TakeLab

Laboratorij za analizu teksta i inženjerstvo znanja
Text Analysis and Knowledge Engineering Lab
Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva
Unska 3, 10000 Zagreb, Hrvatska

UNIVERSITY OF ZAGREB
FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

BSc THESIS No. 5324

Intrinsic Plagiarism Detection in Student Theses

David Lozić

Zagreb, June 2017

UNIVERSITY OF ZAGREB
FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING
BACHELOR THESIS COMMITTEE

Zagreb, 3 March 2017

BACHELOR THESIS ASSIGNMENT No. 5324

Student:	**David Lozić (0036485502)**
Study:	Computing
Module:	Computer Science

Title: **Intrinsic Plagiarism Detection in Student Theses**

Description:

Plagiarism detection is an authorship analysis task that aims at determining the originality of text using natural language processing techniques. In intrinsic plagiarism detection, the plagiarized text is detected based on a statistical stylometric analysis of the text. Namely, the inconsistencies in statistical style features of the different text fragments indicate what parts of text might have been plagiarized. In contrast to extrinsic procedures, intrinsic plagiarism can be used even in cases when the original text is not available.

The topic of this thesis is the intrinsic plagiarism detection for Croatian language based on the analysis of stylometric features and machine learning. Do a literature survey on methods for intrinsic plagiarism detection as well methods for computational stylometry analysis. Devise a system for intrinsic plagiarism detection in student theses. Compile a suitable test collection of student theses' texts composed of texts from a number of different authors. Implement the system and carry out an experimental evaluation on the test collection. Design an application programming interface in such a way that the system can be used as a stand-alone module. All references must be cited, and all source code, documentation, executables, and datasets must be provided with the thesis.

Issue date:	10 March 2017
Submission date:	9 June 2017

Mentor:

Associate Professor Jan Šnajder, PhD

Committee Chair:

Full Professor Siniša Srbljić, PhD

Committee Secretary:

Assistant Professor Tomislav Hrkać, PhD

CONTENTS

1. Introduction

Plagiarism is defined by (Stevenson, 2010) as taking someone else's work or ideas and passing them off as one's own, and has been hindering academic progress for centuries. The explosion of the World Wide Web has made intellectual theft easier than ever. This includes anything from basic copy-paste methods to more elaborate falsification and manipulation.

The two main approaches to this problem are intrinsic and extrinsic plagiarism detection. While extrinsic detection relies on external sources to expose plagiarism, the goal of intrinsic plagiarism detection is to uncover theft without the aid of external references by analyzing the discrepancies within a single corpus. This is a difficult and tedious task for humans. Readers should not be concerned with proving the originality of a paper or thesis. It would be very beneficial if we could leave these tedious tasks to machines and focus our valuable time elsewhere.

While the recent technological advances have made plagiarism easier, they also provided us tools to combat these issues. Machine learning, as first defined by Arthur Samuel, is the branch of computer science which gives computers the ability to learn without being explicitly programmed. More specifically, natural language processing (NLP) is a branch of artificial intelligence which allows computers to understand natural language – any language spoken by humans which evolved naturally.

Depending on the approach, the problem can both be classified as supervised or unsupervised learning. Unsupervised learning is any machine learning task which tries to draw inferences from an unlabeled dataset, while supervised learning uses a known dataset to make predictions.

The goal of this thesis is to create an intrinsic plagiarism detection system for the Croatian language. Because no labeled dataset existed, an artificial dataset was compiled by mixing previous TakeLab student thesis, most of which focus on NLP. Feature extraction was performed on this dataset and two main approaches were considered. The first model applies outlier detection on each document of the artificial dataset, marking outliers as plagiarisms. The second model tests a range of classifiers on the features extracted from the entire dataset. A classic 70/30 train-test split was employed, and the hyperparameter optimization of the classifiers was done using 3-fold cross-validation on the training dataset. Several metrics were used for evaluation, but an emphasis was placed on the F1 score of the plagiarized class.

The following chapter describes previous work done on the topic. After that, a detailed description of the dataset is provided. The features and the two models used in this system are presented in chapter four. Chapter five focuses on experiments and evaluation of the model, where the different metrics and results are explained. The final chapter is a brief conclusion to the thesis, providing some thoughts and ideas for future work.

2. Related Work

To the best of my knowledge, there has not been an attempt at intrinsic plagiarism detection for the Croatian language. There have been, however, many studies which tackled the problem of intrinsic plagiarism detection for the English language.

The concept of averaged word frequency class first introduced in (Zu Eissen i Stein, 2006) was explored in this system. As cited from the paper, the power of a plagiarism approach depends on the quality of the quantified linguistic features. The features used in the paper fall into one of five basic categories: text statistics (character level), syntactic features (sentence-level), part-of-speech, closed-class word sets, and structural features. The most prominent of these features was the averaged word frequency class, which was used in this thesis and is described in more detail in chapter 4. The averaged word frequency class tries to capture the author's vocabulary size and style complexity. In the paper, a base corpus was constructed from the ACM digital library. In the artificial dataset, passages were copied or reformulated from other ACM computer science articles. In their experiments, 450 documents were generated, each of which contained between 3 and 6 plagiarized passages. The documents were later split into 50-100 parts. The parts were relatively small and consisted of 40 to 200 words. A feature matrix was constructed from these parts and discriminant analysis and SVM classification were applied. This paper did not explore discriminant analysis, but one of the models does focus on SVM classification.

A character n-gram approach was taken in (Stamatatos, 2009). A window of character 3-grams with length 1000 and step 200 is slid across the document. The windows are compared to the whole document based on a dissimilarity function. The function is applied to the *profile* of the text, which is is a vector of normalized frequencies of the character 3-grams in the text. The anomalies of the function are used for plagiarism detection. While the described approach was not explored in this thesis, the sliding window approach described in the paper is similar to the sliding window used in this thesis. The difference is that this thesis explores the windows on a sentence level, while the one described in the paper focuses on the character level. In the paper an additional pre-processing step to plagiarism detection is added, where a certain criterion is defined which indicates that the document is plagiarism-free. If the criterion is met, the passage level plagiarism detection is skipped. This thesis does not explore plagiarism detection on the document level.

