter posts can also be subjects of the study. This is primarily aimed at enriching the model with new features, such as message semantics (word meanings, context, used language dialect and chat history), sender's gender, common phrases used by a sender and even information about the device from which the message is sent (e.g. the device model or underlying operating system).

5 Sentiment Classification of Short Messages

This chapter examines the influence of various linguistic features on a sentiment coming across in short messages. The research question asked in this study is the following: Is it possible, based on various linguistic features, to determine the sentiment expressed in a very short message which lacks sufficient semantic information, but potentially contains misspellings, diacritics omission, character repetitions, etc.?

In Section 5.1, a brief overview of the existing work on the task of Sentiment Classification (SC) of short messages is offered, followed by a brief survey of the related work in Section 5.2. A novel method for sentiment classification in short messages is proposed in Section 5.3. The results are discussed in Section 5.4. Finally, conclusions and ideas for future work are presented in Section 5.5.

5.1 Introduction

Authors of short messages have a need to express their mood, voice, facial expressions and much more that oral communication contains. In the written communication, senders can only rely on characters. Researchers performed SC on textual content of various structure. Regardless of the dataset, task of SC in the context of social media relies on predefined sets of emoticons.

Twitter turned out to be one of the most valuable resources for the researchers. Go, Bhayani, and Huang (2009) applied Machine Learning algorithms for classifying the sentiment of Twitter messages using distant supervision on training data consisting of Twitter messages with emoticons. The authors proved that standard Machine Learning algorithms have higher accuracy when trained on data with emoticons. Miličević Petrović, Ljubešić, and Fišer (2017) explored a dataset of Twitter messages and analysed types of transformations that occurred in these texts. They noticed that people tend to write messages in a way that people who read them can experience the whole emotional state of the author. For example, senders use uppercase letters in the case of "shouting"; they excessively use emoticons in order to express their mood and attitude; another often used transformations are common abbreviations and shortened form of words. Davidov, Tsur, and Rappoport (2010) proposed a supervised SC framework which was based on data from Twitter, by using fifty Twitter tags and fifteen smileys as sentiment labels. The authors also explored dependencies and overlap between different sentiment types represented by smileys and Twitter hash tags. Mukherjee, Malu, AR, and Bhattacharyya (2012) also performed a SC on Twitter, where gold standard was obtained by automatically annotating tweets based on their hash tags. In a multi-stage system, the authors addressed the problems of spams, misspellings, slang and abbreviations, entity specificity in the context of the topic searched and pragmatics embedded in text. A lexicon-based approach was implemented by Andriotis and Tryfonas (2014). These authors examined the similarity between Twitter feeds and SMS messages found on smart phones. They investigated common characteristics of both formats for the purpose of SC.

Author's mood classification was studied by Mishne (2005) on a data consisting of a large collection of blog posts which include an indication of the writer's mood. A study developed by Derks, Bos, and Von Grumbkow (2007) examined the influence of social context on the use of emoticons in Internet communication. Participants in a short chat were asked some questions and had to respond either with a text, emoticon or a combination. It turned out that the participants preferred to answer with emoticons in socio-emotional rather than in task-orientated social contexts. Inkpen, Keshtkar, and Ghazi (2009) explored the task of automatic emotion analysis and generation in texts. Authors classified texts by classes of emotions. They also discussed the possibility of generating texts that express specific emotions.

Jibril and Abdullah (2013) made an overview of scholarly research in the field of electronic communication, in order to investigate applications of emoticons in some facets of computer-mediated communication. The focus of Škorić (2017) was the use of emoticon-rich texts on the Web in language-neutral SC. For that purpose, a desktop application Emotiscale was implemented and evaluated.[1]

5.2 Related Work

A lot of research related to Sentiment Classification (SC) in short texts based on emoticons and slang abbreviations has been carried out, but not many researchers have worked on short messages. Walther and D'Addario (2001) conducted an experiment to determine the effects of three common emoticons on message interpretations. The results showed that the contributions of emoticons were outweighed by verbal content.

Neviarouskaya, Prendinger, and Ishizuka (2009) addressed the task of recognising personal emotional state or a sentiment conveyed through text. The authors developed an Affect Analysis model designed to handle the informal messages written in an abbreviated or expressive manner. Ptaszynski, Dybala, Komuda, Rzepka, and Araki (2010) created database of emoticons, collecting emoticons from numerous dictionaries of face marks and online jargon. They decomposed each emoticons into "mouth" and "eyes" elements and then analysed patterns of these semantic areas of emoticons, while Ptaszynski, Rzepka, Araki, and Momouchi (2011) later discussed the importance of emoticons in NLP.

[1] Emotiscale, http://emoti.jerteh.rs/

Pavalanathan and Eisenstein (2015) questioned whether predefined pictographic characters also known es "emojis" will come to replace earlier orthographic methods of paralinguistic communication, known as "emoticons".

Emoticon analysis was not an only approach that gave good results. Kiritchenko, Zhu, and Mohammad (2014) adopted supervised statistical Text Classification approach, leveraging a variety of semantic and sentiment features in order to detect sentiments of short informal textual messages. They also utilise three general-purpose sentiment lexica, that automatically capture many peculiarities of the social media language, containing common intentional and unintentional misspellings.

5.3 Proposed Method for Sentiment Classification of Short Messages

The method proposed in this chapter was initially published in (Šandrih, 2019). It can be described as follows: as an input, a dataset of (sentiment, SMS) pairs should be provided. The following step is the extraction of features. Afterwards, a linear SVM classifier is trained based on these features. For a short message unseen during the training phase, the SVM model tries to determine whether the message contains positive, negative or neutral sentiment. This approach is illustrated in Figure 5.1.

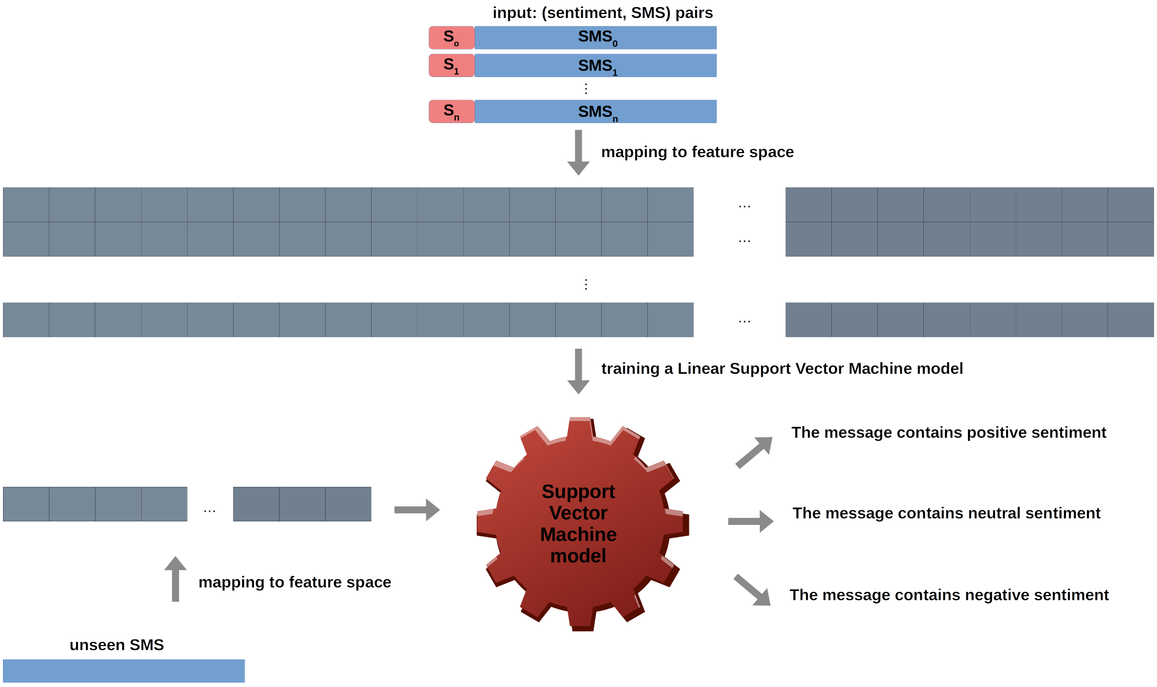

input: (sentiment, SMS) pairs
S_0
SMS_0
S_1
SMS_1
S_n
SMS_n
mapping to feature space
training a Linear Support Vector Machine model
Support Vector Machine model
The message contains positive sentiment
The message contains neutral sentiment
The message contains negative sentiment
mapping to feature space
unseen SMS

FIGURE 5.1: Proposed method for sentiment classification of short messages

The approach proposed in this work is similar to the work of Ojamaa, Jokinen, and Muischenk (2015), in the sense that a sentiment of a message was predicted based on the author's writing style. It neither uses nor creates specific lexica, as in techniques proposed e.g. by Mladenović, Mitrović, Krstev, and Vitas (2016) and Mladenović, Krstev, Mitrović, and Stanković (2017). Similarly as Kiritchenko, Zhu, and Mohammad (2014), a method that relies on previously compiled sets of emoticons and common abbreviations used in modern texting is proposed. Based on an assumption that authors of short messages tend to express their mood with the specific usage of characters (including grouping them into emoticons), different types of features are selected, classified and extracted. Similarly as Aleksieva-Petrova, Minkov, and Petrov (2017), a Web service and a Web application were developed, but not for the text document classification itself, rather for the extraction of the mentioned features.

The intuition behind this classifier was, as shown empirically, SVM model's performance for Text Classification related tasks of similar nature (Shafiabady, Lee, Rajkumar, Kallimani, Akram, and Isa, 2016; Mohammad, Alwada'n, and Al-Momani, 2016; Jain and Mandowara, 2016; Fatima and Srinivasu, 2017; Saad and Shaker, 2017; Amrani, Lazaar, and Kadiri, 2018; Taher, Akhter, and Hasan, 2018).

5.3.1 Proposed Feature Space

In this study, the dataset of $\sim$ 6,000 SMS messages was used, initially published by Šandrih and Vitas (2018) and extended afterwards. The dataset mostly contains messages in Serbian (more than 96%), but also in English and German. Each message in the dataset was first manually annotated as having positive, negative or neutral sentiment. Next, the sets of lexical, syntactic and stylistic linguistic features were extracted and different ML classifiers were trained, evaluated and compared using this set of features.

Various linguistic features were extracted from the *body* attribute of <sms> elements. The full list of the proposed set of stylistic, lexical and syntactical linguistic features is given in Tables A.1, A.2 and A.3, respectively (indicated by x in column S).

The list of lexical features (see Table A.2) includes the following seventy character-based features: counts of each punctuation character, lowercase and uppercase alphabetic characters, digits, diacritics, umlauts, etc. Apart from the absolute counts, ratios of all these numbers to a total number of characters in the message were added as additional features.

Seven word-based lexical features used especially for this task are: average length of tokens, average sentence length, ratio of short words (up to three letters) to a total number of tokens, number of distinct words, ratio of number of distinct words to a total number of words, a number of words that occur more than once in a sentence, and the ratio of number of words that occurred more than once to a total number of words.

The list of proposed syntactic features (see Table A.3) can be divided into two categories: emoticons and abbreviations (as earlier in Section 4.3.1). It is expected that emoticons have the highest influence on the impression about the mood of a message sender. Use of short word forms, slang words and other kind of abbreviations is not uncommon in texting. An extensive list of common slang abbreviations in Serbian and English was compiled. The absolute count of each abbreviation occurrence was then used as a single feature per short message instance.

The set six hand-crafted stylistic features (see Table A.1 and the rationale behind in Section 4.3.1) was selected after careful manual analysis of the dataset, as it seemed that these features could help with differentiating formal and informal tone in messages.

For this particular dataset, the full list of features (for more technical details, refer to Appendix E) was extracted. The dataset is available as a CSV file of 621 features for the full dataset.[2]

5.4 Experimental Results

For this experiment, a dataset of SMS messages from one person's smart phone was used. Most of the senders were in their early twenties and they used the informal texting language. This means the use of short forms of words, abbreviations and emoticons. Therefore, this work relies on the informality of the dataset and tries to discover the influence of modern language patterns on sentiments contained in messages. It should be noted that analysis performed on these texts slightly differs from the general perception of SC. It can be considered as a *mood analysis* since it tries to distinguish in what *tone* should a reader experience a message. Similar service is offered by commercial systems Twilio[3] and Nexmo.[4]

This dataset contains 6, 171 short messages,[5] exported from a phone in an XML format. Each message contains information about sender's number, date, message body and other technical information, as previously explained in Chapter 4. Each message was previously manually labelled as neutral (i.e. carries no sentiment information, 3,272 samples), positive (carries a positive sentiment, 2,719 samples) or negative (180 samples), hence, the dataset is unbalanced.

Messages that contain less than 10 characters (including blanks) were discarded, since they would represent noise. This is justified by the fact that it is almost impossible, even for humans, to tell the mood from so little information. Some of these messages are: `16a`, `BIM`, `k*`. The first two messages refer to office names, while the third represents the continuation of its previous message, where the author made a typo and wanted to make a correction. Examples of some messages along with their annotations are given in Table E.1 and the translated messages in English from Table E.1 are given in Table E.2.

[2] The extracted features for each message from the dataset,
https://github.com/Branislava/sms_sentiment/blob/master/dataset/sms.csv

[3] Twilio, www.twilio.com

[4] Nexmo, www.nexmo.com

[5] This is an expanded version of the dataset used in Chapter 4

Annotation of these messages was not an easy task in many cases. It is important that this was performed adequately, since the outcome of the later classification is tightly connected to the way that messages were manually categorised. Some of the messages that contain ambiguous sentiment are given in the Table E.3, along with their proposed categorisations. The translated messages in English from Table E.3 are given in Table E.4.

In order to evaluate performance of the proposed SVM method, a comparison with various ML classifiers was performed. All feature values were first normalised, i.e. they were mapped to the interval $[0, 1]$. As evaluation metrics, accuracy (Acc), recall (R), precision (P) and F-score (F) are determined. The average values of these metrics in the 5-fold Cross Validation setting in favour of positive (pos), negative (neg) and neutral class (neu) are displayed in Table 5.1.

TABLE 5.1: Evaluation results: Accuracy, Recall, Precision and F-score

CLF	Acc	R	P	F	R	P	F	R	P	F
			POSITIVE			NEGATIVE			NEUTRAL	
kNN	.862	.801	.908	.851	.584	.647	.610	.927	.840	.882
DT	.917	.896	.941	.918	.525	.859	.650	.957	.901	.928
RF	.716	.440	**.954**	.597	.000	.000	.000	**.986**	.657	.788
Ada	.906	.881	.929	.904	.535	.809	.641	.947	.892	.919
NB	.159	.165	.871	.278	**.899**	.031	.060	.113	.753	.197
LR	.910	.867	.949	.906	.606	.904	.725	.963	.883	.921
RBF	.904	**.897**	.907	.902	.496	**.936**	.647	.933	.901	.917
SVM	**.921**	.889	.951	**.919**	.640	.869	**.736**	.963	**.901**	**.931**

Based on the results shown in Table 5.1, the linear SVM classifier delivers the best F_1 scores for all three classes. For further analysis of this model's performance, confusion matrix obtained in a randomly selected iteration is shown in Figure 5.2.

It can be seen that the model suffers from dataset imbalance. Seventeen messages with negative sentiments were classified as neutral. Also, noticeable number of errors occurred when positive samples are classified as neutral (70).

In Table 5.2, some of the messages that were misclassified are listed.

The following paragraphs offer a potential explanation for each miss represented in Table 5.2. In the case of message (1), the content means something positive (English translation would be "Good work"), but since there is no punctuation or emoticon, this message was classified as neutral. It is similar with message (2) — it translates as "Where is my little baby", but this is very problematic case, because it is just a question and it can be considered neutral. In the case of message (3), translated as "I am not coming :-(", it was not annotated well by a human, since this message has a negative content (what can be concluded after the sad emoticon).

In the case of message (4), its sentiment is ambiguous and it is not surprising that the classifier got confused, since this message could be classified as both, positive and negative (English translation would be "Great...but I do not know who that is :/"). Message (5) contains complaining (English translation would be "It is so tough for me!"), but there is

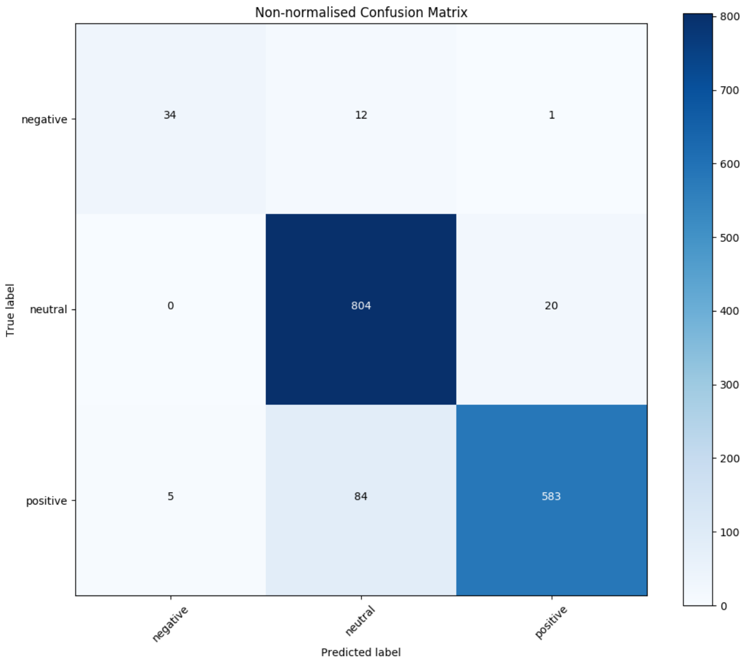


FIGURE 5.2: The confusion matrix

also an exclamation mark. Most of the messages that contained exclamation marks were annotated as positive, and this sample was miss-classified most probably due to this reason. It is similar with message (8), that is a simple statement “Hi, Joca is here!”, but the exclamation mark added makes it sound positive.

Messages (6), (7) and (9) can be compared. Message (9) contains . . . and it was manually annotated as neutral (English translation would be “I am already late, {someone} is getting mad”). Yet, messages (6) and (7), having the similar structure, were manually annotated as negative (Translated as “Oh my, but what can you do...” and “No comment... and what happened next?”, respectively). So the occurrence of . . . is probably common for the both, negative and neutral class.

It can be observed that mood prediction based on a short message is a hard task even for the humans. Due to privacy reasons, original, unprocessed dataset used in this work is not published. Extracted features in a CSV format that can be read as a data frame into R or Python program, along with the code for classification can be found on-line.[6]

[6] The code for sentiment classification of SMS messages represented as feature vectors, https://github.com/Branislava/sms_sentiment/

TABLE 5.2: Misclassified messages

#	Message	Predicted label	True label
1	Vrh brate	NEU	POS
2	Gde je moj bebac?	NEU	
3	ne dolazim :-(	NEG	
4	Super...ali, ne znam ko je :/ :)	POS	NEG
5	Jao.kako mi je tesko!	POS	
6	Bezveze skroz, al sta ces...	NEU	
7	Ccc... I sta je bilo onda?	NEU	
8	Poz, Joca je!	POS	NEU
9	Vec kasnim, ljuti se...	NEG	

5.5 Concluding Remarks

Many messages contain multiple sentiments and they are very hard both to annotate and to classify. One solution for this would be to perform sentence-based sentiment classification. Another approach would be to perform Emotion Recognition on these messages. In this case, each message would contain indicators of presence of certain moods, like anger, surprise, happiness, fear, disgust etc.