Most of the recent notable studies are a product of the PAN competitions. (CLEF, 2017) PAN is a series of scientific events and shared tasks focused on digital text forensics. Tasks are split into three main groups: Authorship, Originality, and Trust. Authorship includes problems such as author identification and author profiling. Trusts mainly concern credibility analysis, such as vandalism detection. Finally, originality focuses on plagiarism detection tasks.

These competitions have produced many papers focusing on intrinsic plagiarism detection. This system is similar to the one described in (Zechner et al., 2009) and tries to apply some of these methods in the Croatian language. The system described in the paper works on a sentence level. A sliding window approach was taken, where windows of length k were generated. Feature vectors are created from features such as the previously mentioned averaged word frequency class, punctuation counts, part-of-speech tags, the number of pronouns and the number of certain stop words. Outlier detection was performed using the cosine distance of the feature vectors and outliers were marked as plagiarisms. The first model in this thesis is similar to the system described in the paper.

3. Dataset

The dataset was provided by TakeLab and consisted of 85 previous bachelor and master's thesis written in Croatian and English. Because this system is focused on the Croatian language, the English documents were not considered. The topic domain of the documents is limited to computer science, specifically NLP. Intrinsic plagiarism detection on such a dataset proved to be a difficult task. Scientific thesis often include many table descriptions, result analyses and such, which do not allow for the author's individual style to stand out.

To combat these issues, the approach described in Subsection 3.3. proved to be efficient. I believe this system would largely benefit from a more homogenous dataset such as essays, book reports, and similar texts.

3.1. Preprocessing

The dataset was provided in pdf format, so it first needed to be converted to plaintext to do any kind of processing. This conversion was accomplished using the Textract tool (Malmgren, 2017). While most of the files were converted correctly, some had problems with Unicode characters and others were completely unreadable. The Unicode problems were mostly linked to the Croatian diacritics: č, ć, đ, ž, š, and ligatures – multiple letters joined into a single glyph. Ligature normalization was done using the NLTK library (Loper i Bird, 2002). The unrecognizable files were discarded, which left out 77 documents.

The second step of preprocessing includes removing irregularities which aren't linked to the author's writing style. These irregularities include headings, tables, image and table captions, lists, references and other deformities. While the regular expressions used to filter out these artifacts did the majority of the work, some deformities remained. A quick manual dataset cleaning was performed to finalize this step.

3.2. Sliding windows

The system uses a sliding window approach in which a window of specified type and length is slid across the text.The distance between windows is specified by a certain stride. The window length and stride are determined by the window type, which can be sentence, n-gram or character n-gram. For example, n-gram windows of length 5 and stride 3 contain 5 n-grams and are distanced by 3 n-grams. This converts every document into a list of passages which are later used for feature extraction.

While the system supports other types of windows such as n-grams or character n-grams, sentence windows of both length and stride 5 were produced for feature extraction.

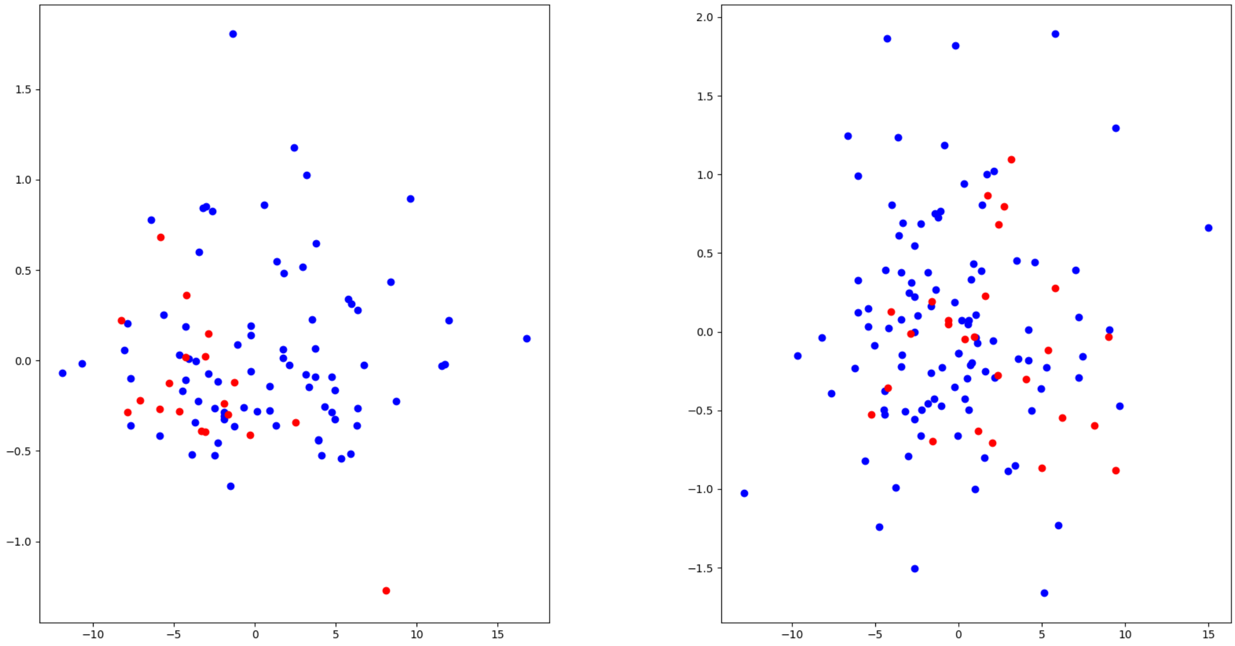
3.3. Artificial plagiarism dataset creation

Since there was no available labeled dataset for the task, an artificial dataset was created by injecting parts of a single donor document into the original. After the sliding window step, windows from a donor document are mixed with the windows from the original document so that the merged document contains anywhere between 0% and 50% of plagiarized windows. The original windows are labeled as 0, while the plagiarized windows are labeled as 1. The donor is selected by random for each original.