Future studies will be dedicated to evaluation of the same procedure on different datasets, differing in origin (SMS, Twitter, Facebook, etc.), size and language. The main contribution of this work is non standard approach for specific use case of Sentiment Classification. It can be concluded that, instead of using predefined lexica, the distribution of characters for very short texts should also be taken into consideration.

6 Conclusions

The focus of the dissertation was on the task of Text Classification, which belongs to both, the field of Natural Language Processing and Machine Learning. Many Natural Language Processing-related problems can be modelled through classification. Such is the case of Sentiment Classification, namely the interpretation and classification of emotions towards a person, service, product, etc. The classes in this case are usually positive, negative and neutral sentiment. Similar problem is Topic Detection, where the task is to identify the theme or topic of a certain input text. In this case, classes are defined depending on the nature of input text and further application. For example, news articles can be classified as belonging to either sport, politics, economy or culture categories. The task of detecting the language of some text can also be considered a TC-task. In this case, the classes are potential natural languages in which the text was written.

In this thesis several problems from NLP domain that can be modelled as Text Classification problems were considered. These problems are: validation of bilingual terminology pairs, classification of good dictionary examples, authorship identification in short messages and sentiment classification in short messages. For all these problems, efficient methods that combine Machine Learning and Natural Language Processing were proposed. Further, this thesis proposed unified approach for modelling classifiers' common feature space for solving all the considered problems. These features are throughout the literature commonly referred to as "linguistic", and can be divided into lexical (character- and word- based), syntactical and stylistic categories.

Validation of Bilingual Terminology Pairs

As terminology within the scientific fields and industry nowadays is primarily established in the English language, for other languages there is a need for standardised terminology that keeps pace with the emergence of new domain terms. Creating bilingual vocabulary manually requires intensive work by domain and language experts. Manual indexing and translating the terms in a domain is a time-consuming job that often fails to keep up with the development of terminology, especially in areas that are constantly evolving.

This was precisely the motive for developing a system for automatic extraction and translation of domain terminology from English into Serbian. For the end results to be credible and usable, and the process fully automated and domain independent, the patterns that exist with good translation pairs were analysed. Upon obtaining potential translation pairs, it is necessary to automatically recognise the patterns already seen in the input pairs that were not available during the training phase. The first of the goals of the thesis was to train a classifier which provides a substitute for human evaluation and which, applying a variety of linguistic features and using mathematical methods, decides whether a pair is a translation between the appropriate languages. Therefore, the dissertation deals with the problem of validating translations of bilingual pairs, so the classes are good translation and bad translation.

The approach proposed in this dissertation consists of several stages. As an input, one needs to provide domain terminology lexica for both the source and target languages. After that, bilingual pairs are formed based on parallel source and target language texts aligned at sentence level. The pairs thus obtained are further subjected to verification by human experts to obtain validated good and bad translation examples. Afterwards, utilisation of the Support Vector Machine with linear and radial kernels allows the prediction of the translation of a bilingual pair not previously seen in the training phase. Such an innovative approach allows more efficient construction of bilingual dictionaries for any domain, since experts from the respective domain can receive suggestions of good bilingual translations and thus significantly save time searching for the final translation.

Classification of Good Dictionary Examples

In any dictionary, electronic or paper, it is useful to have examples of use for each entry. If this problem is posed as a classification task, the goal is to give each use example a score of suitability, that is, to have a mechanism that is able to tell automatically whether a certain text represents a good or a bad use example for a dictionary entry. Illustrating the use of a word with a suitable set of examples is important for good understanding of the meaning, for both speakers and those who are just learning a new language. Examples of language use are everywhere: in the daily press, on social networks, in fora, novels, etc. But not all examples are equally good.

Therefore, this thesis also tried to answer the following question: Is it possible to construct a classifier based on the input set of vocabulary entries and use cases known in advance to be good or bad, that will be able to distinguish the good from the bad candidates? With such an appropriateness scoring system, the selected examples can be ranked among each other. Such ranking can be practically used in software applications, such as vocabulary, whose content is automatically generated and displayed.

The proposed approach first entails the formation of a control dataset verified by linguistic experts. In this set, examples of vocabulary usage are ranked according to their quality, e.g. as inadequate, adequate with big changes, good with few changes and good without any changes. Due to the expected imbalance of subset of data with different rank, techniques for enhancing balancing through sampling or weighting are applied. By further implementation of the classifier based on the Decision Tree, it is possible to automatically rank new use cases according to their quality.

Authorship Identification in Short Messages

The problem of Authorship Identification deals with identifying the author of an arbitrary text document. Some of the applications of Authorship Identification are: in the analysis of literary works and historical documents, in gender recognition, in forensics, etc. The intensive growth of the amount of text corpora available on the web enables the automatic processing of a large amount of text by the use of stylometry and mathematical models. Another consequence is the ability to profile users on the Web. Although the problem of Authorship Identification was widely discussed in literature, the challenges of this problem in the case of short messages are many and require the use of customised linguistic features.

In the dissertation, two cases of authorship identifications were examined. In the first case, the task was to build a model that is able to distinguish automatically whether the message was sent by a specific person or by someone else. In the second case, the model was built to distinguish between messages sent by some public service and messages sent by humans.

The thesis proposed: 1) a method for classifying SMS messages according to whether its author is one specific person or someone else (one-versus-all classification) and 2) a method for classifying SMS messages according to whether its author is a person or the message is automatically generated by a computer. A space of expertly defined linguistic features was proposed, which can be subdivided into the following categories: lexical features (e.g. number of digits or punctuation symbols); syntactical features (e.g., emotograms, abbreviations) and stylistic features (e.g., lowercase letters at the beginning of a sentence, frequency of using negation, etc.). This space especially emphasises the importance of syntactical and stylistic features. After mapping data into the proposed space of linguistic features, the Gradient Boosting method is used to obtain high quality predictions over previously unseen data.

Sentiment Classification of Short Messages

The extremely rapid growth of the amount of the text on the Web made the automatic analysis of public opinion about an object, person or event very desirable. Buyers want to know other people's opinions before buying a product, manufacturing and trading companies are interested in positive and negative criticism about their products and services, political organisations want to get information about voters' opinions and the like. Such analysis is called Sentiment Classification (SC) or Sentiment Analysis. It is also known as Opinion Mining.

In this thesis, a special case of SC is studied: a Sentiment Classification of SMS messages. Starting from the assumption that a text message carries a positive or negative sentiment, but can also be purely informative, classes can be: positive, negative and neutral sentiment. Analysing the sentiments and moods of short messages is a special challenge, as such messages carry significantly less information than, for example, fora discussions. Another problem with short messages lies in their formulation. There is a trend where short message authors strive to achieve similarity to spoken language through the specific use of punctuation, letters and symbols, typing in capital letters, using specialised abbreviations and emoticons. In this way, the authors express themselves, their attitude and their feelings. Authors want their messages to be read in that tone and in the way they were written.

Therefore, this thesis also tries to answer the following question: Is it possible to construct a classifier based on the input set of short messages for which it is known in advance which sentiment they contain and which will be able to distinguish the sentiment contained in previously unseen messages in the future?

A space of expertly defined linguistic features was proposed, as for the previously mentioned task. This space is especially oriented towards the group of syntactical features. Afterwards, for a set of text messages labelled according to the carried sentiment type, various classification models were examined. These methods include Linear and Radial Basis Function Support Vector Machine, k-Nearest Neighbours, Decision Tree, etc. It was shown that Linear SVM proved to be best for the given problem.

All of the proposed approaches are language independent, under the assumption that specific language resources exist for a desired language of application.

6.1 Contributions

The main results representing the scientific contribution of this thesis are:

- For all the problems this thesis deals with, the first step was common: selecting the appropriate mapping function to represent a raw text as a vector. The dissertation proposed a common feature space into which all the above input data, of different nature and structure, were previously mapped.
- A new approach for automatic Bilingual Terminology Extraction and Validation was developed, which consists of extraction and the validation step. As the solution for the task of bilingual domain term pairs validation, a Support Vector Machine binary classification algorithm was created that is able to detect good and poor translation pairs from the certain domain. The effectiveness of the previously proposed approach was shown on the case of English-Serbian domain terminology lists compilation in the domain of Library and Information Science.
- A new approach based on the Decision Trees for validation of Good Dictionary EXamples was proposed. This is the first step towards the creation of modern electronic dictionaries, which enables automatic scoring and ranking of dictionary entries use examples, without human intervention.
- A new technique based on Gradient Boosting for profiling and recognising authors of SMS messages using lexical, syntactic and stylistic features was proposed. For this problem, two tasks were effectively solved: 1) Authorship Identification of a

certain human sender among other senders, and 2) recognition of SMS messages sent by public services.

- A new method based on Support Vector Machine classifier that uses various lexical, syntactic and stylistic linguistic features for analysing sentiments in SMS messages was developed.

The proposed approaches are important because they successfully and efficiently solve mentioned tasks and offer potential for a number of practical applications. The research undertaken in in this thesis represents a contribution to Natural Language Processing field, especially for the general task of Text Classification. All of the results presented in this work were previously published in international journals and proceedings of international conferences.

6.2 Future Work

The studies intended for the future are as follows. In the first place, the proposed list of linguistic features will be expanded with more hand-crafted features, but also with more generic features, such as POS-tags and word embedding features.

The proposed approach for automatic Bilingual Terminology Extraction and Validation will be further improved by redefining the so-called “match” function and by adjusting the number of well extracted pairs. Similarly, the method will be evaluated on other domains. There is also a lot of room for improvement of the classifier, e.g. by refining the classification and by introducing more levels of “goodness”, i.e. to expand the number of classes to good, bad and “partially-good” translations. On the technical side, the Web application is going to be improved by adding the classifier at the end of the on-line routine, which besides classification, outputs the confidence of the translation correctness.

As the first steps are made, further work towards creating modern electronic dictionaries with the ability of automatic ranking of examples should be pursued. The enrichment of the existing Web service is intended by offering a Web interface that enables users to automatically extract the list of linguistic features used for the analysis, along with the assigned score by the classifier that assigns scores to examples trained in this thesis.

The technique for profiling and recognising authors should be improved, e.g. by introducing more features and examining the effectiveness of this technique on other types of text, such as social network posts and messages retrieved from applications for instant messaging. The same applies for the proposed technique for Sentiment Classification: a further improvement by introducing more generic features is intended. Evaluation on other types of short texts is also planned. Finally, an enrichment of the existing Web application is planned, by enabling automatic classification of the message given an input, which is the one performed by the selected best-performing model.

Bibliography

Abbasi, Ahmed, Hsinchun Chen, and Jay F. Nunamaker (2008). "Stylometric Identification in Electronic Markets: Scalability and Robustness". In: *Journal of Management Information Systems* 25(1), pp. 49–78. DOI: 10.2753/MIS0742-1222250103.

Abbasi, Ahmed and Hsiuchin Chen (2008). "Writeprints: A Stylometric Approach to Identity-level Identification and Similarity Detection in Cyberspace". In: *ACM Transactions on Information Systems* 26, pp. 1–29. DOI: 10.1145/1344411.1344413.

ACL (2005). *What is the ACL and what is Computational Linguistics?* URL: https://www.aclweb.org/portal/what-is-cl.

Agarwal, Apoorv, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau (2011). "Sentiment Analysis of Twitter Data". In: *Proceedings of the Workshop on Language in Social Media (LSM 2011)*. Association for Computational Linguistics, pp. 30–38.

Agarwal, Basant, Heri Ramampiaro, Helge Langseth, and Massimiliano Ruocco (2018). "A Deep Network Model for Paraphrase Detection in Short Text Messages". In: *Information Processing & Management* 54(6), pp. 922–937.

Agresti, Alan (1996). *An Introduction to Categorical Data Analysis*. John Wiley and Sons, Inc. ISBN: 9780471113386.

Aker, Ahmet, Monica Paramita, and Rob Gaizauskas (2013). "Extracting Bilingual Terminologies from Comparable Corpora". In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics: Sofia, Bulgaria, pp. 402–411.

Aleksieva-Petrova, Adelina, Emilyan Minkov, and Milen Petrov (2017). "A Web Application for Text Document Classification based on K-Nearest Neighbor Algorithm". In: *Serdica Journal of Computing* 11(2), pp. 183–198.

Amrani, Yassine Al, Mohamed Lazaar, and Kamal Eddine El Kadiri (2018). "Random Forest and Support Vector Machine based Hybrid Approach to Sentiment Analysis". In: *Procedia Computer Science* 127. Proceedings of the 1st International Conference on Intelligent Computing in Data Sciences, ICDS2017, pp. 511–520. ISSN: 18770509. DOI: https://doi.org/10.1016/j.procs.2018.01.150.

Andonovski, Jelena, Branislava Šandrih, and Olivera Kitanović (2019). "Bilingual Lexical Extraction based on Word Alignment for Improving Corpus Search". In: *The Electronic Library* 37(2), pp. 722–739. DOI: 10.1108/EL-03-2019-0056.

Andriotis Panagiotisand Takasu, Atsuhiro and Theo Tryfonas (2014). "Smartphone Message Sentiment Analysis". In: *Advances in Digital Forensics X*. Springer Berlin Heidelberg: Berlin, Heidelberg, pp. 253–265. ISBN: 9783662449523.

Anđelković, Jelena, Danica Seničić, and Ranka Stanković (2019). "Aligned Parallel Corpus for the Domain of Management". In: *Infotheca – Journal for Digital Humanities* 18(2), pp. 7–28. ISSN: 22179461. DOI: 10.18485/infotheca.2018.18.2.1.

Arcan, Mihael, Marco Turchi, Sara Tonelli, and Paul Buitelaar (2017). "Leveraging Bilingual Terminology to Improve Machine Translation in a Computer Aided Translation Environment". In: *Natural Language Engineering* 23(5), pp. 763–788. DOI: 10.1017/S1351324917000195.

Atkins, B. T. Sue and Michael Rundell (2008). *The Oxford Guide to Practical Lexicography*. Oxford: Oxford University Press. ISBN: 9780199277711.

Baccianella, Stefano, Andrea Esuli, and Fabrizio Sebastiani (2010). "Sentiwordnet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining". In: *Language Resources and Evaluation (LREC 2010)*. Vol. 10. European Language Resources Association (ELRA): Valletta, Malta, pp. 2200–2204.

Baldwin, Timothy and Su Nam Kim (2010). "Multiword Expressions". In: *Handbook of Natural Language Processing*. Ed. by Nitin Indurkhya and Fred J. Damerau. 2nd ed. New York: Chapman and Hall/CRC, pp. 267–292. ISBN: 9780429149207. DOI: 10.1201/9781420085938.

Batanović, Vuk and Boško Nikolić (2017). "Sentiment Classification of Documents in Serbian: The Effects of Morphological Normalization". In: *Telfor Journal* 9(2), pp. 104–109. DOI: 10.5937/telfor1702104B.

Batanović, Vuk, Boško Nikolić, and Milan Milosavljević (2016). "Reliable Baselines for Sentiment Analysis in Resource-Limited Languages: The Serbian Movie Review Dataset". In: *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC'16)*. European Language Resources Association (ELRA): Portorož, Slovenia, pp. 2688–2696.

Bhattacharjee, Joydeep (2018). *fastText Quick Start Guide: Get started with Facebook's library for text representation and classification*. Packt Publishing. ISBN: 9781789130997.

Bird, Steven, Ewan Klein, and Edward Loper (2009). *Natural Language Processing with Python – Analyzing Text with the Natural Language Toolkit*. O'Reilly Media. ISBN: 9780596516499.

Birkeneder, Bastian, Jelena Mitrović, Julia Niemeier, Leon Teubert, and Siegfried Handschuh (2019). "upInf – Offensive Language Detection in German Tweets". In: *Proceedings of the 14th Conference on Natural Language Processing (KONVENS 2018)*. Österreichische Akademie der Wissenschaften, pp. 71–79.

Bishop, Christopher M. (2006). *Pattern Recognition and Machine Learning*. Springer, New York, USA. ISBN: 9788132209065.

Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov (2017). "Enriching Word Vectors with Subword Information". In: *Transactions of the Association for Computational Linguistics* 5, pp. 135–146. ISSN: 2307387X.

Boser, Bernhard E., Isabelle M. Guyon, and Vladimir N. Vapnik (1992). "A Training Algorithm for Optimal Margin Classifiers". In: *Proceedings of the 5th Annual Workshop on Computational Learning Theory*. COLT '92. Association for Computing Machinery: Pittsburgh, Pennsylvania, USA, pp. 144–152. ISBN: 089791497X. DOI: 10.1145/130385.130401.

Bottou, Léon (2010). "Large-Scale Machine Learning with Stochastic Gradient Descent". In: *Proceedings of the 19th International Conference on Computational Statistics (COMPSTAT 2010)*. Physica-Verlag: Heidelberg, Germany, pp. 177–186. ISBN: 9783790826043. DOI: 10.1007/978-3-7908-2604-3_16.

Bouamor, Dhouha, Nasredine Semmar, and Pierre Zweigenbaum (2012). "Identifying Bilingual Multi-Word Expressions for Statistical Machine Translation". In: *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)*. European Language Resources Association (ELRA): Istanbul, Turkey, pp. 674–679. ISBN: 9782951740877.

Bowker, Lynne and Gloria Corpas Pastor (2021). "Translation technology". In: *The Oxford Handbook of Computational Linguistics (preprint)*. Ed. by Ruslan Mitkov. 2nd ed. Oxford University Press. ISBN: 9780199573691.

Breck, Eric and Claire Cardie (2021). "Opinion mining and sentiment analysis". In: *The Oxford Handbook of Computational Linguistics (preprint)*. Ed. by Ruslan Mitkov. 2nd ed. Oxford University Press. ISBN: 9780199573691.

Brill, Eric (1992). "A Simple Rule-Based Part of Speech Tagger". In: *Proceedings of the 3rd conference on Applied Natural Language Processing*. Association for Computational Linguistics, pp. 152–155.

Burges, Christopher J. C. (1998). "A Tutorial on Support Vector Machines for Pattern Recognition". In: *Data Mining and Knowledge Discovery* 2(2), pp. 121–167.

Constant, Mathieu, Gülşen Eryiğit, Johanna Monti, Lonneke Van Der Plas, Carlos Ramisch, Michael Rosner, and Amalia Todirascu (2017). "Multiword Expression Processing: A Survey". In: *Computational Linguistics* 43(4), pp. 837–892. DOI: 10.1162/COLI_a_00302.

Constant, Matthieu and Joakim Nivre (2016). "A Transition-Based System for Joint Lexical and Syntactic Analysis". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. Vol. 1. Association for Computational Linguistics: Berlin, Germany, pp. 161–171.

Cortes, Corinna and Vladimir Vapnik (1995). "Support Vector Networks". In: *Machine Learning* 20(3), pp. 273–297. ISSN: 15730565. DOI: 10.1007/BF00994018.

Cram, Damien and Béatrice Daille (2016). "Terminology Extraction with Term Variant Detection". In: *Proceedings of ACL-2016 System Demonstrations*. Association for Computational Linguistics: Berlin, Germany, pp. 13–18. DOI: 10.18653/v1/P16-4003.

Cristani, Marco, Giorgio Roffo, Cristina Segalin, Loris Bazzani, Alessandro Vinciarelli, and Vittorio Murino (2012). "Conversationally-Inspired Stylometric Features for Authorship Attribution in Instant Messaging". In: *Proceedings of the 20th ACM international conference on Multimedia*. Association for Computing Machinery (ACM), pp. 1121–1124. DOI: 10.1145/2393347.2396398.

Cutting, Doug, Julian Kupiec, Jan Pedersen, and Penelope Sibun (1992). "A Practical Part-of-Speech Tagger". In: *3rd Conference on Applied Natural Language Processing*. Association for Computational Linguistics: United States, pp. 133–140. DOI: 10.3115/974499.974523.

Dale, Robert (2021). "Spoken Language Dialogue Systems". In: *The Oxford Handbook of Computational Linguistics (preprint)*. Ed. by Ruslan Mitkov. 2nd ed. Oxford University Press. ISBN: 9780199573691.

Davidov, Dmitry, Oren Tsur, and Ari Rappoport (2010). "Enhanced Sentiment Learning Using Twitter Hashtags and Smileys". In: *Proceedings of the 23rdInternational Conference on Computational Linguistics: Posters*. Association for Computational Linguistics, pp. 241–249.

Dekker, Peter, Tanara Zingano Kuhn, Branislava Šandrih, and Rina Zviel-Girshin (2019). "Corpus Cleaning via Crowdsourcing for Developing a Learner's Dictionary". In: *Electronic lexicography in the 21st century (eLex 2019): Smart Lexicography. Book of abstracts.* Lexical Computing CZ s.r.o., Brno, Czech Republic, pp. 84–85.

Derks, Daantje, Arjan ER Bos, and Jasper Von Grumbkow (2007). "Emoticons and Social Interaction on the Internet: the Importance of Social Context". In: *Computers in human behavior* 23(1), pp. 842–849.

Didakowski, Jörg, Lothar Lemnitzer, and Alexander Geyken (2012). "Automatic Example Sentence Extraction for a Contemporary German Dictionary". In: *Proceedings of the 15th EURALEX International Congress, 7–11 August 2012, Oslo: Department of Linguistics and Scandinavian Studies*. University of Oslo, pp. 343–349.

Diederich, Joachim, Jörg Kindermann, Edda Leopold, and Gerhard Paass (2003). "Authorship Attribution with Support Vector Machines". In: *Applied intelligence* 19(1-2), pp. 109–123.

Ebert, Sebastian (2017). "Artificial Neural Network Methods Applied to Sentiment Analysis". PhD thesis. Ludwig-Maximilians-Universität München.

Eckersley, Peter (2010). "How Unique is your Web Browser?" In: *Proceedings of the 10th International Conference on Privacy Enhancing Technologies*. Vol. 6205. Springer Verlag: Berlin, Heidelberg, pp. 1–18.

Fatima, Shugufta and B. Dr. Srinivasu (2017). "Text Document Categorization using Support Vector Machine". In: *International Research Journal of Engineering and Technology (IRJET)* 4(2), pp. 141–147.

Fawcett, Tom (2006). "An Introduction to ROC analysis". In: *Pattern Recognition Letters* 27(8). ROC Analysis in Pattern Recognition, pp. 861–74. ISSN: 01678655. DOI: 10.1016/j.patrec.2005.10.010.

Fawi, Fathi Hassan Ahmed and Rodolfo Delmonte (2015). "Italian-Arabic Domain Terminology Extraction From Parallel Corpora". In: *Proceedings of the 2nd Italian Conference on Computational Linguistics CLiC-it 2015*. Accademia University Press, pp. 130–134. DOI: 10.4000/books.aaccademia.1473.

Firth, John R. (1957). "A Synopsis of Linguistic Theory, 1930–1955". In: *Studies in Linguistic Analysis (special volume of the Philological Society)*. Basil Blackwell, Oxford, pp. 1–32. ISBN: 06311130029780631113003.

Friedman, Jerome, Trevor Hastie, and Robert Tibshirani (2001). *The Elements of Statistical Learning*. Vol. 1. 10. Springer series in statistics New York, USA.

Friedman, Jerome H. (2001). "Greedy Function Approximation: a Gradient Boosting Machine". In: *Annals of Statistics* 29(5), pp. 1189–1232. DOI: 10.1214/aos/1013203451.

Gambette, Philippe and Jean Véronis (2010). "Visualising a Text with a Tree Cloud". In: *Classification as a Tool for Research*. Springer Berlin Heidelberg: Berlin, Heidelberg, pp. 561–569. ISBN: 9783642107450.

Garabík, Radovan and Ludmila Dimitrova (2015). "Extraction and Presentation of Bilingual Correspondences from Slovak-Bulgarian Parallel Corpus". In: *Cognitive Studies | Études cognitives*(15), pp. 327–334.

Garside, Roger (1987). "The CLAWS Word-Tagging System". In: *The Computational Analysis of English: A Corpus-Based Approach*. London: Longman, pp. 30–41.

Gitari, Njagi Dennis, Zhang Zuping, Hanyurwimfura Damien, and Jun Long (2015). "A Lexicon-Based Approach for Hate Speech Detection". In: *International Journal of Multimedia and Ubiquitous Engineering* 10(4), pp. 215–230.

Go, Alec, Richa Bhayani, and Lei Huang (2009). *Twitter Sentiment Classification using Distant Supervision*. Tech. rep. CS224N Project Report, Stanford.

Gorjanc, Vojko, Polona Gantar, Iztok Kosem, and Simon Krek, eds. (2017). *Dictionary of Modern Slovene: Problems and Solutions*. Ljubljana University Press, Faculty of Arts. ISBN: 9789612379131.

Graovac, Jelena (2014). "A Variant of n-gram Based Language-Independent Text Categorization". In: *Intelligent Data Analysis* 18(4), pp. 677–695. DOI: 10.3233/IDA-140663.

Graovac, Jelena, Miljana Mladenović, and Ivana Tanasijević (2019). "NgramSPD: Exploring Optimal n-gram Model for Sentiment Polarity Detection in Different Languages". In: *Intelligent Data Analysis* 23(2), pp. 279–296.

Ha, Le An, Ruslan Mitkov, and Gloria Corpas (2008). "Mutual Terminology Extraction using a Statistical Framework". In: *Procesamiento del lenguaje natural* 41, pp. 107–112.

Hakami, Huda, and Danushka Bollegala (2017). "A Classification Approach for Detecting Cross-lingual Biomedical Term Translations". In: *Natural Language Engineering* 23(1), 31–51. DOI: 10.1017/S1351324915000431.

Hamon, Thierry and Natalia Grabar (2018). "Adaptation of Cross-Lingual Transfer Methods for the Building of Medical Terminology in Ukrainian". In: *Computational Linguistics and Intelligent Text Processing*. Ed. by Alexander Gelbukh. Springer International Publishing: Cham, pp. 230–241. ISBN: 9783319754772. DOI: 10.1007/978-3-319-75477-2_15.

Hazem, Amir and Emmanuel Morin (2016). "Efficient Data Selection for Bilingual Terminology Extraction from Comparable Corpora". In: *Proceedings of the 26th International Conference on Computational Linguistics: Technical Papers (COLING 2016)*. The COLING 2016 Organizing Committee: Osaka, Japan, pp. 3401–3411.

Heafield, Kenneth (2011). "KenLM: Faster and Smaller Language Model Queries". In: *Proceedings of the 6th Workshop on Statistical Machine Translation*. Association for Computational Linguistics, pp. 187–197.

Hewavitharana, Sanjika and Stephan Vogel (2016). "Extracting Parallel Phrases from Comparable Data for Machine Translation". In: *Natural Language Engineering* 22(4), 549–573. DOI: 10.1017/S1351324916000139.

Hirschberg, Julia and Christopher D. Manning (2015). "Advances in Natural Language Processing". In: *Science* 349(6245), pp. 261–266.

Hosmer, David W. Jr, Stanley Lemeshow, and Rodney X. Sturdivant (2013). *Applied Logistic Regression*. Vol. 398. John Wiley & Sons. ISBN: 9781118548394.

Hovy, Ed (2021). "Text summarisation". In: *The Oxford Handbook of Computational Linguistics (preprint)*. Ed. by Ruslan Mitkov. 2nd ed. Oxford University Press. ISBN: 9780199573691.

Hutchins, John (2005). "Machine Translation: General Overview". In: *The Oxford Handbook of Computational Linguistics*. Ed. by Ruslan Mitkov. 1st ed. Oxford University Press. ISBN: 9780199276349. DOI: 10.1093/oxfordhb/9780199276349.013.0027.

Inkpen, Diana, Fazel Keshtkar, and Diman Ghazi (2009). "Analysis and Generation of Emotion in Texts". In: *Proceedings of International Conference on Knowledge Engineering Principles and Techniques (KEPT 2009)*, pp. 3–13.

Irvine, Ann and Chris Callison-Burch (2016). "End-to-end Statistical Machine Translation with Zero or Small Parallel Texts". In: *Natural Language Engineering* 22(4), 517–548. DOI: 10.1017/S1351324916000127.

Jaćimović, Jelena, Cvetana Krstev, and Drago Jelovac (2015). "A Rule-Based System for Automatic De-Identification of Medical Narrative Texts". In: *Informatica* 39(1), pp. 45–53.

Jain, Aaditya and Jyoti Mandowara (2016). "Text Classification by combining Text Classifiers to Improve the Efficiency of Classification". In: *International Journal of Computer Application* 6(2). ISSN: 22501797.

Jarvis, Scott and Scott A. Crossley, eds. (2012). *Approaching Language Transfer Through Text Classification: Explorations in the Detection-Based Approach*. Multilingual Matters. ISBN: 9781847696977.

Jibril, Tanimu Ahmed and Mardziah Hayati Abdullah (2013). "Relevance of Emoticons in Computer-Mediated Communication Contexts: An Overview". In: *Asian Social Science* 9(4), p. 201.

Joachims, Thorsten (1998). "Text Categorization with Support Vector Machines: Learning with many Relevant Features". In: *Machine Learning: ECML-98*. Springer Berlin Heidelberg: Berlin, Heidelberg, pp. 137–142. ISBN: 9783540697817. DOI: 10.1007/BFb0026683.

Joachims, Thorsten (2002). *Learning to Classify Text using Support Vector Machines: Methods, Theory and Algorithms*. Vol. 186. Kluwer Academic Publishers Norwell.

Joulin, Armand, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov (2016). "FastText.zip: Compressing text classification models". In: *arXiv preprint arXiv:1612.03651*.

Joulin, Armand, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov (2017). "Bag of Tricks for Efficient Text Classification". In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Association for Computational Linguistics, pp. 427–431.

Jovanović, Mioljub, Goran Šimić, Milan Čabarkapa, Dragan Ranđelović, Vojkan Nikolić, Slobodan Nedeljković, and Petar Čisar (2019). "SEFRA-Web-based Framework Customizable for Serbian Language Search Applications". In: *Acta Polytechnica Hungarica* 16(3), pp. 59–78.

Juang, Biing Hwang and Laurence R. Rabiner (1991). "Hidden Markov Models for Speech Recognition". In: *Technometrics* 33(3), pp. 251–272.

Juola, Patrick (2008). "Authorship Attribution". In: *Foundations and Trends® in Information Retrieval* 1(3), pp. 233–334. DOI: 10.1561/1500000005.

Jurafsky, Daniel and James H. Martin (2019). *Speech and Language Processing*. Preprint on webpage at https://web.stanford.edu/~jurafsky/slp3/.

Kaplan, Ronald M. (2005). "Syntax". In: *The Oxford Handbook of Computational Linguistics*. Ed. by Ruslan Mitkov. 1st ed. Oxford University Press. ISBN: 9780199276349. DOI: 10.1093/oxfordhb/9780199276349.013.0004.

Kecman, Vojislav (2001). *Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models*. MIT Press, Cambridge, MA, United States. ISBN: 9780262112550.

Kešelj, Vlado and Danko Šipka (2008). “A Suffix Subsumption-Based Approach to Building Stemmers and Lemmatizers for Highly Inflectional Languages with Sparse Resources”. In: *Infotheca – Journal for Digital Humanities* 9(1–2), 23a–33a.

Kilgarriff, Adam, Milos Husák, Katy McAdam, Michael Rundell, and Pavel Rychlỳ (2008). “GDEX: Automatically Finding Good Dictionary Examples in a Corpus”. In: *Proceedings of 13th EURALEX international congress*. Universitat Pompeu Fabra Barcelona, Spain, pp. 425–432.

Kiritchenko, Svetlana, Xiaodan Zhu, and Saif M Mohammad (2014). “Sentiment Analysis of Short Informal Texts”. In: *Journal of Artificial Intelligence Research* 50, pp. 723–762. DOI: 10.1613/jair.4272.

Koehn, Philipp (2009). *Statistical Machine Translation*. Cambridge University Press. DOI: 10.1017/CBO9780511815829.

Koehn, Philipp (2020). *Neural Machine Translation*. Cambridge University Press. ISBN: 9781108608480.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, and Richard Zens (2007). “Moses: Open Source Toolkit for Statistical Machine Translation”. In: *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*. Association for Computational Linguistics, pp. 177–180.

Koehn, Philipp, Franz Josef Och, and Daniel Marcu (2003). “Statistical Phrase-based Translation”. In: *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*. Vol. 1. Association for Computational Linguistics, pp. 48–54.

Kontonatsios, Georgios, Claudiu Mihăilă, Ioannis Korkontzelos, Paul Thompson, and Sophia Ananiadou (2014). “A Hybrid Approach to Compiling Bilingual Dictionaries of Medical Terms from Parallel Corpora”. In: *Statistical Language and Speech Processing* 8791, pp. 57–69.

Kosem, Iztok, Kristina Koppel, Tanara Zingano Kuhn, Jan Michelfeit, and Carole Tiberius (2019). “Identification and Automatic Extraction of Good Dictionary Examples: The Case(s) of GDEX”. In: *International Journal of Lexicography* 32(2), pp. 119–137. DOI: 10.1093/ijl/ecy014.

Kovačević, Ljiljana, Vesna Injac Malbaša, and Dobrila Begenišić (2017). *Dictionary of Library and Information Sciences*. National Library of Serbia: Scientific Research Department: Belgrade. ISBN: 9788670353848.

Krstev, Cvetana (2008). *Processing of Serbian. Automata, Texts and Electronic Dictionaries*. Faculty of Philology of the University of Belgrade, p. 227.

Krstev, Cvetana (2014). *Serbian WordNet*. URL: http://korpus.matf.bg.ac.rs/SrpWN/.

Krstev, Cvetana, Ivan Obradović, Miloš Utvić, and Duško Vitas (2014). “A System for Named Entity Recognition Based on Local Grammars”. In: *Journal of Logic and Computation* 24(2), pp. 473–489. DOI: 10.1093/logcom/exs079.

Krstev, Cvetana, Gordana Pavlović-Lažetić, Duško Vitas, and Ivan Obradović (2004). "Using Textual and Lexical Resources in Developing Serbian Wordnet". In: *Romanian Journal of Information Science and Technology* 7(1-2), pp. 147–161.

Krstev, Cvetana, Ranka Stanković, Ivan Obradović, and Biljana Lazić (2015). "Terminology Acquisition and Description Using Lexical Resources and Local Grammars". In: *Proceedings of the Conference Terminology and Artificial Intelligence 2015*, pp. 81–89.

Krstev, Cvetana, Ranka Stanković, and Duško Vitas (2018). "Knowledge and Rule-Based Diacritic Restoration in Serbian". In: *Proceedings of the 3rd International Conference Computational Linguistics in Bulgaria (CLIB-2018)*. The Institute for Bulgarian Language, Bulgarian Academy of Sciences, Sofia, Bulgaria, pp. 41–51.

Krstev, Cvetana, Staša Vujičić-Stanković, and Duško Vitas (2014). "Approximate Measures in the Culinary Domain: Ontology and Lexical Resources". In: *Proceedings of the 9th Language Technologies Conference IS-LT*, pp. 38–43.

Krstev, Cvetana, Branislava Šandrih, Ranka Stanković, and Miljana Mladenović (2018). "Using English Baits to Catch Serbian Multi-Word Terminology". In: *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018)*. European Language Resources Association (ELRA): Miyazaki, Japan, pp. 2487–2494. ISBN: 9791095546009.

Lahbib, Wiem, Ibrahim Bounhas, and Bilel Elayeb (2014). "Arabic-English Domain Terminology Extraction from Aligned Corpora". In: *On the Move to Meaningful Internet Systems: OTM 2014 Conferences*. Springer Berlin Heidelberg: Berlin, Heidelberg, pp. 745–759. ISBN: 9783662455630.

Lamel, Lori and Jean-Luc Gauvain (2021). "Speech recognition". In: *The Oxford Handbook of Computational Linguistics (preprint)*. Ed. by Ruslan Mitkov. 2nd ed. Oxford University Press. ISBN: 9780199573691.

Laperdrix, Pierre, Walter Rudametkin, and Benoit Baudry (2016). "Beauty and the Beast: Diverting Modern Web Browsers to Build Unique Browser Fingerprints". In: 37th *IEEE Symposium on Security and Privacy (S&P 2016), San Jose, United States*. Institute of Electrical and Electronics Engineers, pp. 878–894.

Lemnitzer, Lothar, Christian Pölitz, Jörg Didakowski, and Alexander Geyken (2015). "Combining a Rule-based Approach and Machine Learning in a Good-example Extraction Task for the Purpose of Lexicographic Work on Contemporary Standard German". In: *Electronic Lexicography in the* 21st *Century: Linking Lexical Data in the Digital Age. Proceedings of the eLex 2015 Conference, 11-13 August 2015, Herstmonceux Castle, United Kingdom*. Ljubljana/Brighton: Trojina, Institute for Applied Slovene Studies/Lexical Computing Ltd., pp. 21–31.

Liaw, Andy and Matthew Wiener (2002). "Classification and Regression by Random Forest". In: *R news* 2(3), pp. 18–22.

Liu, Bing (2012). "Sentiment Analysis and Opinion Mining". In: *Synthesis Lectures on Human Language Technologies* 5(1), pp. 1–167.

Ljajić, Adela and Ulfeta Marovac (2019). "Improving Sentiment Analysis for Twitter Data by Handling Negation Rules in the Serbian Language". In: *Computer Science and Information Systems* 16(1), pp. 289–311.

Ljubešić, Nikola and Mario Peronja (2015). "Predicting Corpus Example Quality via Supervised Machine Learning". In: *Electronic Lexicography in the* 21^{st} *Century: Linking Lexical Data in the Digital Age. Proceedings of the eLex 2015 Conference, 11-13 August 2015, Herstmonceux Castle, United Kingdom*. Trojina, Institute for Applied Slovene Studies/Lexical Computing Ltd., pp. 477–485.

Lovins, Julie Beth. (1968). "Development of a Stemming Algorithm". In: *Translation and Computational Linguistics* 11(1), pp. 22–31.

Maas, Andrew L., Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts (2011). "Learning Word Vectors for Sentiment Analysis". In: *Proceedings of the* 49^{th} *annual meeting of the association for computational linguistics: Human language technologies*. Vol. 1. Association for Computational Linguistics, pp. 142–150.

Macklovitch, Elliott (1994). "Using Bi-textual Alignment for Translation Validation: the TransCheck System". In: *Proceedings of the* 1^{st} *Conference of the Association for Machine Translation in the Americas*, pp. 157–168.

Mairesse, Francois, Marilyn Walker, Matthias Mehl, and Roger Moore (Sept. 2007). "Using Linguistic Cues for the Automatic Recognition of Personality in Conversation and Text". In: *Journal of Artificial Intelligence Research (JAIR)* 30, pp. 457–500. DOI: 10.1613/jair.2349.

Manevitz, Larry M. and Malik Yousef (2001). "One-class SVMs for Document Classification". In: *Journal of Machine Learning research* 2(Dec), pp. 139–154.

Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze (2008). *Introduction to Information Retrieval*. Vol. 39. Cambridge University Press. ISBN: 0521865719.

Manning, Christopher D. and Hinrich Schütze (1999). *Foundations of Statistical Natural Language Processing*. MIT press. ISBN: 9780262312134.

Michie, D., D. Spiegelhalter, and Charles Taylor (1999). "Machine Learning, Neural and Statistical Classification". In: *Technometrics* 37. DOI: 10.2307/1269742.