Since we want to focus on the core writing style of the different authors, outlier elimination was performed before merging. This was done to remove any possible irregularities left from the preprocessing step and to select windows that best represent the author's writing style.

We can see in Figure 3.1. that, after applying principal component analysis, the plagiarized window in the bottom of Example 1 may not be the best representation of the writing style of the donor document author. Similarly, the top original window may not be the best representation of the original author. Principal component analysis (PCA) is a method of reducing the dimensionality of a matrix. (Wold et al., 1987) Red dots represent plagiarized windows while the blue dots represent original windows.

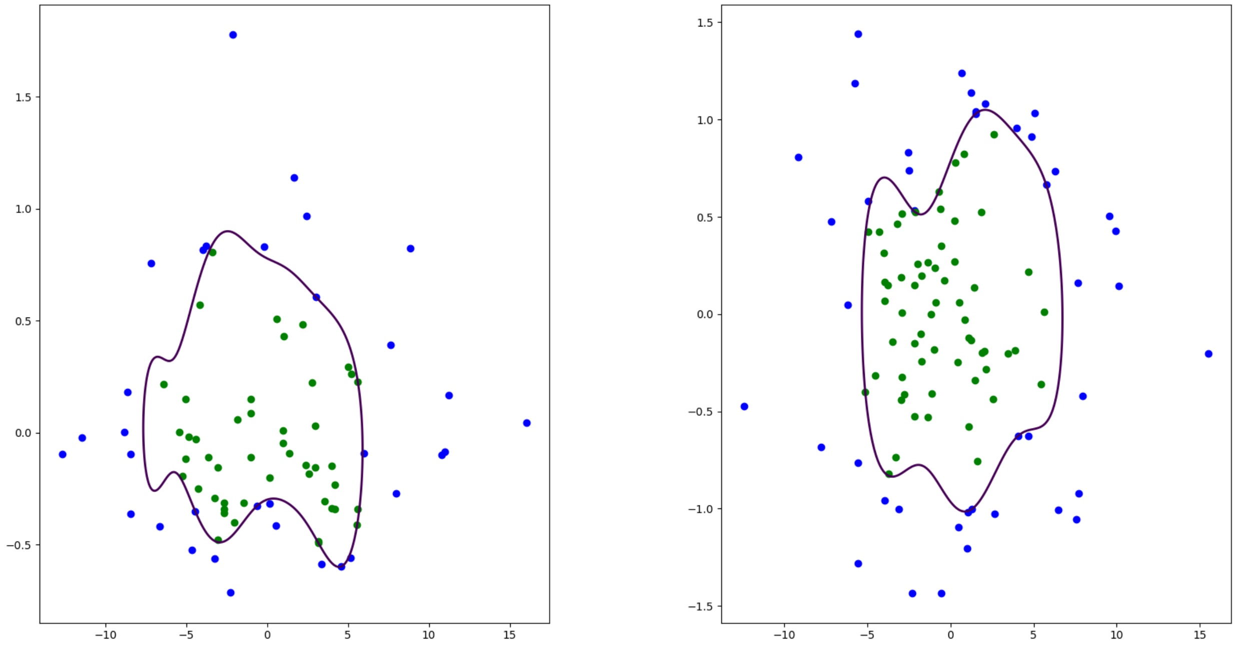
To address this, a homogenous dataset was created by performing outlier elimination before merging. This was done using the `OneClassSVM` from the scikit-learn package (Pedregosa et al., 2011) and can be seen in Figure 3.2.



(a) Example 1

(b) Example 2

Figure 3.1: Mixed documents before removing outliers

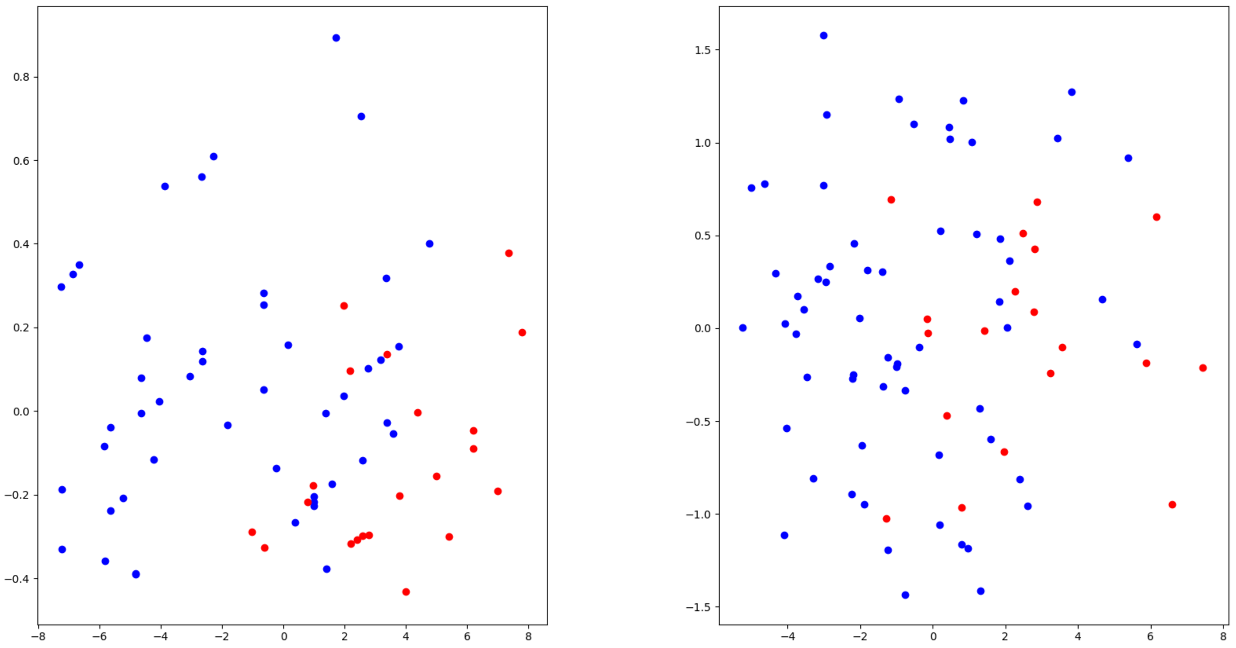


(a) Example 1

(b) Example 2

Figure 3.2: Outlier removal

The intuition behind this was to focus on the core writing style of the author and to make a clearer separation between the two groups. This would make it easier for the classifiers to recognize plagiarized passages. Figure 3.3 shows the previous two examples merged after outlier elimination. The left example in Figure 3.3 shows a clearer separation from the two authors. The right example, while not as good as the left one, is still better than the equivalent example from the first dataset.



(a) Example 1

(b) Example 2

Figure 3.3: Mixed documents after removing outliers

3.4. Dataset Statistics

While the system is very flexible at creating an artificial dataset, the characteristics of the explored dataset are as follows. There are 77 documents in the dataset. The windows generated contain 5 sentences. The maximum number of windows is 149, while the minimum number of windows generated in a document is 19. The average number of windows in the entire dataset is 60. There is a total of 4668 windows, 3678 of which are originals and 990 are plagiarized. The train set includes 53 documents containing 3362 windows, 2677 of which are originals and 685 are plagiarized. The test set has 24 documents containing 1306 windows, 1001 of which are originals and 305 are plagiarized.

Here is an example of a original and a plagiarized window from a document in the artificial dataset:

```
   [’Metoda je donijela poboljšanja s obzirom na
referentnu metodu, no i dalje su rezultati istojezičnog
korpusa mnogo bolji.’, ’U budućem radu bilo bi poželjno
testirati programsko ostvarenje nad novim korpusom
kako bi se dokazala modularnost sustava, a kao moguće
poboljšanje svakako bi bilo uvođenje značajki koje sadrže
prijevod na hrvatski jezik.’, ’Cilj ekstrakcije složenih
kratica hrvatskoga jezika je razvitak tehnika koje
automatski ekstrahiraju kratice i njihove pripadajuće
ekspanzije iz teksta.’, ’Razvijena su tri različita
pristupa:  referentna metoda, metoda potpornih vektora te
njihova kombinacija.’, ’Više je pristupa korišteno kako
bi se mogli usporediti te iz njih izvuči zaključci.’]  0
```

```
   [’Ovaj sustav je najjednostavniji primjer tipiziranog
-računa, pa je posebno pogodan za demonstriranje ključnih
pojmova i koncepata općenitih tipiziranih sustava.’,
’Uz terme, potrebno je definirati i tipove.’, ’Za
početak smatramo da imamo skup baznih tipova, koji
može biti proizvoljan neprazan skup.’, ’Za generiranje
kompleksnijih tipova koriste se konstruktori tipova.’,
’U jednostavno tipiziranom -računu postoji samo jedan
konstruktor tipova, u oznaci ”.’]  1
```

4. Model

Several approaches were considered in this system. The main difference is that the first group of models works on a document level, while the second group analyses the entire dataset.

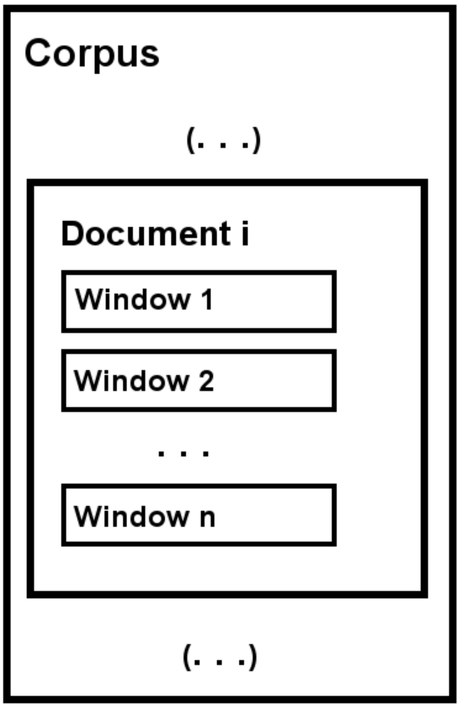


Figure 4.1: Model 1

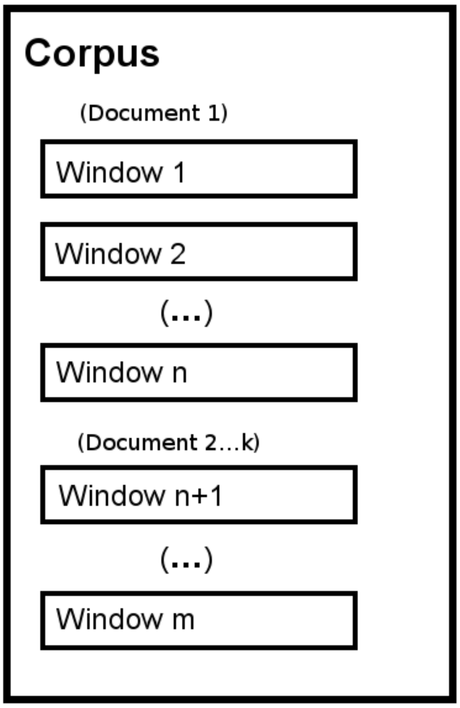


Figure 4.2: Model 2 – corpus contains m windows from k documents

The first model iterates through every document in the dataset and performs outlier detection with the one-class SVM. Features are extracted from each window of the document and the outliers are marked as plagiarisms. Figure 4.1. shows the input data structure for the first model. Windows are grouped into documents which are processed separately. Notice that there is no training involved in this model, as every prediction is done on a per-document basis. This can be labeled as an unsupervised learning algorithm.