Mikheev, Andrei (2021). "Text segmentation". In: *The Oxford Handbook of Computational Linguistics (preprint)*. Ed. by Ruslan Mitkov. 2nd ed. Oxford University Press. ISBN: 9780199573691.

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean (2013). "Efficient Estimation of Word Representations in Vector Space". In: 1^{st} *International Conference on Learning Representations, ICLR 2013*. URL: http://arxiv.org/abs/1301.3781.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean (2013). "Distributed Representations of Words and Phrases and their Compositionality". In: *Advances in neural information processing systems*, pp. 3111–3119.

Mikolov, Tomas, Wen-Tau Yih, and Geoffrey Zweig (2013). "Linguistic Regularities in Continuous Space Word Representations". In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics: Atlanta, Georgia, pp. 746–751.

Miličević, Maja (2015). "Semi-Automatic Construction of Comparable Genre-Oriented Corpora of Serbian in Cyrillic and Latin Scripts". In: *Anali Filološkog fakulteta* 27(2), pp. 285–300. ISSN: 05228468. DOI: 10.18485/analiff.2015.27.2.14.

Miličević Petrović, Maja, Nikola Ljubešić, and Darja Fišer (2017). "Nestandardno zapisivanje srpskog jezika na Tviteru: mnogo buke oko malo odstupanja?" serbian. In: *Anali*

Filološkog fakulteta 29(2), pp. 111–136. ISSN: 0522-8468. DOI: 10.18485/analiff.2017.29.2.8.

Miller, George A. (1995). "WordNet: A Lexical Database for English". In: *Communications of the ACM* 38(11), pp. 39–41.

Mishne, Gilad (2005). "Experiments with Mood Classification in Blog Posts". In: *Proceedings of ACM SIGIR 2005 workshop on stylistic analysis of text for information access*. Vol. 19. Citeseer, pp. 321–327.

Mitchell, Tom (1997). *Machine Learning*. Burr Ridge, IL: McGraw Hill. ISBN: 9780070428072.

Mitkov, Ruslan (2016). "Computational Phraseology Light: Automatic Translation of Multiword Expressions without Translation Resources". In: *Yearbook of Phraseology* 7(1), pp. 149–166. DOI: 10.1515/phras-2016-0008.

Mitkov, Ruslan (2020). "Translation Memory". In: *The Routledge Handbook of Translation and Memory*. Ed. by Sue-Ann Harding and Ovidi Carbonell Cortés. Basingstoke: Routledge. ISBN: 9781315670898.

"Computational Treatment of Multiword Expressions" (2021). In: *The Oxford Handbook of Computational Linguistics (preprint)*. Ed. by Ruslan Mitkov. 2nd ed. Oxford University Press. ISBN: 9780199573691.

Mitrović, Jelena, Bastian Birkeneder, and Michael Granitzer (2019). "nlpUP at SemEval-2019 Task 6: A Deep Neural Language Model for Offensive Language Detection". In: *Proceedings of The 13th International Workshop on Semantic Evaluation SemEval*. Association for Computational Linguistics: Minneapolis, Minnesota, USA, pp. 722–726. DOI: 10.18653/v1/S19-2127.

Mladenović, Miljana, Cvetana Krstev, Jelena Mitrović, and Ranka Stanković (2017). "Using Lexical Resources for Irony and Sarcasm Classification". In: *Proceedings of the 8th Balkan Conference in Informatics (New York, NY, USA, 2017) BCI17*. Association for Computing Machinery (ACM), pp. 1–8. DOI: 10.1145/3136273.3136298.

Mladenović, Miljana and Jelena Mitrović (2013). "Ontology of Rhetorical Figures for Serbian". In: *International Conference on Text, Speech and Dialogue*. Springer, pp. 386–393.

Mladenović, Miljana, Jelena Mitrović, and Cvetana Krstev (2014). "Developing and Maintaining a WordNet: Procedures and Tools". In: *Proceedings of the 7th Global Wordnet Conference*. University of Tartu Press: Tartu, Estonia, pp. 55–62.

Mladenović, Miljana, Jelena Mitrović, and Cvetana Krstev (2016). "A Language-Independent Model for Introducing a New Semantic Realation between Adjectives and Nouns in a WordNet". In: *The Proceedings of 8th Global WordNet Conference*. "Alexandru Ioan Cuza" University of Iași: Bucharest, Romania, pp. 218–225.

Mladenović, Miljana, Jelena Mitrović, Cvetana Krstev, and Duško Vitas (2016). "Hybrid Sentiment Analysis Framework for a Morphologically Rich Language". In: *Journal of Intelligent Information Systems* 46(3), pp. 599–620.

Mohammad, Adel Hamdan, Tariq Alwada'n, and Omar Al-Momani (2016). "Arabic Text Categorization using Support Vector Machine, Naïve Bayes and Neural Network". In: *GSTF Journal on Computing (JoC)* 5(1), pp. 108–115. ISSN: 22513043. DOI: 10.7603/s40601-016-0016-9.

Monti, Johanna, Violeta Seretan, Gloria Corpas Pastor, and Ruslan Mitkov (2018). *Multiword Units in Machine Translation and Translation Technology*. John Benjamin Publishers. ISBN: 9789027264206.

Mukherjee, Subhabrata, Akshat Malu, Balamurali AR, and Pushpak Bhattacharyya (2012). "TwiSent: a Multistage System for analyzing Sentiment in Twitter". In: *Proceedings of the 21st ACM international conference on Information and knowledge management*. Association for Computing Machinery (ACM), pp. 2531–2534.

Nadeau, David and Satoshi Sekine (2007). "A Survey of Named Entity Recognition and Classification". In: *Lingvisticae Investigationes* 30(1), pp. 3–26.

Neviarouskaya, Alena, Helmut Prendinger, and Mitsuru Ishizuka (2009). "Compositionality Principle in Recognition of Fine-Grained Emotions from Text". In: *Proceedings of 3rd International ICWSM Conference on Weblogs and Social Media*. The AAAI Press, pp. 278–281.

Nobata, Chikashi, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang (2016). "Abusive Language Detection in Online User Content". In: *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, pp. 145–153.

Oakes, Michael (2014). *Literary Detective Work on the Computer*. John Benjamins. ISBN: 9789027270139.

Oakes, Michael (2021). "Author profiling and related applications". In: *The Oxford Handbook of Computational Linguistics (preprint)*. Ed. by Ruslan Mitkov. 2nd ed. Oxford University Press. ISBN: 9780199573691.

Obradović, Ivan, Vorkapić Dalibor, Stanković Ranka, Vulović Nikola, and Kotorčević Miladin (2016). "Towards Translation of Educational Resources using GIZA++". In: *Proceedings of The 7th International Conference on e-Learning (eLearning-2016)*. Metropolitan University Belgrade, pp. 58–63.

Och, Franz Josef and Hermann Ney (2000). "Improved Statistical Alignment Models". In: 38th *Annual Meeting on Association for Computational Linguistics*. Stroudsburg, PA: Association for Computational Linguistics, pp. 440–447. DOI: http://doi.org/10.3115/1075218.1075274.

Ojamaa, Birgitta, Päivi Kristiina Jokinen, and Kadri Muischenk (2015). "Sentiment Analysis on Conversational Texts". In: *Proceedings of the 20th Nordic Conference of Computational Linguistics, NODALIDA 2015, May 11-13, 2015, Vilnius, Lithuania*. 109. Linköping University Electronic Press, pp. 233–237.

Oliver, Antoni (2017). "A System for Terminology Extraction and Translation Equivalent Detection in Real Time: Efficient use of Statistical Machine Translation Phrase Tables". In: *Machine Translation* 31(3), pp. 147–161.

Orebaugh, Angela and Jeremy E. Allnutt (2009). "Classification of Instant Messaging Communications for Forensics Analysis". In: *The International Journal of FORENSIC COMPUTER SCIENCE* 4(1), pp. 22–28. DOI: 10.5769/J200901002.

Đorđević, B. (2014). "Initial Steps in Building Serbian Treebank: Morphological Annotation". In: *Natural Language Processing for Serbian: Resources and Applications*, pp. 41–53.

Đorđević, Bojana P. (2017). "Izrada osnova formalne gramatike srpskog jezika upotrebom metagramatike". PhD thesis. Univerzitet u Beogradu-Filološki fakultet.

Paice, Chris D. (1990). "Another Stemmer". In: *SIGIR Forum* 24(3), pp. 56–61.

Pajić, Vesna, Duško Vitas, Gordana Pavlović-Lažetić, and Miloš Pajić (2013). "WebMonitoring Software System: Finite State Machines for Monitoring the Web". In: *Computer Science and Information Systems* 10(1), pp. 1–23.

Pajić, Vesna, Staša Vujičić Stanković, Ranka Stanković, and Miloš Pajić (2018). "Semi-Automatic Extraction of MultiWord Terms from Domain-Specific Corpora". In: *The Electronic Library* 36(3), pp. 550–567.

Pak, Alexander and Patrick Paroubek (2010). "Twitter as a Corpus for Sentiment Analysis and Opinion Mining". In: *Language Resources and Evaluation (LREC 2010)*. Vol. 10. European Language Resources Association (ELRA), pp. 1320–1326.

Pang, Bo and Lillian Lee (2004). "A Sentimental Education: Sentiment Analysis using Subjectivity Summarization based on Minimum Cuts". In: *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, pp. 271–278. DOI: 10.3115/1218955.1218990.

Pang, Bo and Lillian Lee (2008). "Opinion Mining and Sentiment Analysis". In: *Foundations and Trends® in Information Retrieval* 2(1–2), pp. 1–135.

Pavalanathan, Umashanthi and Jacob Eisenstein (2015). "Emoticons vs. Emojis on Twitter: A Causal Inference Approach". In: *arXiv preprint arXiv:1510.08480*.

Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, and Vincent Dubourg (2011). "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12(Oct), pp. 2825–2830.

Pennebaker, James W. and Laura A. King (1999). "Linguistic Styles: Language Use as an Individual Difference". In: *Journal of Personality and Social Psychology* 77(6), pp. 1296–1312. DOI: 10.1037//0022-3514.77.6.1296.

Pianta, Emanuele, Christian Girardi, and Roberto Zanoli (2008). "The TextPro Tool Suite". In: *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC'08)*. European Language Resources Association (ELRA), pp. 2603–2607. ISBN: 2951740840.

Pianta, Emanuele and Sara Tonelli (2010). "KX: A Flexible System for Keyphrase Extraction". In: *Proceedings of the 5th international workshop on semantic evaluation*. Association for Computational Linguistics, pp. 170–173.

Pilán, Ildikó, Sowmya Vajjala, and Elena Volodina (2016). "A Readable Read: Automatic Assessment of Language Learning Materials Based on Linguistic Complexity". In: *arXiv preprint arXiv:1603.08868*.

Pilán, Ildikó, Elena Volodina, and Lars Borin (2016). "Candidate Sentence Selection for Language Learning Exercises: from a Comprehensive Framework to an Empirical Evaluation". In: *TAL* 57(3), pp. 67–91.

Pilán, Ildikó, Elena Volodina, and Richard Johansson (2013). "Automatic Selection of Suitable Sentences for Language Learning Exercises". In: *20 Years of EUROCALL: Learning from the Past, Looking to the Future: 2013 EUROCALL Conference Proceedings*. Dublin: Research-publishing.net, pp. 218–225.

Pilán, Ildikó, Elena Volodina, and Richard Johansson (2014). "Rule-based and Machine Learning Approaches for Second Language Sentence-level Readability". In: *Proceedings of the 9th Workshop on Innovative Use of NLP for Building Educational Applications*, pp. 174–184.

Pinnis, Marcis, Nikola Ljubešic, Dan Stefanescu, Inguna Skadina, Marko Tadic, and Tatiana Gornostay (2012). “Term Extraction, Tagging, and Mapping Tools for Under-resourced Languages”. In: *Proceedings of the 10th Conference on Terminology and Knowledge Engineering (TKE 2012), June*, pp. 20–21.

Popović, Zoran (2010). “Taggers Applied on Texts in Serbian”. In: *Infotheca – Journal for Digital Humanities* 11(2), 21a–38a.

Porter, Martin F. (1980). “An Algorithm for Suffix Stripping”. In: *Program* 14(3), pp. 130–137.

Powers, David (2011). “Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation”. In: *Journal of Machine Learning Technologies* 2(1), pp. 37–63.

Prager, John (2021). “Question answering”. In: *The Oxford Handbook of Computational Linguistics (preprint)*. Ed. by Ruslan Mitkov. 2nd ed. Oxford University Press. ISBN: 9780199573691.

Ptaszynski, Michal, Pawel Dybala, Radoslaw Komuda, Rafal Rzepka, and Kenji Araki (2010). “Development of Emoticon Database for Affect Analysis in Japanese”. In: *Proceedings of the 4th International Symposium on Global COE Program of the knowledge Federation*, pp. 203–204.

Ptaszynski, Michal, Rafal Rzepka, Kenji Araki, and Yoshio Momouchi (2011). “Research on Emoticons: Review of the Field and Proposal of Research Framework”. In: *Proceedings of 17th Association for Natural Language Processing*, pp. 1159–1162.

Rangel, Francisco, Paolo Rosso, Moshe Koppel, Efstathios Stamatatos, and Giacomo Inches (2013). “Overview of the Author Profiling Task at PAN 2013”. In: *CLEF Conference on Multilingual and Multimodal Information Access Evaluation (CLEF 2013)*. CEUR-WS.org, pp. 352–365.

Rangel, Francisco, Paolo Rosso, Martin Potthast, Benno Stein, and Walter Daelemans (2015). “Overview of the 3rd Author Profiling Task at PAN 2015”. In: *CLEF Conference on Multilingual and Multimodal Information Access Evaluation (CLEF 2015)*. CEUR-WS.org, pp. 1–8.

Read, Jonathon (2005). “Using Emoticons to Reduce Dependency in Machine Learning Techniques for Sentiment Classification”. In: *Proceedings of the ACL student research workshop*. Association for Computational Linguistics, pp. 43–48.

Repar, Andraž and Senja Pollak (2017). “Good Examples for Terminology Databases in Translation Industry”. In: *eLex 2017: eLex 2017: The 5th biennial conference on electronic lexicography, Netherlands, 19-21 September 2017*, pp. 651–661.

Rish, Irina (2001). “An Empirical Study of the Naive Bayes Classifier”. In: *IJCAI 2001 workshop on empirical methods in artificial intelligence*. Vol. 3. 22. IBM New York, pp. 41–46.

Roffo, Giorgio, Cinzia Giorgetta, Roberta Ferrario, and Marco Cristani (2014). “Just the Way You Chat: Linking Personality, Style and Recognizability in Chats”. In: *International Workshop on Human Behavior Understanding*. Springer, pp. 30–41.

Rosenblatt, Frank (1957). *The Perceptron – a Perceiving and Recognizing Automaton*. Report 85-460-1. Cornell Aeronautical Laboratory.

Rosso, Paolo and Francisco Rangel (2020). “Author Profiling Tracks at FIRE”. In: *SN Computer Science* 1(72). DOI: 10.1007/s42979-020-0073-1.

Saad, Yaqeen and Khaled Shaker (2017). “Support Vector Machine and Back Propagation Neural Network Approach for Text Classification”. In: *Journal of University of Human Development* 3(2), pp. 869–876.

Sabtan, Yasser (2016). “Bilingual Lexicon Extraction from Arabic-English Parallel Corpora with a View to Machine Translation”. In: *Arab World English Journal (AWEJ), Special Issue on Translation* 7(5), pp. 317–336. DOI: 10.2139/ssrn.2795900.

Salton, Gerard and Christopher Buckley (1988). “Term-Weighting Approaches in Automatic Text Retrieval”. In: *Information Processing & Management* 24(5), pp. 513 –523. ISSN: 03064573. DOI: https://doi.org/10.1016/0306-4573(88)90021-0.

Sammut, Claude and Geoffrey I. Webb, eds. (2017). *Encyclopedia of Machine Learning and Data Mining*. Springer Science + Business Media New York. ISBN: 9781489976864.

Santhanam, Ramraj, S. Saranya, and Yashwant Kundathil (2018). “Comparative Study of Bagging, Boosting and Convolutional Neural Network for Text Classification”. In: *Indian Journal of Public Health Research & Development* 9(9), pp. 1041–1047. DOI: 10.5958/0976-5506.2018.01138.5.

Schmid, Helmut (1999). “Improvements in Part-of-Speech Tagging with an Application to German”. In: *Natural language processing using very large corpora*. Springer, pp. 13–25.

Schneider, Nathan and Noah A. Smith (2015). “A Corpus and Model Integrating Multiword Expressions and Supersenses”. In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, pp. 1537–1547. DOI: 10.3115/v1/N15-1177.

Scholkopf, Bernhard and Alexander J. Smola (2001). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press: Cambridge, MA, USA. ISBN: 0262194759.

Segaran, Toby (2007). *Programming Collective Intelligence: Building Smart Web 2.0 Applications*. O’Reilly Media, Inc.

Semmar, Nasredine (2018). “A Hybrid Approach for Automatic Extraction of Bilingual Multiword Expressions from Parallel Corpora”. In: *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018)*. European Language Resources Association (ELRA): Miyazaki, Japan. ISBN: 9791095546009.

Shafiabady, Niusha, Lam Hong Lee, Rajprasad Rajkumar, VP Kallimani, Nik Ahmad Akram, and Dino Isa (2016). “Using Unsupervised Clustering Approach to train the Support Vector Machine for Text Classification”. In: *Neurocomputing* 211, pp. 4–10.

Socher, Richard, Eric H. Huang, Jeffrey Pennin, Christopher D. Manning, and Andrew Y. Ng (2011). “Dynamic Pooling and Unfolding Recursive Autoencoders for Paraphrase Detection”. In: *Advances in neural information processing systems*. Curran Associates, Inc., pp. 801–809.

Spasić, Irena, Mark Greenwood, Alun Preece, Nick Francis, and Glyn Elwyn (2013). “FlexiTerm: a Flexible Term Recognition Method”. In: *Journal of Biomedical Semantics* 4(1). ISSN: 20411480. DOI: 10.1186/2041-1480-4-27.

Specia, Lucia and Yorick Wilks (2021). “Machine translation”. In: *The Oxford Handbook of Computational Linguistics (preprint)*. Ed. by Ruslan Mitkov. 2nd ed. Oxford University Press. ISBN: 9780199573691.

Srivastava, Ashok N. and Mehran Sahami (2009). *Text Mining: Classification, Clustering, and Applications*. Chapman and Hall/CRC. ISBN: 9781420059458.

Stamatatos, E. (2009). "A Survey of Modern Authorship Attribution Methods". In: *Journal of the Association for Information Science and Technology (JASIST)* 30(3), pp. 538–556.

Stanković, Ranka and Cvetana Krstev (2016). *LeXimir*. URL: http://korpus.matf.bg.ac.rs/soft/LeXimir.html.

Stanković, Ranka, Cvetana Krstev, Ivan Obradović, Biljana Lazić, and Aleksandra Trtovac (2016). "Rule-based Automatic Multi-word Term Extraction and Lemmatization". In: *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*. European Language Resources Association (ELRA), pp. 507–514. ISBN: 9782951740891.

Stanković, Ranka, Cvetana Krstev, Duško Vitas, Nikola Vulović, and Olivera Kitanović (2017). "Keyword-Based Search on Bilingual Digital Libraries". In: *Semantic Keyword-Based Search on Structured Data Sources: COST Action IC1302 2nd International KEYSTONE Conference, IKC 2016, Cluj-Napoca, Romania, September 8–9, 2016, Revised Selected Papers*. Springer International Publishing: Cham, pp. 112–123. ISBN: 9783319536408. DOI: 10.1007/978-3-319-53640-8_10.