In the second model, the windows from the entire dataset are processed at once. The input data structure for the second model is shown in Figure 4.2. The second model is a supervised learning task.

Both models work with two types of features which are explained in the following subsection.

4.1. Features

In the first model, separate feature matrices are constructed for every document. The second model uses a single feature matrix created from every window in every document.

In the second model, two approaches were considered. In the first approach, *comparison features* are created by dividing features extracted from every window with features from the documents to which they belong. In the second approach, every window is paired with the average features from the document to which it belongs. The new feature vector is a concatenation of the window feature vector and the document feature vector. These features are later referred to as *concatenation features*.

Counting features. These include the average sentence length and the average number of punctuations within a window. While simple, these features proved to be very effective. The token count within every sentence is taken and averaged within a 5-sentence window. Similarly, the occurrence of punctuation characters is measured. Each of these features represents a single dimension in the final feature vector.

Part of speech. (POS) The Croatian POS tagger developed by (Agić et al., 2013) was used. Every token within each sentence of a window is assigned a POS tag. Attributes like tense, number, and gender were discarded which left out labels such as verb, adjective, noun, etc. The total number of each tag within a window is measured which results in a 12-dimensional feature vector.

Averaged word frequency class. As described in (Zu Eissen i Stein, 2006), a word's frequency class $c(w)$ is defined as $log_2(f(w^*)/f(w))$, where $f(w)$ denotes the frequency of word w and w^* represents the most frequent word in the document. The averaged word frequency class tries to capture the author's vocabulary size and writing style.

Common words and phrases. During data processing a trend quickly became apparent. Some authors demonstrated a heavy usage of certain phrases and words. These features try to capture those observations by using the number of the 10 most common 2-grams, 3-grams and 4-grams and the number of the 20 most common unique tokens. Unique tokens do not include stop words and punctuations.

4.2. Classifiers

The following list of classifiers from the scikit-learn package (Pedregosa et al., 2011) were used: `LinearSVC`, `SVC`, `RandomForestClassifier`, `SGDClassifier`, `OneClassSVM`, and `LogisticRegression`. These classifiers were trained and tested with a wide range of hyperparameters.

The first model only uses the one-class SVM and does not need training. Features are extracted from each document and outliers are marked as plagiarisms. The second model is fitted on the entire training dataset.

5. Experimental Evaluation

This section describes the experimentation, evaluation, and results of the outlier detection model and the classification model.

5.1. Experimental Setup

The artificial dataset is split into a 70/30 train-test split. Each document in the artificial dataset contains anywhere between 0% and 50% of plagiarized windows. Since the first model does not require training, all of the experiments were performed on the test set. The second model uses a range of classifiers with tuned hyperparameters. The hyperparameter tuning was performed using sklearn's `GridSearchCV` with 3-fold cross-validation, where the training set is split into 3 subsamples. Of the 3 subsamples, one is used for validation and the rest are used for training. This process is repeated 3 times.

5.2. Metrics

To analyze results we need to understand the metrics which were used. A simple test of accuracy is not very relevant since the amount of plagiarized passages is relatively small compared to the originals.

To get a clearer representation of the results, we use *precision*, *recall*, and the *F1* score. Precision is defined as the number of true positives (tp) divided by the number of true positives plus the number of false positives (fp).

$$Precision = \frac{tp}{tf + fp}$$

Recall is defined as the number of true positives (tp) divided by the number of true positives plus the number of false negatives (fn).

$$Recall = \frac{tp}{tp + fn}$$

The $(F1)$ score is defined as the harmonic mean of precision and recall.

$$F1 = 2\frac{Precision \times Recall}{Precision + Recall}$$

A visual explanation for precision and recall is provided in Figure 5.1.

If we label original passages as positives and plagiarized passages as negatives, then tp would be the number of correctly classified originals, fp would be the number of falsely classified originals, tn would be the number of correctly classified plagiarisms, and fn would be the number of falsely classified plagiarisms.

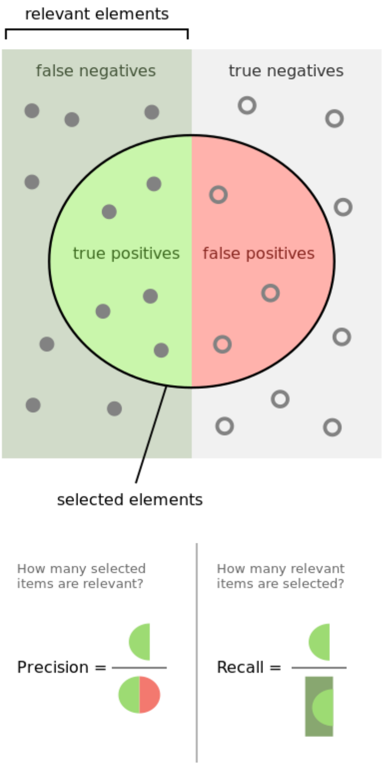


Figure 5.1: A visual explanation of precision and recall (Wikipedia, 2017)

Since the classes are unbalanced, precision, recall, and F1 would seem very high if originals are labeled as positives. For better analysis, both scenarios with positive originals and positive plagiarisms are evaluated. Emphasis is placed on the F1 score when plagiarisms are positive. This is later referred to as the *plagiarism F1* score.

5.3. Model 1

In this model, a feature matrix is constructed for every document. The model then performs outlier detection on the extracted features.

5.3.1. Experimentation

Figures 5.2 and 5.3 show the features and one-class SVM decision boundaries plotted in a two-dimensional space after applying PCA. As shown in Figure 5.2, the regular feature set successfully selparates plagiarized windows from the originals in a single document. Figure 5.3 shows the effect of comparison features on the same model and different documents.

While the groups are successfully separated and the model performs relatively well, the plagiarized windows are not really outliers to the originals. This indicates that clustering methods might be a better approach.

5.3.2. Feature Analysis

Features removed	Plagiarism F1
None	0.27
Counting features	0.28
Averaged word frequency class	0.25
POS features	0.25
Phrase features	0.25
POS + phrase features	0.3

Table 5.1: Plagiarism F1 scores for removed features

Table 5.1 shows the effect of removing certain features on the plagiarism F1 score. Removing POS and phrase features yields the best result. These features have been chosen because they are the most computationally expensive.

5.3.3. Scores

The following results are for the reduced subset of features.

Table 5.2 shows the confusion matrix for regular and comparison features when the original windows are marked as positives. It provides a good indication of how the model handles certain scenarios.

Features	TP	FP	TN	FN
Regular features	756	123	72	220
Comparison features	753	124	71	223

Table 5.2: Confusion matrix

We can see that the model has correctly classified 72 plagiarized passages out of 195, but has also incorrectly classified 220 originals out of 976.

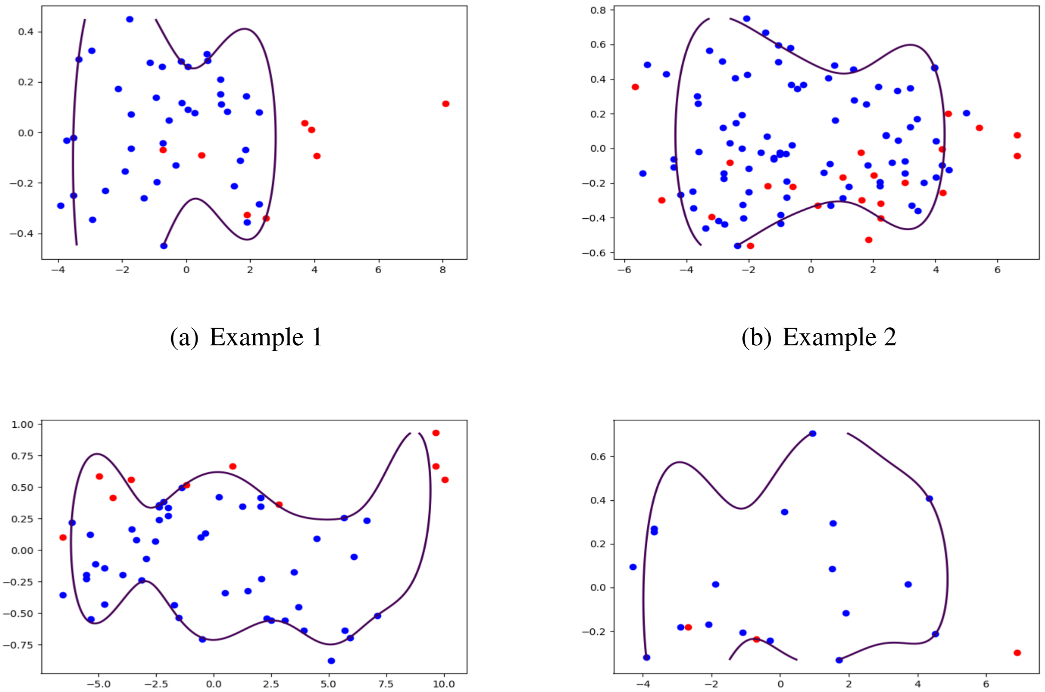
Features	Precision	Recall	F1
Regular features	0.86	0.77	0.82
Comparison features	0.86	0.77	0.81

Table 5.3: Model 1 scores with originals as positives

Tables 5.3 and 5.4 provide scores of both regular and comparison features. We can observe that there is no significant difference in scores for the two groups of features.

Features	Precision	Recall	F1
Regular features	0.25	0.37	0.30
Comparison features	0.24	0.36	0.29

Table 5.4: Model 1 scores with plagiarisms as positives



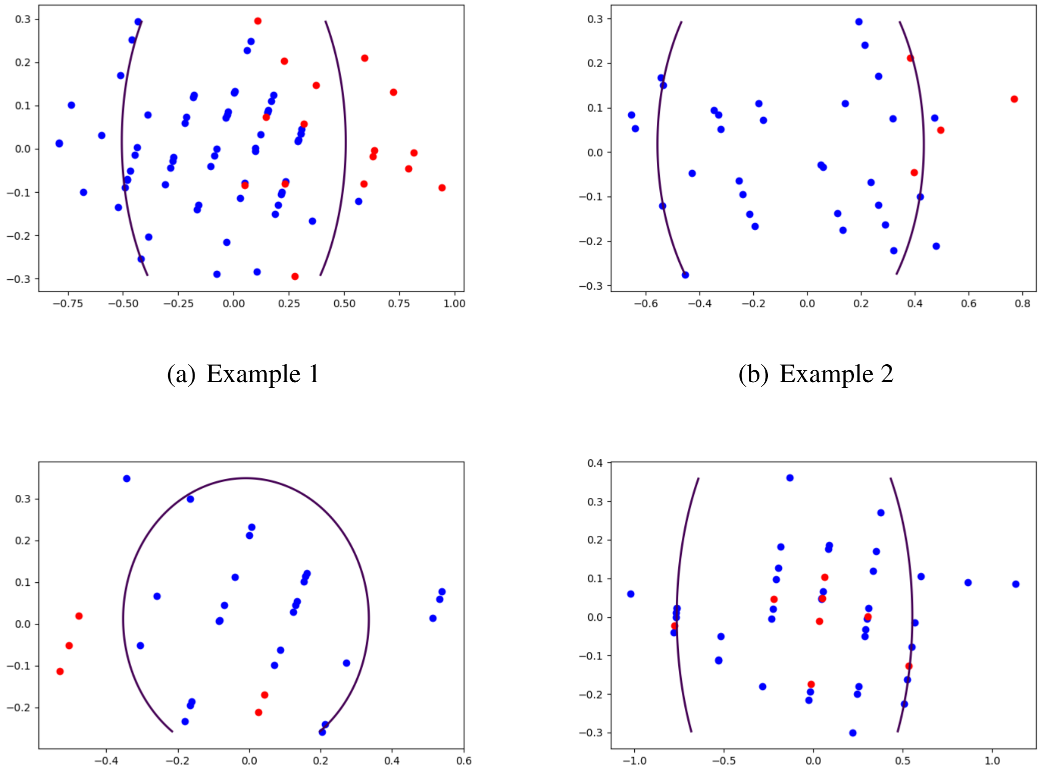
(a) Example 1

(b) Example 2

(c) Example 3

(d) Example 4

Figure 5.2: One-class SVM with regular features detecting outliers



(a) Example 1

(b) Example 2

(c) Example 3

(d) Example 4

Figure 5.3: One-class SVM with comparison features detecting outliers

5.4. Model 2

The second model uses a feature matrix constructed by extracting features from the entire dataset. Two separate feature matrices were created. The first group of features is referred to as comparison features, where features extracted from every window of a document are divided by the average features of the entire document. The second feature matrix is referred to as the concatenation matrix, where the features from the window and its document are concatenated.