Stanković, Ranka, Branislava Šandrih, Rada Stijović, Cvetana Krstev, Duško Vitas, and Aleksandra Marković (2019). "SASA Dictionary as the Gold Standard for Good Dictionary Examples for Serbian". In: *Electronic lexicography in the 21st century. Proceedings of the eLex 2019 conference*. 1-3 October 2019, Sintra, Portugal. Brno: Lexical Computing CZ, s.r.o., pp. 248–269.

Stanković, Ranka, Cvetana Krstev, Biljana Lazić, and Mihailo Škorić (2018). "Electronic Dictionaries — from File System to Lemon Based Lexical Database". In: *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018)*. European Language Resources Association (ELRA), pp. 48–56. ISBN: 9791095546191.

Stanković, Ranka, Ivan Obradović, Cvetana Krstev, and Duško Vitas (2011). "Production of Morphological Dictionaries of Multi-Word Units Using a Multipurpose Tool". In: *Computational Linguistics-Applications Conference*. Jachranka: Polskie Towarzystwo Informatyczne, pp. 77–84.

Stijović, Rada and Ranka Stanković (2018). "Digitalno izdanje Rečnika SANU: formalni opis mikrostrukture Rečnika SANU". In: *Srpska leksikografija – rečnici srpskog jezika kao izvorišta gramatičkih i semantičkih istraživanja*. Vol. 47. Naučni sastanak slavista u Vukove dane 1. Međunarodni slavistički centar, Filološki fakultet, Univerzitet u Beogradu, pp. 427–440. ISBN: 9788661535062. DOI: 10.18485/msc.2018.47.1.ch40.

Taboada, Maite, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede (2011). "Lexicon-based Methods for Sentiment Analysis". In: *Computational linguistics* 37(2), pp. 267–307.

Taher, SM Abu, Kazi Afsana Akhter, and KM Azharul Hasan (2018). "N-gram based Sentiment Mining for Bangla Text using Support Vector Machine". In: *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*. Institute of Electrical and Electronics Engineers, pp. 1–5.

Talmor, Alon, Mor Geva, and Jonathan Berant (2017). "Evaluating Semantic Parsing against a Simple Web-based Question Answering Model". In: *Proceedings of the 6th Joint*

Conference on Lexical and Computational Semantics (SEM 2017). Association for Computational Linguistics, pp. 161–167.

Taslimipoor, Shiva, Anna Desantis, Manuela Cherchi, Ruslan Mitkov, and Johanna Monti (2016). "Language Resources for Italian: Towards the Development of a Corpus of Annotated Italian Multiword Expressions". In: *Proceedings of the 3rd Italian Conference on Computational Linguistics (CLiC-it), Napoli*. Accademia University Press, pp. 285–290. DOI: 10.4000/books.aaccademia.1850.

Thurmair, Gregor and Vera Aleksić (2012). "Creating Term and Lexicon Entries from Phrase Tables". In: *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT 2012)*, pp. 253–260.

Tomašević, Aleksandra, Ranka Stanković, Miloš Utvić, Ivan Obradović, and Božo Kolonja (2018). "Managing Mining Project Documentation using Human Language Technology". In: *The Electronic Library* 36(6), pp. 993–1009. DOI: 10.1108/EL-11-2017-0239.

Tong, Simon and Daphne Koller (2001). "Support Vector Machine Active Learning with Applications to Text Classification". In: *Journal of Machine Learning Research* 2(Nov), pp. 45–66.

Trost, Harald (2005). "Morphology". In: *The Oxford Handbook of Computational Linguistics*. Ed. by Ruslan Mitkov. 1st ed. Oxford University Press. ISBN: 9780199276349. DOI: 10.1093/oxfordhb/9780199276349.013.0002.

Tsvetkov, Yulia and Shuly Wintner (2010). "Extraction of Multi-word Expressions from Small Parallel Corpora". In: *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*. COLING '10. Association for Computational Linguistics: Beijing, China, pp. 1256–1264.

Tufis, Dan and Radu Ion (2021). "POS tagging". In: *The Oxford Handbook of Computational Linguistics (preprint)*. Ed. by Ruslan Mitkov. 2nd ed. Oxford University Press. ISBN: 9780199573691.

Utvić, Miloš (2011). "Annotating the Corpus of Contemporary Serbian". In: *Infotheca – Journal for Digital Humanities* 12(2), 36a–47a.

Utvić, Miloš (2014). "Izgradnja referentnog korpusa savremenog srpskog jezika". PhD thesis. Univerzitet u Beogradu, Filološki fakultet.

Vapnik, Vladimir N. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag. ISBN: 9781475732641.

Vapnik, Vladimir N. (1999). "An Overview of Statistical Learning Theory". In: *Institute of Electrical and Electronics Engineers Transactions on Neural Networks* 10(5), pp. 988–999.

Varga, Dániel, Péter Halácsy, András Kornai, Viktor Nagy, László Németh, and Viktor Trón (2005). "Parallel Corpora for Medium Density Languages". In: *Proceedings of the Recent Advances in Natural Language Processing (RANLP 2005)*, pp. 590–596.

Vasiljević, Nebojša (2015). "Automatska obrada pravnih tekstova na srpskom jeziku". PhD thesis. Univerzitet u Beogradu, Filološki fakultet.

Vintar, Špela and Darja Fišer (2008). "Harvesting Multi-Word Expressions from Parallel Corpora". In: *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 08)*. European Language Resources Association (ELRA), pp. 1091–1096. ISBN: 2951740840.

Vitas, Duško and Cvetana Krstev (2012). "Processing of Corpora of Serbian Using Electronic Dictionaries". In: *Prace Filologiczne* 63, pp. 279–292.

Vitas, Duško, Ljubomir Popović, Cvetana Krstev, Ivan Obradović, Gordana Pavlović-Lažetić, and Mladen Stanojević (2012). *The Serbian Language in the Digital Age*. Ed. by Georg Rehm and Hans Uszkoreit. META-NET White Paper Series. Springer. ISBN: 9783642307546. DOI: 10.1007/978-3-642-30755-3.

Vujičić Stanković, Staša, Cvetana Krstev, and Duško Vitas (2014). "Enriching Serbian WordNet and Electronic Dictionaries with Terms from the Culinary Domain". In: *Proceedings of the 7th Global Wordnet Conference*. University of Tartu Press: Tartu, Estonia, pp. 127–132.

Vujičić Stanković, Staša and Vesna Pajić (2012). "Information Extraction from the Weather Reports in Serbian". In: *Balkan Conference in Informatics (BCI 12)*. Citeseer, pp. 105–108.

Walther, Joseph B. and Kyle P. D'Addario (2001). "The Impacts of Emoticons on Message Interpretation in Computer-Mediated Communication". In: *Social science computer review* 19(3), pp. 324–347.

Warner, William and Julia Hirschberg (2012). "Detecting Hate Speech on the World Wide Web". In: *Proceedings of the 2nd Workshop on Language in Social Media*. Association for Computational Linguistics, pp. 19–26.

Wilson, Theresa, Janyce Wiebe, and Paul Hoffmann (2005). "Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis". In: *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pp. 347–354. DOI: 10.3115/1220575.1220619.

Witten, Ian H., Eibe Frank, Mark A. Hall, and Christopher J. Pal (2016). *Data Mining: Practical Machine Learning Tools and Techniques*. 4th ed. Morgan Kaufmann. ISBN: 9780128042915.

Xu, Yan, Luoxin Chen, Junsheng Wei, Sophia Ananiadou, Yubo Fan, Yi Qian, I Eric, Chao Chang, and Junichi Tsujii (2015). "Bilingual Term Alignment from Comparable Corpora in English Discharge Summary and Chinese Discharge Summary". In: *BMC Bioinformatics* 16(1), p. 149. DOI: 10.1186/s12859-015-0606-0.

Yih, Wen-Tau, Xiaodong He, and Christopher Meek (2014). "Semantic Parsing for Single-Relation Question Answering". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 643–648.

Zečević, Anđelka (2011). "N-gram Based Text Classification According to Authorship". In: *Proceedings of the 2nd Student Research Workshop associated with RANLP 2011*. Association for Computational Linguistics: Hissar, Bulgaria, pp. 145–149.

Zečević, Anđelka and Staša Vujičić-Stanković (2013). "Language Identification: The Case of Serbian". In: *Proceedings of 35th Anniversary of Computational Linguistics in Serbia*. University of Belgrade, Faculty of Mathematics, pp. 101–112.

Zgusta, Ladislav (1971). *Manual of lexicography*. Vol. 39. Walter de Gruyter. ISBN: 9783111349183.

Zhang, Ziqi and Lei Luo (2019). "Hate Speech Detection: A Solved Problem? The Challenging Case of Long Tail on Twitter". In: *Semantic Web* 10(5), pp. 925–945. DOI: 10.3233/SW-180338.

Zhang, Ziqi, Johann Petrak, and Diana Maynard (2018). "Adapted TextRank for Term Extraction: A Generic Method of Improving Automatic Term Extraction Algorithms".

In: *Procedia Computer Science* 137, pp. 102–108. ISSN: 18770509. DOI: 10.1016/j.procs.2018.09.010.

Zheng, Rong, Jiexun Li, Hsinchun Chen, and Zan Huang (2006). “A Framework for Authorship Identification of Online Messages: Writing-style Features and Classification Techniques”. In: *Journal of the Association for Information Science and Technology (JASIST)* 57(3), pp. 378–393. DOI: 10.1002/asi.20316.

Zingano Kuhn, Tanara, Peter Dekker, Branislava Šandrih, Rina Zviel-Girshin, Špela Arhar Holdt, and Tanneke Schoonheim (2019). “Crowdsourcing Corpus Cleaning for Language Learning Resource Development”. In: *EuroCALL 2019: European Association of Computer Assisted Language Learning*, p. 159.

Šandrih, Branislava (2018). “Fingerprints in SMS messages: Automatic Recognition of a Short Message Sender Using Gradient Boosting”. In: 3rd *International Conference Computational Linguistics in Bulgaria (CLIB 2018)*. Department of Computational Linguistics at the Institute for Bulgarian Language with the Bulgarian Academy of Sciences: Sofia, Bulgaria, pp. 203–210.

Šandrih, Branislava (2019). “SMS Sentiment Classification based on Lexical Features, Emoticons and Informal Abbreviations”. In: *Serdica Journal of Computing* 13(1-2), pp. 81–94. ISSN: 13147897.

Šandrih, Branislava, Cvetana Krstev, and Ranka Stanković (2019). “Development and Evaluation of Three Named Entity Recognition Systems for Serbian - The Case of Personal Names”. In: *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*. INCOMA Ltd.: Varna, Bulgaria, pp. 1060–1068. DOI: 10.26615/978-954-452-056-4_122.

Šandrih, Branislava, Cvetana Krstev, and Ranka Stanković (2020). “Two Approaches to Compilation of Bilingual Multi-Word Terminology Lists from Lexical Resources”. In: *Natural Language Engineering*. DOI: 10.1017/S1351324919000615.

Šandrih, Branislava B. and Duško M. Vitas (2018). “Kvantitativni pregled jezika u kratkim porukama”. In: *Naučni sastanak slavista u Vukove dane – Srpski jezik i njegovi resursi: teorija, opis i primene*. Vol. 47. 3. Međunarodni slavistički centar, Beograd, pp. 155–165. ISBN: 9788661535215. DOI: 10.18485/msc.2018.47.3.ch10.

Šimić, Goran P. (2019). “Improving e-government Services For Advanced Search”. In: *Vojnotehnički glasnik* 67(2), pp. 307–325.

Škorić, Mihailo (2017). “Classification of Terms on a Positive-negative Feelings Polarity Scale Based on Emoticons”. In: *Infotheca – Journal for Digital Humanities* 17(1), pp. 67–91. DOI: 10.18485/infotheca.2017.17.1.4.

List of Abbreviations

Acc	**A**ccuracy
AI	**A**utorship **I**dentification
ANN	**A**rtificial **N**eural **N**etworks
AUC ROC	**A**rea **U**nder the **R**eceiver **O**perating **C**haracteristic **C**urve
BOW	**B**ag **O**f **W**ords
CL	**C**omputational **L**inguistics
CM	**C**onfusion **M**atrix
CNN	**C**onvolutional **N**eural **N**etwork
DT	**D**ecision **T**ree
DL	**D**eep **L**earning
FN	**F**alse **N**egative
FP	**F**alse **P**ositive
GB	**G**radient **B**oosting
GDEX	**G**ood **D**ictionary **EX**ample
HMM	**H**idden **M**arkov **M**odel
IDF	**I**nverse **D**ocument **F**requency
kNN	*k* **Nearest N**eighbours
LR	**L**ogistic **R**egression
MAP	**M**ax **A P**osteriori
ML	**M**achine **L**earning
MLP	**M**ulti **L**ayer **P**erceptron
MSE	**M**ean **S**quared **E**rror
MT	**M**achine **T**ranslation
MWE	**Multi**-**W**ord **E**xpressions
MWT	**Multi**-**W**ord **T**erms
NB	**N**aïve **B**ayes
NER	**N**amed **E**entity **R**ecognition
NLP	**N**atural **L**anguage **P**rocessing
POS	**P**art **O**f **S**peech
RBF	**R**adial **B**asis **F**unction
RNN	**R**ecurrent **N**eural **N**etwork
SA	**S**entiment **A**nalysis
SASA	**S**erbian **A**cademy of **S**ciences and **A**rts
SC	**S**entiment **C**classification
SR	**S**peech **R**ecognition
SOA	**S**tate **O**f the **A**rt
SMT	**S**tatistical **M**achine **T**ranslation

SVM	**S**upport **V**ector **M**achines
SWT	**S**ingle-**W**ord **T**erms
TC	**T**ext **C**lassification
TF	**T**erm **F**requency
TN	**T**rue **N**egative
TP	**T**rue **P**ositive
TTS	**T**ext-**to**-**S**peech

A Space of Linguistic Features

TABLE A.1: The common space of linguistic stylistic features

Short name	Description	T	V	X	A	S
glued_sents	Count of sentences separated by fullstop that does not precede space	N			x	
num_double_dot	Count of ‘..‘ token	N			x	x
num_double_question	Count of ‘?¿ token	N			x	x
consecutive_chars	Times the same consecutive characters occur	N			x	x
sent_start_lower	Count of sentences that start with a lowercase character	N			x	x
space_follows_punct	Number of times space follows a punctuation mark	N			x	

TABLE A.2: The common space of linguistic lexical features

Short name	**Description**	**G**	**T**	V	X	A	S
num_consonants	Ratio of consonants to all characters	c	N	x			
num_vocals	Ratio of vocals to all characters	c	N	x			
sentence-based	Count of Cyrillic characters	c	N			x	x
num_diacritics	Count of diacritic characters	c	N	x		x	x
num_digits	Count of digits	c	N		x	x	x
num_lowercase	Count of lowercase characters	c	N			x	x
num_punctuation	Count of all punctuation marks	c	N		x	x	x
num_umlauts	Count of German umlauts	c	N			x	x
num_uppercase	Count of uppercase characters	c	N			x	x
num_weird_characters	Count of characters: !"#$%&\′()*+-/:;<=>?@[\\]^_‘{|}∼’„”... and letters with accents	c	N		x		
perc_lexical_diversity	Ratio of different characters to all characters	c	N	x			
punct_to_alpha	Ratio of total punctuation to all alphabetic characters	c	N			x	x
sentence_length	Count of all characters	c	N	x	x	x	x
num_*_punctuation	Count of * punctuation mark	c	N		x	x	x
cmn_first_letters_1	Terms begin with the same char	c	C	x			
cmn_first_letters_2	Terms begin with the same 2 characters	c	C	x			
cmn_first_letters_3	Terms begin with the same 3 characters	c	C	x			
cmn_substr_2	Exist common substr of a length of 2	c	C	x			

Short name	**Description**	**G**	**T**	V	X	A	S
cmn_substr_3	Exist common substr of a length of 3	c	C	x			
cmn_substr_4	Exist common substr of a length of 4	c	C	x			
cmn_substr_longer_5	Exist common substr longer than 5	c	C	x			
cmn_substr_longer_6	Exist common substr longer than 6	c	C	x			
cyrillic_to_alpha	Ratio of Cyrillic to alphabetic characters	c	N			x	x
digits_to_alpha	Ratio of digits to alphabetic characters	c	N			x	x
grammarly_sentence	Sentence begins with uppercase and ends with punctuation	c	N		x		
lower_to_upper	Ratio of lowercase and uppercase characters	c	N			x	x
num_nospace_chars	Count of all characters that are not spaces	c	N		x		
num_alphas	Count of alphabetic characters	c	N			x	x
stoplist_intiial	Initial word in a sentence is present in a stoplist	s	C		x		
kwic_abs_position	Position of KWIC in a sentence (absolute value, character position)	s	N		x		
avg_sent_length	Average sentence length	s	N			x	x
avg_freq_in_corpus	Average words' frequency in a referent corpus	t	N		x		
avg_token_len	Average tokens length	t	N	x	x		
num_rare_tokens	Number of tokens with frequency <= 10 in a referent corpus	t	N		x		
num_repeated_lemmas	Number of lemmas that appear multiple times	t	N		x		
num_distinct_words	Number of distinct words	N		x			
num_words_occur_1	Number of words that occur more than once	N		x			

Short name	**Description**	**G**	**T**	V	X	A	S
num_stopwords	Count of stop-words	t	N	x	x		
num_mixed_tokens	Number of tokens with mixed symbols (e.g. letters and digits)	t	N		x		
perc_cmn_tokens	Ratio of common tokens to a total number of tokens	t	N	x	x		
perc_tokens_longer_10	Ratio tokens longer than 10 to all	t	N	x			
perc_tokens_longer_6	Ratio tokens longer than 6 to all	t	N	x			
perc_tokens_longer_8	Ratio tokens longer than 8 to all	t	N	x			
perc_tokens_shorter_3	Ratio tokens shorter than 3 to all	t	N	x			
perc_tokens_shorter_4	Ratio tokens shorter than 4 to all	t	N	x			
perc_tokens_shorter_5	Ratio tokens shorter than 5 to all	t	N	x			
perc_vocab_richness	Ratio of unique tokens to all tokens	t	N	x			
avg_word_len	Average words length	t	N		x		
between_15_40_tokens	Sentence contains between 15 and 40 tokens	t	N		x		
exist_token_12	There is a token with more than 12 characters	t	N		x		
kwic_more_1	True if KWIC appears more than 1	t	N		x		
less_than_60_tokens	There are less than 60 tokens	t	N		x		
max_token_len	Max token length	t	N		x		
max_word_len	Max word length	t	N		x		
min_word_len	Min word length	t	N		x		
more_than_7_tokens	There are more than 7 tokens	t	N		x		
num_tokens	Count of all tokens	t	N	x	x		
num_words	Count of all words	t	N		x		
num_capitalised	Count of words that begin with uppercase, not on the 1st position	t	N		x		