5.4.1. Experimentation

Figure 5.4 shows comparison features extracted from the test corpus after applying PCA and the decision boundaries of multiple classifiers. We can see that the original and plagiarized passages are grouped very closely, with a few extreme outlier plagiarisms to the top and right.

The strict decision boundaries of these classifiers seem to be effective at detecting extreme cases of plagiarisms, but fail to differentiate between the two groups.

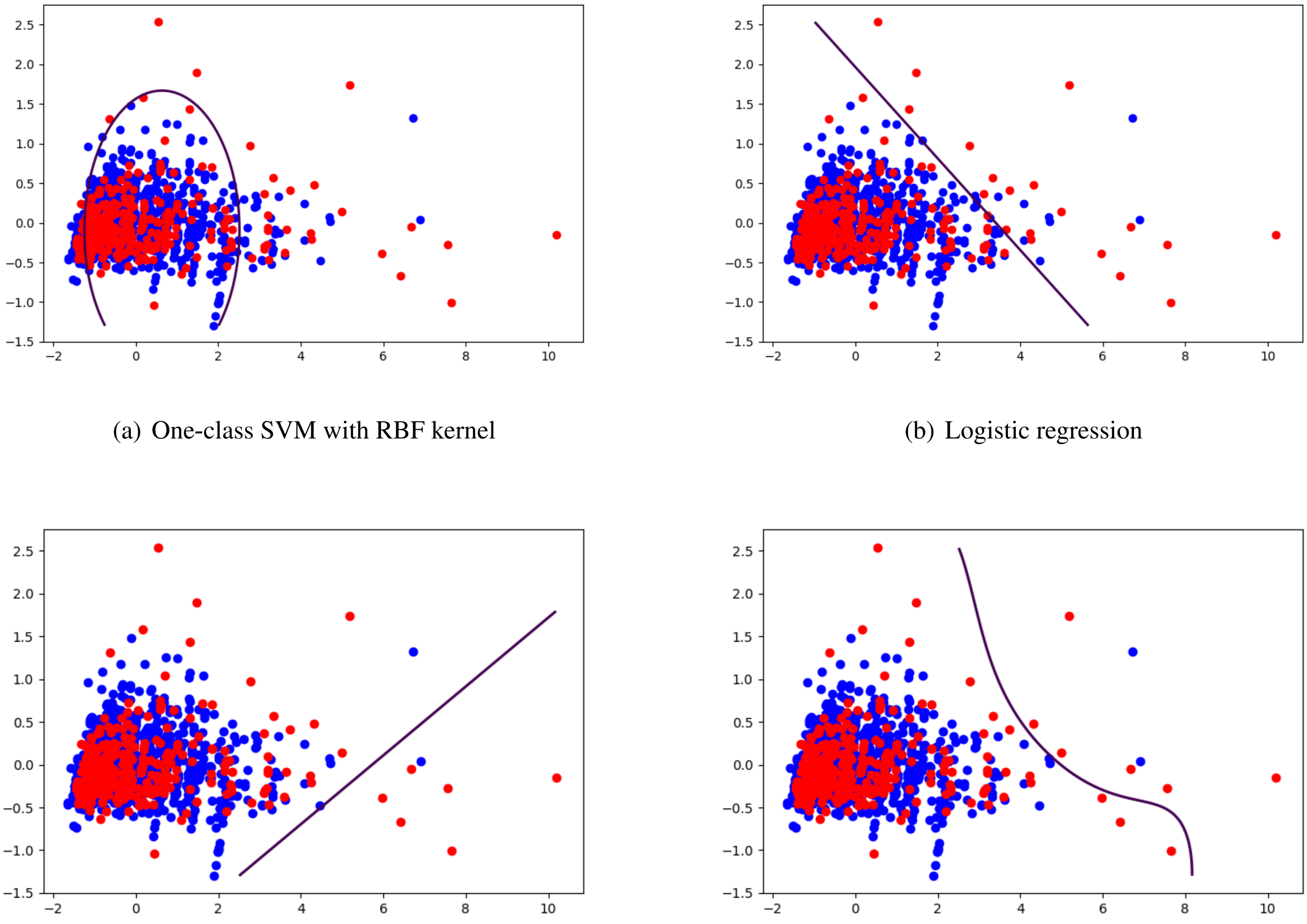
5.4.2. Feature Analysis

	Plagiarism F1		
Features removed	One-class SVM	LR	RF
None	0.22	0.32	0.06
Counting features	0.16	0.19	0
Averaged word frequency class	0.23	0.33	0.03
POS features	0.18	0.15	0.11
Phrase features	0.21	0.13	0.14
POS + phrase features	0.25	0.09	0.2

Table 5.5: Model 2 feature analysis on multiple classifiers

Table 5.5 shows us the impact of removing certain features on the plagiarism F1 score for multiple classifiers.

We can see that, similar to the first model, the removal of POS and phrase features results in the highest score for the One-class SVM and random forest classifier, but the lowest score in the logistic regression classifier. We can also observe the importance of the counting features, which when removed, yield lower scores for all classifiers.