TABLE A.3: The common space of linguistic syntactic features

Short name	**Description**	**G**	**T**	V	X	A	S
abbrev_*	Times * abbreviation occurs	o	N			x	x
num_happy_*_emot	Times * happy emoticon occurs	e	N			x	x
num_kiss_*_emot	Times * kiss emoticon occurs	e	N			x	x
num_misc_*_emot	Times * misc emoticon occurs	e	N			x	x
num_sad_*_emot	Times * sad emoticon occurs	e	C			x	x
num_skeptic_*_emot	Times * skeptic emoticon occurs	e	N			x	x
num_smiley_*_emot	Times * smiley emoticon occurs	e	N			x	x
num_surprised_*_emot	Times * surprised emoticon occurs	e	C			x	x
num_tongue_*_emot	Times * tongue emoticon occurs	e	N			x	x
num_wink_*_emot	Times * wink emoticon occurs	e	N			x	x
num_kiss_emots	Count of kiss emoticons	e	N			x	x
num_sad_emots	Count of sad emoticons	e	N			x	x
num_surprised_emots	Count of surprised emoticons	e	N			x	x
num_wink_emots	Count of wink emoticons	e	N			x	x
num_misc_emots	Count of miscellaneous emoticons	e	N			x	x
num_tongue_emots	Count of tongue emoticons	e	N			x	x
num_happy_emots	Count of happy emoticons	e	N			x	x
num_skeptic_emots	Count of skeptic emoticons	e	N			x	x
num_smiley_emots	Count of smiley emoticons	e	N			x	x

Short name	**Description**	**G**	**T**	V	X	A	S
contains_web	Contains an email or a web address	o	N		x		
is_compound	Component is a compound	o	N	x			
init_word_tag	POS-tag of the first word	p	N	x	x		
perc_adjectives	Ratio of adjectives to all tokens	p	N	x			
perc_adverbs	Ratio of adverbs to all tokens	p	N	x			
perc_conjunctions	Ratio of conjunctions to all tokens	p	N	x			
perc_nouns	Ratio of nouns to all tokens	p	N	x			
perc_numerals	Ratio of numerals to all tokens	p	N	x			
perc_prepositions	Ratio of prepositions to all tokens	p	N	x			
perc_pronouns	Ratio of pronouns to all tokens	p	N	x	x		
perc_verbs	Ratio of verbs to all tokens	p	N	x			
tag_0	POS-tag of the 1st word	p	C	x			
tag_1	POS-tag of the 2nd word	p	C	x			
tag_2	POS-tag of the 3rd word	p	C	x			
tag_3	POS-tag of the 4th word	p	C	x			
tag_4	POS-tag of the 5th word	p	C	x			
tag_5	POS-tag of the 6th word	p	C	x			

B Extraction and Validation of Bilingual Terminology Pairs

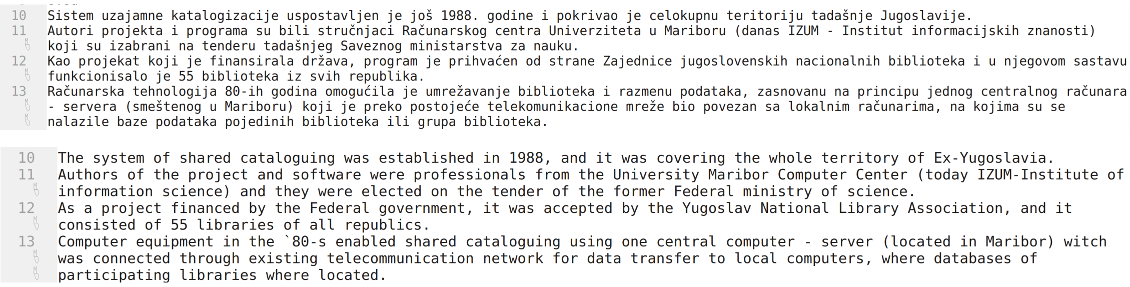
Web Application for Terminology Extraction

In this section, a Web application that implements the proposed technique for terminology extraction is presented. It is available on-line.[1]

The Web application comprises of three modules: 1) input, 2) alignment and post-processing and 3) results module. Each module is briefly described and shown in the following subsections.

Input Module

First, a user has to upload two sentence-aligned text files. These input files must have the same names, but the files' extensions should indicate the language contained (e.g. *f1.en* and *f1.sr*). These files are later fed into GIZA++. An example of the input file pair in English/Serbian is given in Figure B.1.



```
10  Sistem uzajamne katalogizacije uspostavljen je još 1988. godine i pokrivao je celokupnu teritoriju tadašnje Jugoslavije.
11  Autori projekta i programa su bili stručnjaci Računarskog centra Univerziteta u Mariboru (danas IZUM - Institut informacijskih znanosti)
    koji su izabrani na tenderu tadašnjeg Saveznog ministarstva za nauku.
12  Kao projekat koji je finansirala država, program je prihvaćen od strane Zajednice jugoslovenskih nacionalnih biblioteka i u njegovom sastavu
    funkcionisalo je 55 biblioteka iz svih republika.
13  Računarska tehnologija 80-ih godina omogućila je umrežavanje biblioteka i razmenu podataka, zasnovanu na principu jednog centralnog računara
    - servera (smeštenog u Mariboru) koji je preko postojeće telekomunikacione mreže bio povezan sa lokalnim računarima, na kojima su se
    nalazile baze podataka pojedinih biblioteka ili grupa biblioteka.
```

```
10  The system of shared cataloguing was established in 1988, and it was covering the whole territory of Ex-Yugoslavia.
11  Authors of the project and software were professionals from the University Maribor Computer Center (today IZUM-Institute of
    information science) and they were elected on the tender of the former Federal ministry of science.
12  As a project financed by the Federal government, it was accepted by the Yugoslav National Library Association, and it
    consisted of 55 libraries of all republics.
13  Computer equipment in the `80-s enabled shared cataloguing using one central computer - server (located in Maribor) witch
    was connected through existing telecommunication network for data transfer to local computers, where databases of
    participating libraries where located.
```

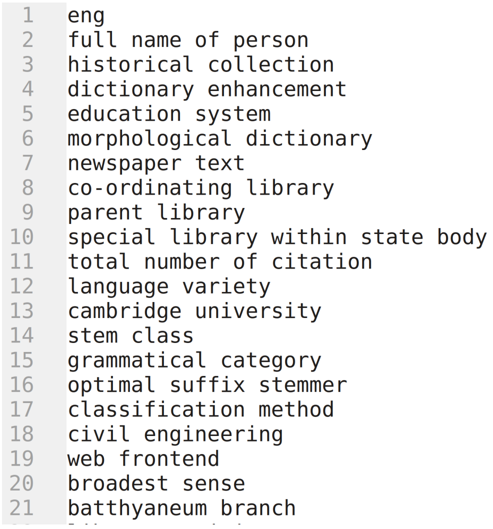
FIGURE B.1: Sentence-aligned file pair in English/Serbian

Afterwards, a user has to upload a list of English terms. First line should contain a header, and each line should contain one term. An example of the input file is given in Figure B.2.

Next, a user has to upload a list of terms in Serbian (not necessarily MWUs). First line is a header, each line contains a term and its frequency (for filtering later), separated with | ("pipe" character). An example of the input file is given in Figure B.3.

The interface of this module is displayed in Figure B.4.

[1] Bilingual Terminology Extraction (BiLTE), http://bilte.jerteh.rs



```
eng
full name of person
historical collection
dictionary enhancement
education system
morphological dictionary
newspaper text
co-ordinating library
parent library
special library within state body
total number of citation
language variety
cambridge university
stem class
grammatical category
optimal suffix stemmer
classification method
civil engineering
web frontend
broadest sense
batthyaneum branch
```

FIGURE B.2: A verticalised list of English terms

Alignment and Post-Processing Module

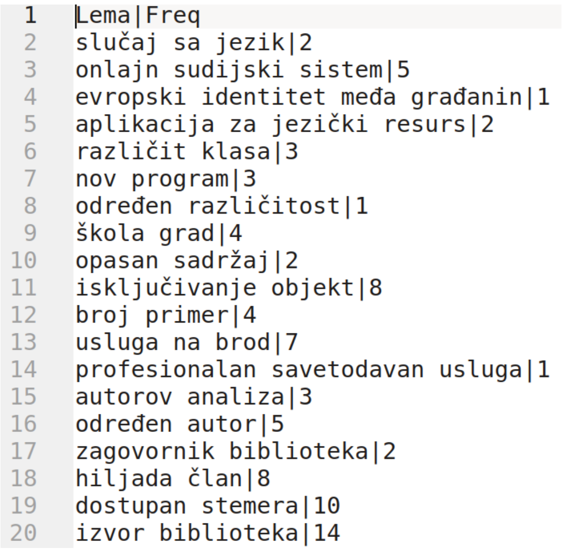
After running GIZA++ on aligned sentences, two post-processing steps follow. The first step is filtering by discarding terms that are out of the domain. This step is followed by a lemmatisation of English chunks with WordNet (Miller, 1995) and Serbian chunks with e-dictionaries for Serbian and Unitex Krstev (2008) (explained in Section 2.3, Processing i).

The interface of this module is displayed in Figure B.5.

Results Module

Which results can be obtained depends on the input file with English terms. If, instead of input shown in Figure B.2, user uploads pairs of source and target terms separated by comma, more results that are useful for evaluation can be retrieved. The basic steps of this module are: 1) keeping only candidates present in the English list (explained in Section 2.3, Processing ii), 2) performing intersection with Serbian extracted MWUs (explained in Section 2.3, Processing iii) and 3) additional filtering (optional) by eliminating bad candidates from the previous step. For all the above described steps, the resulting Excel tables containing obtained pairs can be downloaded.

The interface of this module is displayed in Figure B.6.



```
Lema|Freq
slučaj sa jezik|2
onlajn sudijski sistem|5
evropski identitet među građanin|1
aplikacija za jezički resurs|2
različit klasa|3
nov program|3
određen različitost|1
škola grad|4
opasan sadržaj|2
isključivanje objekt|8
broj primer|4
usluga na brod|7
profesionalan savetodavan usluga|1
autorov analiza|3
određen autor|5
zagovornik biblioteka|2
hiljada član|8
dostupan stemera|10
izvor biblioteka|14
```

FIGURE B.3: A verticalised list of Serbian terms

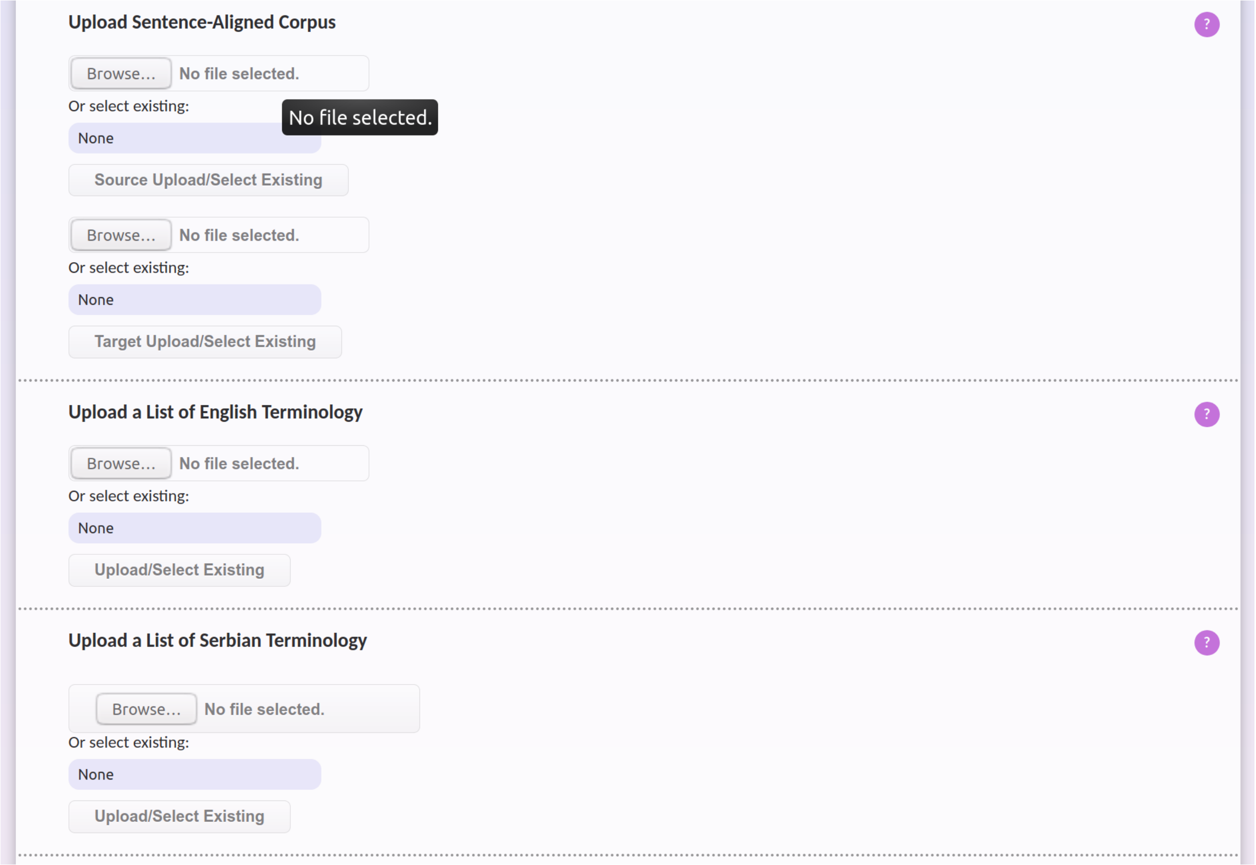
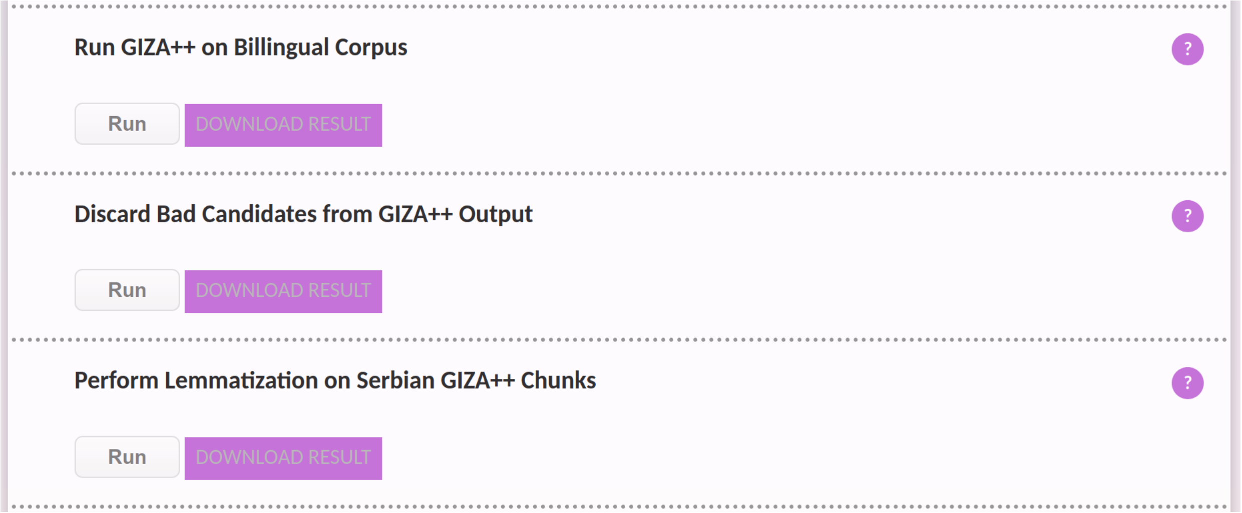


FIGURE B.4: Input module of the BiLTe Web application

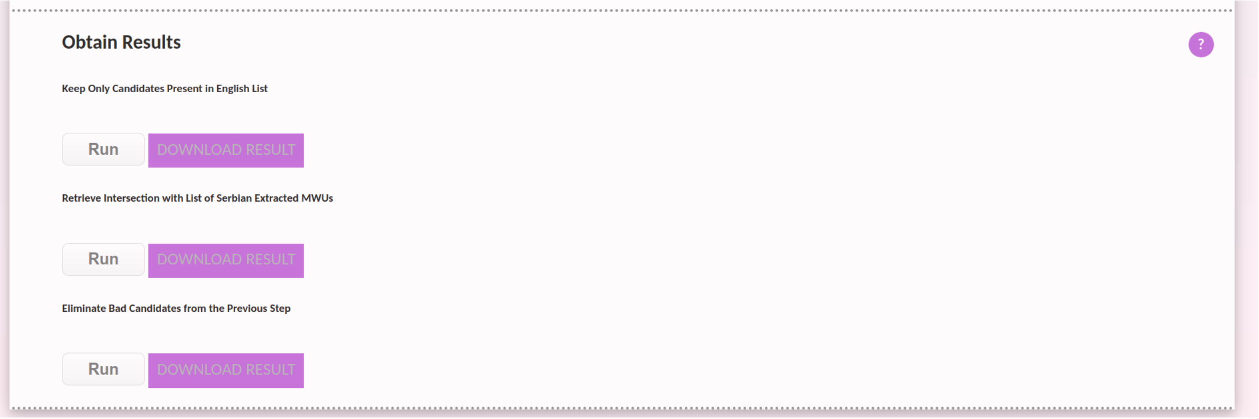


Run GIZA++ on Billingual Corpus

Discard Bad Candidates from GIZA++ Output

Perform Lemmatization on Serbian GIZA++ Chunks

FIGURE B.5: Pre-processing and Alignment module of the BiLTe Web application



Obtain Results

Keep Only Candidates Present in English List

Retrieve Intersection with List of Serbian Extracted MWUs

Eliminate Bad Candidates from the Previous Step

FIGURE B.6: The module for obtaining results of the BiLTe Web application

C Classification of Good Dictionary EXamples for Serbian

In Figures C.1-C.10, different feature distributions over five partitions of the SASA dictionary (labelled D01, D02, D18, D19 and D20) and over the partitions of the control dataset (CN, DP, ON and SJ) are shown.

Figure C.1 presents frequency distribution by number of words in the examples (feature *num_words* in Table A.2). Each volume of SASA dictionary is represented by a histogram with POS in different colours.

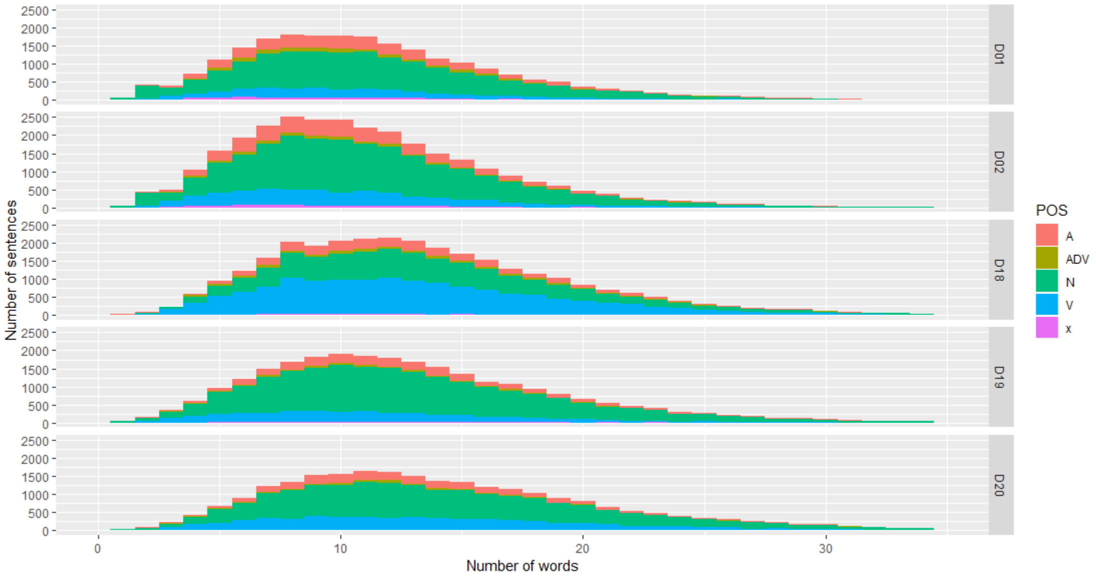


FIGURE C.1: The number of words histograms per dictionary volume partitions

Figure C.2 presents histograms of partitions of the control dataset.

Figure C.3 presents a boxplot, with part of speech on x-axis and sentence length in characters (feature *sentence_length* in Table A.2) on y-axis. Box denotes inter-quartile interval (IQR) with lower (Q1) and upper quartile (Q3), middle bold line presenting the median (Q2); and a rhombus in the middle of the box presenting the average value. Dots present outlier samples with examples longer than $Q3 + 1.5 \cdot IQR$.

Figure C.4 presents a boxplot, with part of speech on x-axis and average token length in characters (feature *avg_token_length* in Table A.2) on y-axis.

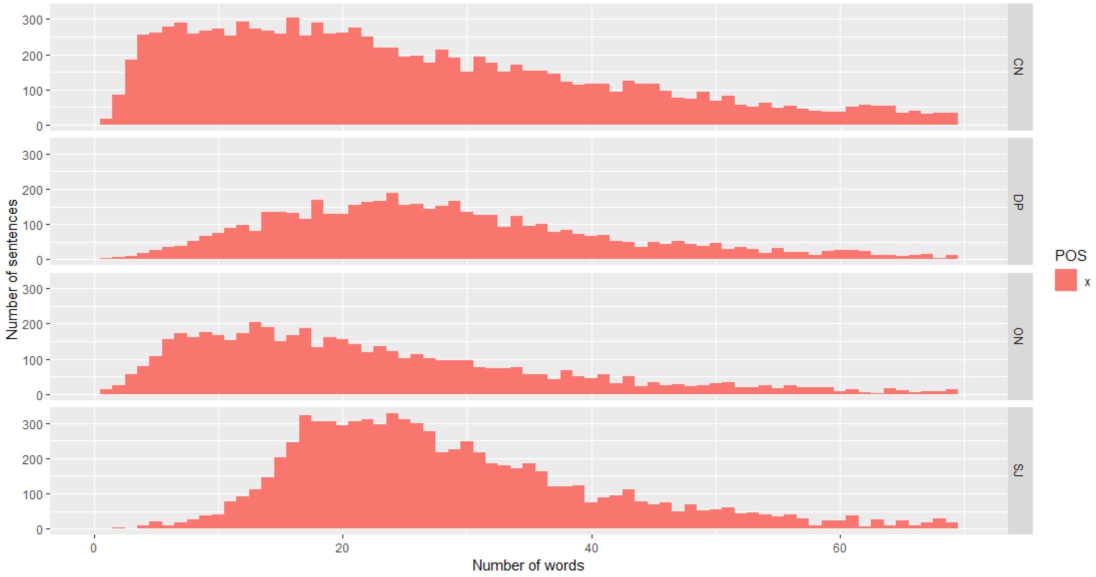


FIGURE C.2: The number of words histograms per text sources in the control set

Figure C.5 presents a boxplot diagram of sentence length (in characters) statistical values per each partition (volume and text collection, feature *sentence_length* in Table A.2).

The distribution of punctuation marks (normalised on sentence size, feature *num_punctuation* in Table A.2) is presented in Figure C.6.

Boxplot for the average token length (feature *avg_token_len* in Table A.2) is given in Figure C.7.

Boxplot of number of pronouns per partition (feature *perc_pronouns* in Table A.2, absolute number of pronouns) is displayed in Figure C.8.

Boxplot of average token frequency per partition (feature *avg_freq_in_corpus* in Table A.2) is shown in Figure C.9.

Boxplot of number of words per language type partitions is given in Figure C.10, left, and for the evaluated dataset on the right.

Web Service

The feature extractor is available online. This developed service receives a string as an input, which can have additional metadata attached, and returns an associative array comprised of feature names and their values. The list of acquired features can also be customised.

The full list of features that can be extracted, along with the guidelines, are available online at `http://gdex.jerteh.rs/`.

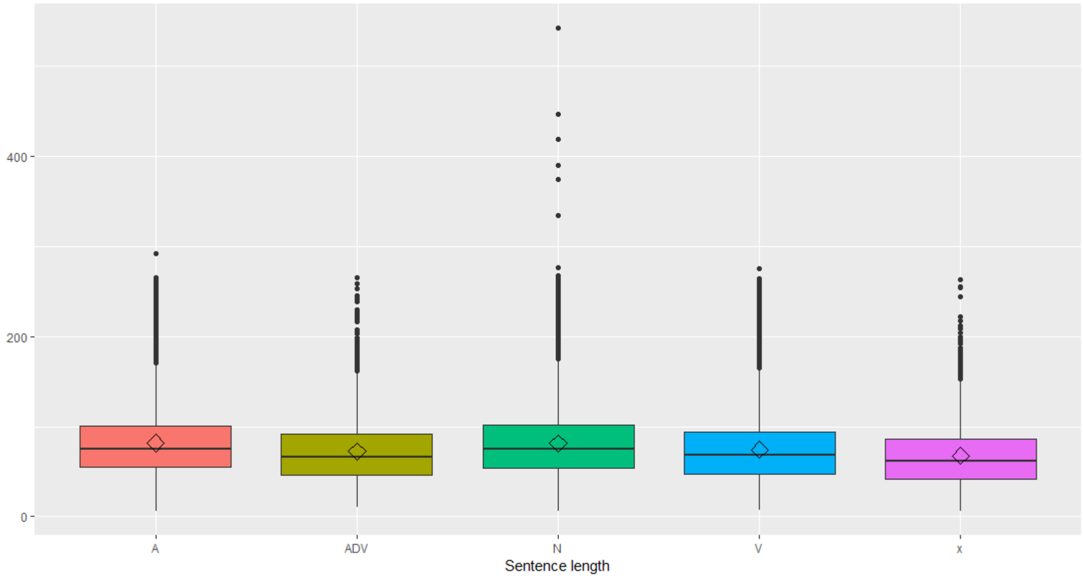


FIGURE C.3: Boxplots showing sentence length per POS in SASA Dictionary

An example of the use of this web service using curl in Unix is the following:

```
curl -d '{"data": "We are demonstrating our feature extractor!",
          "lang":"en",
          "kwic": "use",
          "feature_names": ["sentence_length",
                            "avg_word_len",
                            "no_all_tokens"]
          }'
     -H "Content-Type: application/json"
     -X POST http://147.91.183.8:12347/features
```

and the fields are:

data (string) mandatory, contains text for which features are being extracted

lang (string) optional (the default value is “sr” for Serbian, but most of the features can be extracted for English, as well)

kwic (string) optional (only for headword-dependent features)

feature_names (list of strings) optional (if omitted, returns list of all feature values)

For the given example, the output would be:

```
{
 "no_all_tokens": 7,
 "avg_word_len": 6.166666666666667,
 "sentence_length": 43
}
```

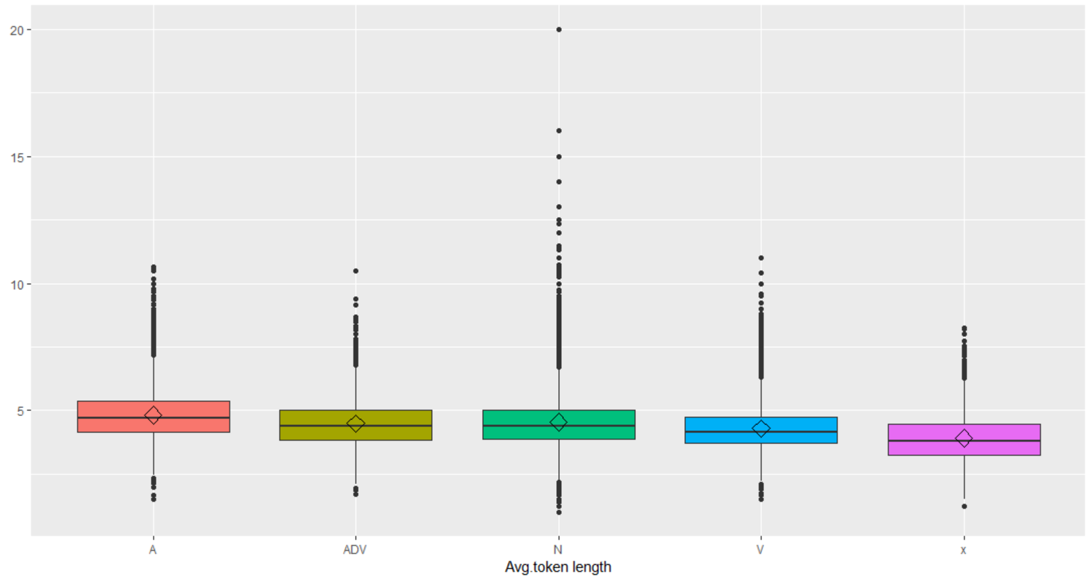


FIGURE C.4: Boxplots showing average token length per POS in SASA Dictionary

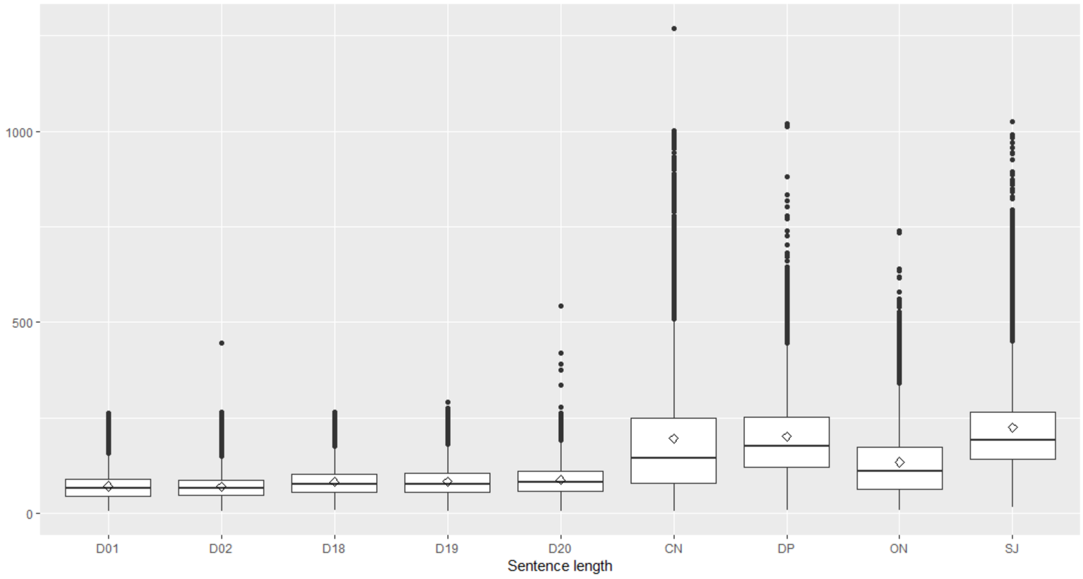


FIGURE C.5: Boxplot of sentence (example) length (in number of characters) per partition

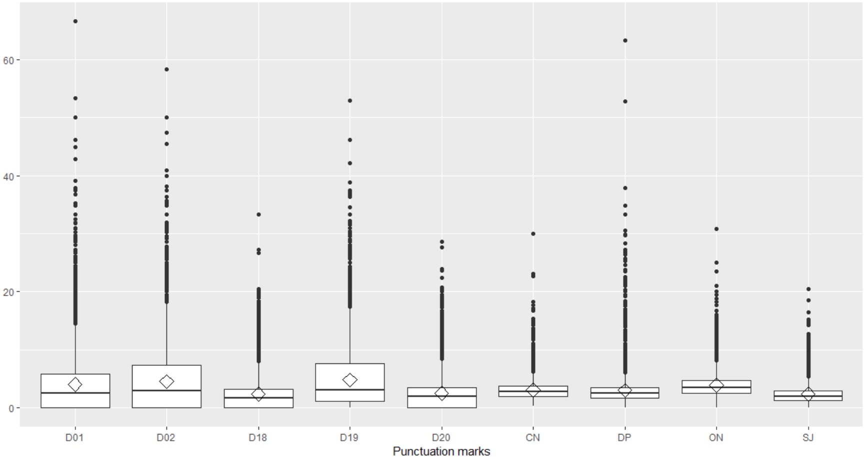


FIGURE C.6: Boxplot for the number of punctuation marks

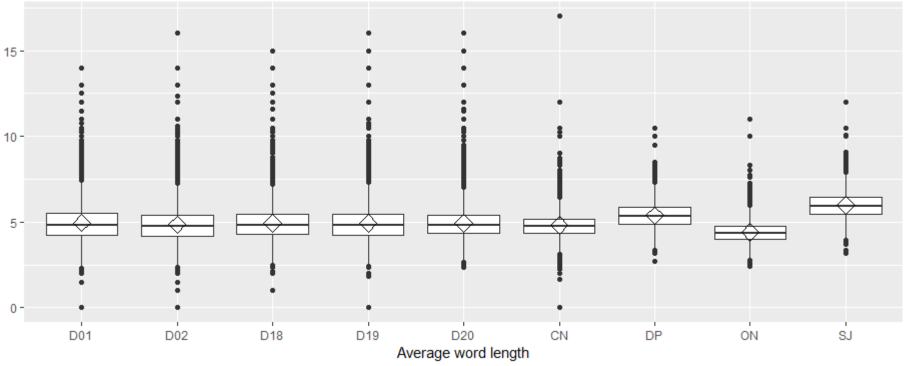


FIGURE C.7: Boxplot for the token length

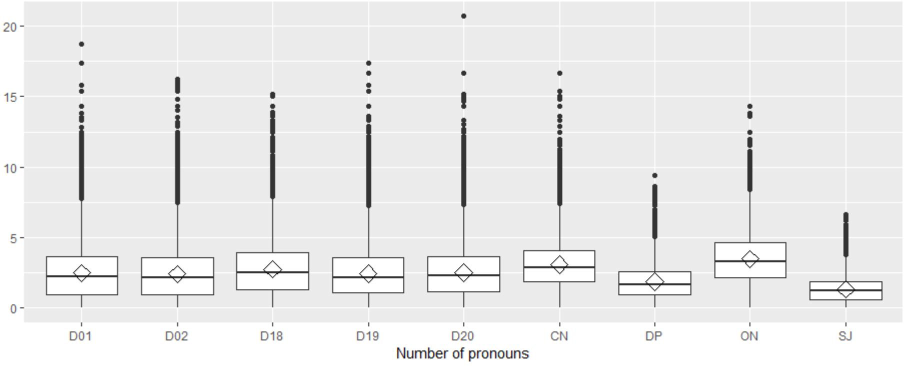


FIGURE C.8: Boxplot of number of pronouns per partition

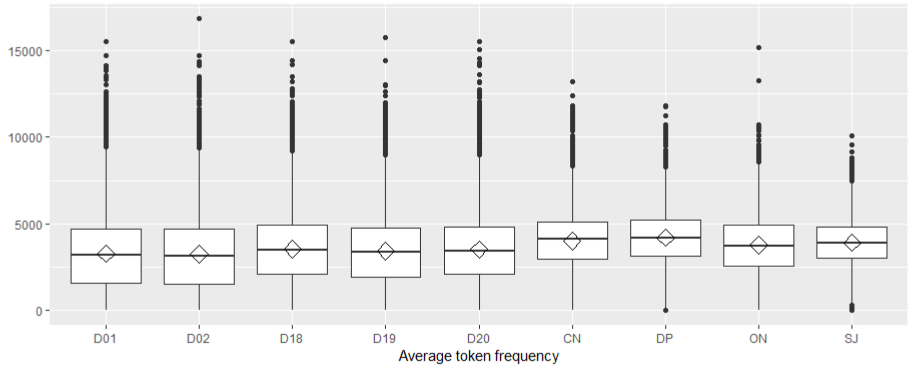


FIGURE C.9: Boxplot of average token frequency per partition

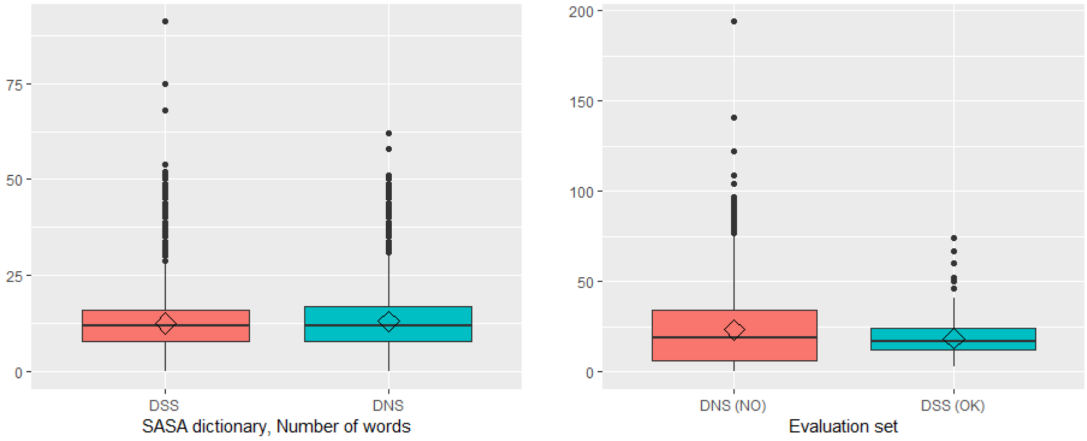


FIGURE C.10: Boxplot of number of words per language type partitions

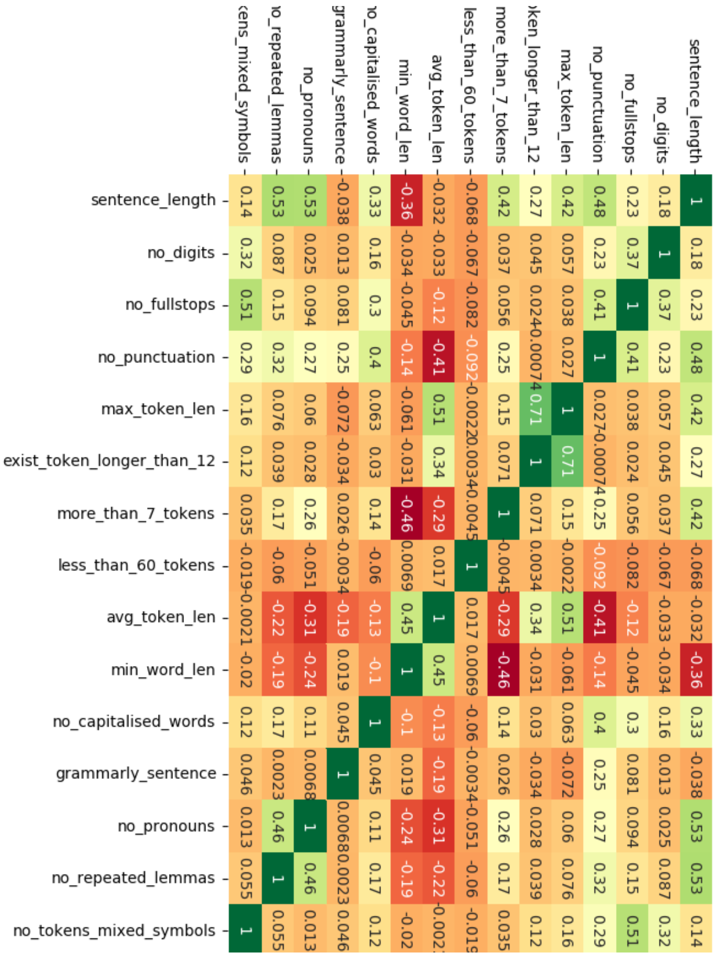


FIGURE C.11: The Pearson correlation matrix

D Authorship Identification of Short Messages

LISTING D.1: Regular expressions for stylistic features

```
class StylisticFeatures():

    regex_table = {
        'spaces_after_punctuation': \
            r'[%s] +' % re.escape(string.punctuation),
        'glued_sentences': \
            r'[^{0}]{0}[^{0}]' % re.escape(string.punctuation),
        'ne_joined_verb': r'ne[a-zA-Z]+',
        'non_capital_sent': r'\. ?[a-z]',
        'bad_dot': r'\.{2}|\.{4}',
        'bad_question': r'\?{2}|\?{4}',
    }
```

LISTING D.2: Regular expressions for character-based lexical features

```
class CharacterBasedFeatures():

    regex_table = {
        'exclamation_mark': r'!',
        'question_mark': r'\?',
        'dot': r'[^\.]*\.[^\.]*',
        'two_dot': r':',
        'comma': r',',
        'tab': r'\t',
        'lt': r'<',
        'gt': r'>',
        'proc': r'%',
        'or': r'\|',
        'curly_open': r'\{',
        'curly_closed': r'\}',
        'slash': r'\/',
        'backslash': r'\\',
        'at': r'@',
        'hashtag': r'#',
        'tilde': r'~',
        'plus': r'\+',
        'minus': r'\-',
        'times': r'\*',
        'dollar': r'\$',
        'hat': r'\^',
        'ampersand': r'\&',
        'underscore': r'\_',
    }
```

LISTING D.3: Regular expressions code for emoticons

```
class EmoticonFeatures():

    regex_table = {
        # smiley emoticons
        'smiley_wo_nose': (r':\)', 'smiley'),
        'smiley_w_nose': (r':-\)', 'smiley'),
        'smileys_wo_nose_reverse': (r'\({2,}:', 'smiley'),
        ...
        # happy emoticons
        'happies_wo_nose': (r':D{2,}', 'happy'),
        'happies_w_nose': (r':-D{2,}', 'happy'),
        'happy_wo_nose': (r':D', 'happy'),
        ...
        # sad emoticons
        'sad_wo_nose': (r':\(', 'sad'),
        'sad_w_nose': (r':-\(', 'sad'),
        'sad_oblique_wo_nose': (r':\[', 'sad'),
        ...
        # surprise emoticons
        'surprised_wo_nose_small': (r':o', 'surprised'),
        'surprised_wo_nose': (r':O', 'surprised'),
        'surprised_w_nose_small': (r':-o', 'surprised'),
        ...
        # kiss emoticons
        'kisses_wo_nose': (r':\*{2,}', 'kiss'),
        'kisses_w_nose': (r':-\*{2,}', 'kiss'),
        'kiss_wo_nose_closed': (r'x\*', 'kiss'),
        ...
        # wink emoticons
        'winks_wo_nose': (r';\){2,}', 'wink'),
        'winks_w_nose': (r';-\){2,}', 'wink'),
        'winks_happy_wo_nose': (r';D{2,}', 'wink'),
        ...
        # tongue emoticons
        'tongue_w_nose': (r':-P{2,}', 'tongue'),
        'tongue_w_nose_small': (r':-p{2,}', 'tongue'),
        'tongues_wo_nose_small': (r':p{2,}', 'tongue'),
        ...
        # skeptic emoticons
        'skeptics_wo_nose': (r':/{2,}', 'skeptic'),
        'skeptics_w_nose': (r':-/{2,}', 'skeptic'),
        'skeptic_wo_nose': (r':/', 'skeptic'),
        ...
        # others
        'relax_obliques': (r'=\){2,}', 'misc'),
        'relax_oblique': (r'=\)', 'misc'),
        'glasses_wo_nose': (r'8\){2,}', 'misc'),
    }
```

E Sentiment Classification of Short Messages

TABLE E.1: Example of some messages with their annotations

Body	Label
ae! :D cemo na fb da skupljamo ekipu? u koju cemo? :)))) Nisam mislila na zadatak, zadatak je interesantan. :)	POS
Eeeeeeeeeeeeeeeee bre, djubre prehlada. :/ Brande moj, cu li ti...	NEG
Kredit je dopunjen sa 200,00 din i vazi do 24.11.2016. Poštovani , vozilo 15 je na adresi. Vaš GOLUB TAXI	NEU

TABLE E.2: The translated messages in English from Table E.1

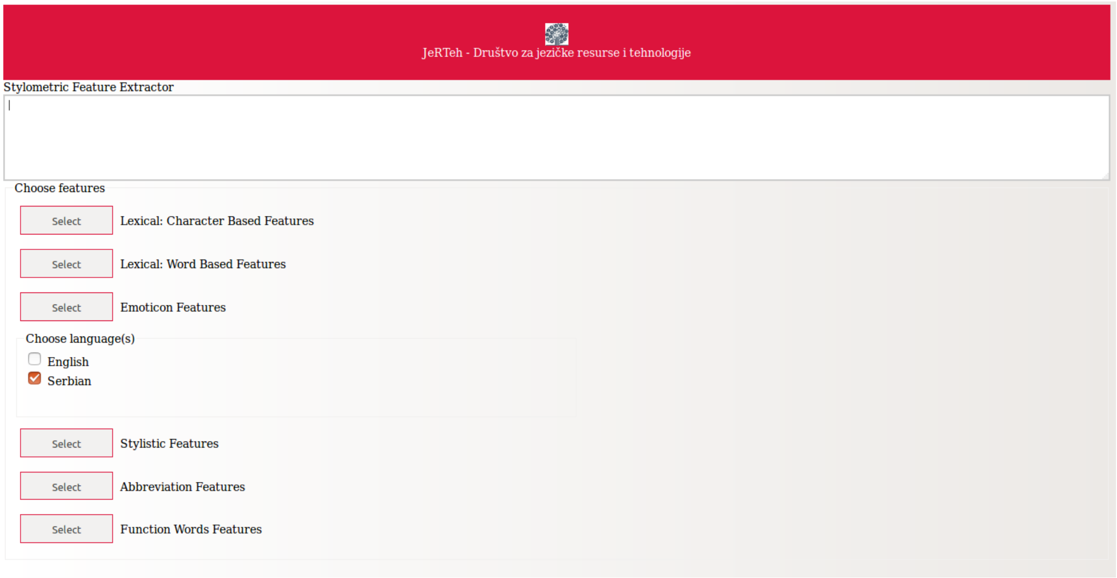
Body
Great! :D Are we gathering people on FB? Where shall we go? :)))) I was not thinking about the task, the task is interesting. :)
Noooooooooooo, stupid cold. :/ Have you heard about the news...
Your account has been reloaded with 200,00 RSD and it expires in 24.11.2016. To whom it may concern, the taxi 15 has arrived. Your GOLUB TAXI

TABLE E.3: Messages with confusing or multiple moods

Body	Label
Ozb, kako, gde? :) Ajd vazi, posalji link. Nisam znao :/ Spic braso	POS
Svasta :/ Zao mi je sto si se namucila :) Hvala ti...samo, ne znam koje drugo postoji :/ :)	NEG

TABLE E.4: Messages with confusing or multiple moods

Body
Really, how, where? :) OK, send me the link. I did not know :/ Top bro'
Nonsense :/ I am sorry that you put so much effort :) Thanks...but, I do not know which other is there :/ :)



□ English
☑ Serbian

FIGURE E.1: Web interface for feature extraction

Web Service

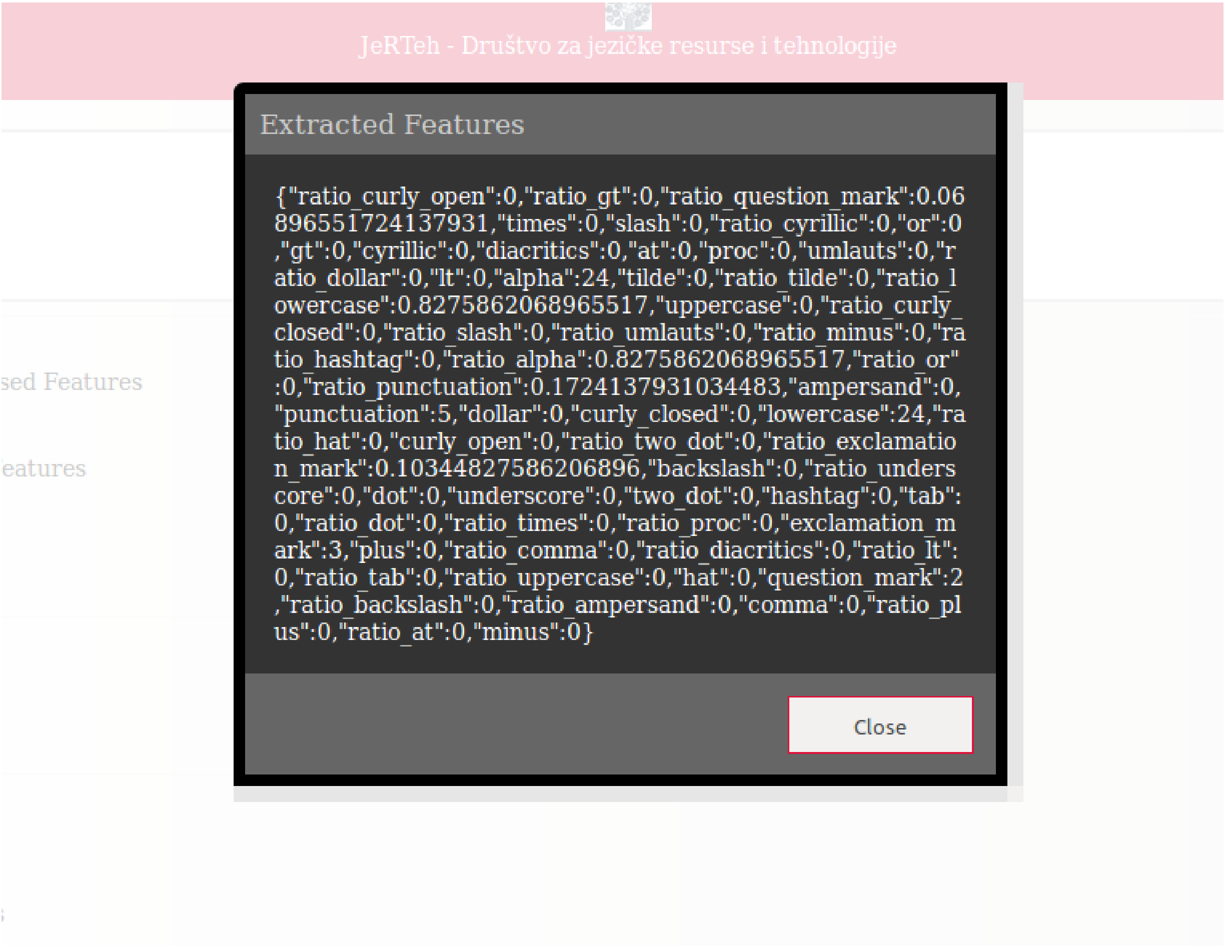
We developed a Web service and a corresponding Web interface, since these features are often used for many tasks, especially in tasks of Sentiment Classification (Derks, Bos, and Von Grumbkow, 2007; Neviarouskaya, Prendinger, and Ishizuka, 2009; Škorić, 2017) and Authorship Identification (Šandrih, 2018). The code was written in Python, and RESTful request dispatching was implemented using Flask micro-framework.[1] Most of the features are represented with corresponding regular expressions, as shown in Listings D.2, D.3 and D.1.

Web service for feature extraction can be used by sending POST requests to URIs listed in Table E.5.[2] Body of a request should be a JSON string containing text to be classified as a value of a key named *data*, and when features exist for Serbian and English, then the JSON object should also contain *lang_list* key.

For example, in order to extract emoticon features, corresponding Unix *curl* command would be:

[1] `http://flask.pocoo.org/`

[2] The service is hosted at `http://147.91.183.8:12348/`



```
{"ratio_curly_open":0,"ratio_gt":0,"ratio_question_mark":0.06
896551724137931,"times":0,"slash":0,"ratio_cyrillic":0,"or":0
,"gt":0,"cyrillic":0,"diacritics":0,"at":0,"proc":0,"umlauts":0,"r
atio_dollar":0,"lt":0,"alpha":24,"tilde":0,"ratio_tilde":0,"ratio_l
owercase":0.8275862068965517,"uppercase":0,"ratio_curly_
closed":0,"ratio_slash":0,"ratio_umlauts":0,"ratio_minus":0,"ra
tio_hashtag":0,"ratio_alpha":0.8275862068965517,"ratio_or"
:0,"ratio_punctuation":0.1724137931034483,"ampersand":0,
"punctuation":5,"dollar":0,"curly_closed":0,"lowercase":24,"ra
tio_hat":0,"curly_open":0,"ratio_two_dot":0,"ratio_exclamatio
n_mark":0.10344827586206896,"backslash":0,"ratio_unders
core":0,"dot":0,"underscore":0,"two_dot":0,"hashtag":0,"tab":
0,"ratio_dot":0,"ratio_times":0,"ratio_proc":0,"exclamation_m
ark":3,"plus":0,"ratio_comma":0,"ratio_diacritics":0,"ratio_lt":
0,"ratio_tab":0,"ratio_uppercase":0,"hat":0,"question_mark":2
,"ratio_backslash":0,"ratio_ampersand":0,"comma":0,"ratio_pl
us":0,"ratio_at":0,"minus":0}
```

FIGURE E.2: Resulting JSON of lexical feature counts

```
curl
-d '{
"data": "This is a happy message! :-)",
"lang_list": ["sr", "en"]
}'
-H "Content-Type: application/json"
-X POST http://147.91.183.8:12348/emoticon_features
```

Web application is available on-line.[3] The Web interface can be seen in Figure E.1. After entering text into the text area, user can select a group of features by clicking on a corresponding *Select* button. As a result a window pops up, with text in JSON having feature names as keys and their counts as values, as shown in Figure E.2.

[3] Web application for the extraction of lexical, syntactic and stylistic features for Sentiment Classification, http://features.jerteh.rs/

TABLE E.5: Feature Extraction via Web service

URI	Language list	Description
`/char_based_features`	no	Char-based lexical features
`/word_based_features`	no	Word-based lexical features
`/emoticon_features`	no	Emoticon syntactic features
`/abbreviation_features`	yes	Slang abbreviation syntactic features
`/stylistic_features`	no	Stylistic features
`/functionword_features`	yes	Counts of function words

Biografija autora

Branislava Šandrih je rođena 19.08.1991. u Pančevu. Zaposlena je kao asistent na Katedri za bibliotekarstvo i informatiku pri Filološkom fakultetu u Beogradu. Nakon završene srednje Elektrotehničke škole „Nikola Tesla" u Pančevu, i sticanja zvanja elektrotehničar računara, 2010. godine upisuje osnovne studije na Matematičkom fakultetu. Četiri godine kasnije stiče zvanje diplomirani matematičar na smeru Računarstvo i informatika sa prosečnom ocenom 9.48, a godinu dana kasnije i zvanje master matematičar sa prosečnom ocenom 9.80 na istom smeru. Tokom studija, bila je višestruki dobitnik različitih stipendija i nagrada. Godine 2015. upisuje doktorske studije na smeru Informatika na Matematičkom fakultetu.

U međuvremenu je aktivno angažovana na projektu Ministarstva prosvete, nauke i tehnološkog razvoja pod nazivom „Srpski jezik i njegovi resursi: teorija, opis i primene" (ON 178006). Član je Društva za JEzičke Resurse i TEHnologije (JeRTeh) sa sedištem u Beogradu. Kao spoljni istraživač, uključena je u rad Istraživačke grupe za računarsku lingvistiku u Vulverhamptonu, Velika Britanija. Pored maternjeg, tečno govori engleski, nemački i španski jezik.

Прилог 1.

Изјава о ауторству

Потписани-а __Бранислава Шандрих______

број уписа __2019/2015______

Изјављујем

да је докторска дисертација под насловом

__**IMPACT OF TEXT CLASSIFICATION ON NATURAL LANGUAGE PROCESSING APPLICATIONS** [УТИЦАЈ КЛАСИФИКАЦИЈЕ ТЕКСТА НА ПРИМЕНЕ У ОБРАДИ ПРИРОДНИХ ЈЕЗИКА]______

- резултат сопственог истраживачког рада,
- да предложена дисертација у целини ни у деловима није била предложена за добијање било које дипломе према студијским програмима других високошколских установа,
- да су резултати коректно наведени и
- да нисам кршио/ла ауторска права и користио интелектуалну својину других лица.

Потпис докторанда

У Београду, _________________

____

_

Прилог 2.

Изјава о истоветности штампане и електронске верзије докторског рада

Име и презиме аутора __Бранислава Шандрих______

Број уписа __2019/2015______

Студијски програм __Информатика______

Наслов рада __**IMPACT OF TEXT CLASSIFICATION ON NATURAL LANGUAGE PROCESSING APPLICATIONS** [УТИЦАЈ КЛАСИФИКАЦИЈЕ ТЕКСТА НА ПРИМЕНЕ У ОБРАДИ ПРИРОДНИХ ЈЕЗИКА]______

Ментор __проф. др Александар Картељ______

Потписани ____Бранислава Шандрих______

изјављујем да је штампана верзија мог докторског рада истоветна електронској верзији коју сам предао/ла за објављивање на порталу **Дигиталног репозиторијума Универзитета у Београду.**

Дозвољавам да се објаве моји лични подаци везани за добијање академског звања доктора наука, као што су име и презиме, година и место рођења и датум одбране рада.

Ови лични подаци могу се објавити на мрежним страницама дигиталне библиотеке, у електронском каталогу и у публикацијама Универзитета у Београду.

Потпис докторанда

У Београду, _________________________

____

Прилог 3.

Изјава о коришћењу

Овлашћујем Универзитетску библиотеку „Светозар Марковић“ да у Дигитални репозиторијум Универзитета у Београду унесе моју докторску дисертацију под насловом:

IMPACT OF TEXT CLASSIFICATION ON NATURAL LANGUAGE PROCESSING APPLICATIONS [УТИЦАЈ КЛАСИФИКАЦИЈЕ ТЕКСТА НА ПРИМЕНЕ У ОБРАДИ ПРИРОДНИХ ЈЕЗИКА]

која је моје ауторско дело.

Дисертацију са свим прилозима предао/ла сам у електронском формату погодном за трајно архивирање.

Моју докторску дисертацију похрањену у Дигитални репозиторијум Универзитета у Београду могу да користе сви који поштују одредбе садржане у одабраном типу лиценце Креативне заједнице (Creative Commons) за коју сам се одлучио/ла.

[✓ 1. Ауторство]

2. Ауторство - некомерцијално

3. Ауторство – некомерцијално – без прераде

4. Ауторство – некомерцијално – делити под истим условима

5. Ауторство – без прераде

6. Ауторство – делити под истим условима

(Молимо да заокружите само једну од шест понуђених лиценци, кратак опис лиценци дат је на полеђини листа).

Потпис докторанда

У Београду, ________________________

____

1. Ауторство - Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце, чак и у комерцијалне сврхе. Ово је најслободнија од свих лиценци.

2. Ауторство – некомерцијално. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела.

3. Ауторство - некомерцијално – без прераде. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела. У односу на све остале лиценце, овом лиценцом се ограничава највећи обим права коришћења дела.

4. Ауторство - некомерцијално – делити под истим условима. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца не дозвољава комерцијалну употребу дела и прерада.

5. Ауторство – без прераде. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца дозвољава комерцијалну употребу дела.

6. Ауторство - делити под истим условима. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца дозвољава комерцијалну употребу дела и прерада. Слична је софтверским лиценцама, односно лиценцама отвореног кода.