(a) One-class SVM with RBF kernel

(b) Logistic regression

(c) SGD

(d) SVM with RBF kernel

Figure 5.4: Decision boundaries from multiple classifiers

5.4.3. Scores

Scores are evaluated for comparison and concatenation features separately.

Classifier	TP	FP	TN	FN
Logistic regression	964	277	28	37
One-class SVM	914	248	57	87
SVM	998	296	9	3
SGD	679	220	85	322
Random forest	962	266	39	39

Table 5.6: Confusion matrix for the second model with comparison features

Table 5.6 shows the confusion matrix for multiple classifiers using the comparison features. Compared to the first model, the second labels fewer plagiarisms, but with a greater accuracy, excluding the SGD classifier.

Notice that these values do not correspond to the decision boundaries in the experimentation step. This is because the visualization in Figure 5.4 was performed after applying PCA.

Classifier	Precision	Recall	F1
Logistic regression	0.43	0.09	0.15
One-class SVM	0.40	0.19	0.25
SVM	0.75	0.03	0.06
SGD	0.21	0.28	0.24
Random forest	0.50	0.13	0.2

Table 5.7: Model 2 scores with comparison features and plagiarisms as positives

Table 5.7 and 5.8 provide positive plagiarism scores and positive original scores for multiple classifiers using the comparison features. Because of the imbalance of classes, table 5.8 is more interesting to analyze. The second model scores higher precision and lower recall than the first. This tells us that the documents which the second model classifies as positive are more likely to truly be plagiarisms, but it returns less correctly detected plagiarisms.

Classifier	Precision	Recall	F1
Logistic regression	0.78	0.96	0.86
One-class SVM	0.79	0.91	0.85
SVM	0.77	1.00	0.87
SGD	0.76	0.68	0.71
Random Forest	0.78	0.96	0.86

Table 5.8: Model 2 scores with comparison features and originals as positives

Classifier	TP	FP	TN	FN
One-class SVM	938	277	28	63
SVM	941	268	37	60
RF	964	279	26	37

Table 5.9: Model 2 confusion matrix with concatenated features

Table 5.9 shows the confusion matrix for multiple classifiers using the concatenation features. We can see that the only classifier which shows improvement with the concatenation features is the SVM.

Classifier	Precision	Recall	F1
One-class SVM	0.77	0.94	0.85
SVM	0.78	0.97	0.85
RF	0.78	0.97	0.86

Table 5.10: Model 2 scores using concatenated features and originals as positives

Classifier	Precision	Recall	F1
One-class SVM	0.31	0.09	0.14
SVM	0.38	0.12	0.18
RF	0.46	0.09	0.15

Table 5.11: Model 2 scores using concatenated features and plagiarisms as positives

Tables 5.10 and 5.11 show scores of multiple classifiers using the concatenated features. The overall performance of the concatenated features is lower than the comparison features.

While there are some positive results in both models, this system alone is not satisfactory enough for confident plagiarism detection on the sentence level. In the best case scenario, only half of the passages marked as plagiarisms truly are plagiarized. The system could, however, be used in conjunction with an external plagiarism detection system for highlighting suspicious passages.

6. Conclusion

The goal of intrinsic plagiarism detection is to uncover theft without the aid of external references by analyzing the discrepancies within a single corpus. This is a machine learning (ML) task, or more specifically a natural language processing (NLP) task.

This thesis presents two models for intrinsic plagiarism detection. The first performs outlier detection using the one-class SVM on features extracted from artificially generated plagiarized documents. The second model constructs a feature matrix from the entire dataset by comparing every window to its respective document and performs classification using a range of classifiers.

While there have been some positive results in both models, there is still a lot of room for improvement. The first model scored an F1 score of 0.30 on the plagiarism class and 0.82 on the original class. The best F1 score achieved in the second model was 0.25 on the plagiarism class and 0.85 on the original class. The second model marked fewer passages as plagiarisms, but with a greater accuracy.

Visualizing the extracted features with PCA clearly showed that the plagiarisms usually formed a separate group from the author's original passages. Since these groups aren't really outliers to the whole document, to detect plagiarisms the outlier detection model would also need to label many original windows as outliers. It would be worthwhile to explore the effect of clustering methods on the extracted features.

The comparison features were also successful at separating the two groups on a document level, but because the relative location of the groups is not consistent, the merged feature matrix does not provide a clear distinction between the groups. Making a more consistent separation would greatly benefit the second model. As is, the second model can effectively detect significant plagiarisms.

Adding more stylometric features focused on the character level similar to the character n-gram profiles explored in (Stamatatos, 2009) would likely improve both models.

It would also be interesting to see how the model performs on a different dataset. Scientific papers usually follow a set path of chapters, each often carrying a different writing style from the other. Sentiments shared in the introduction greatly differ from the table descriptions of the results. The system might perform better on a more homogenous dataset of a less scientific nature.

BIBLIOGRAPHY

Željko Agić, Nikola Ljubešić, i Danijela Merkler. Lemmatization and morphosyntactic tagging of croatian and serbian. U *Proceedings of the 4th Biennial International Workshop on Balto-Slavic Natural Language Processing*, stranice 48–57, 2013.

CLEF. Pan, 2017. URL `http://pan.webis.de/`. [Online; accessed 1-Jun-2017].

Edward Loper i Steven Bird. Nltk: The natural language toolkit. U *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, ETMTNLP '02, stranice 63–70, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1118108.1118117. URL `http://dx.doi.org/10.3115/1118108.1118117`.

Dean Malmgren. Textract, 2017. URL `https://github.com/deanmalmgren/textract/blob/f3e2902f36c0f07b3ec4b81905bbcdefc04e5188/docs/index.rst`. [Online; accessed 28-May-2017].

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, i E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Efstathios Stamatatos. Intrinsic plagiarism detection using character n-gram profiles. *threshold*, 2(1,500), 2009.

Angus Stevenson. *Oxford dictionary of English*. Oxford University Press, USA, 2010.

Wikipedia. Precision and recall — Wikipedia, the free encyclopedia, 2017. URL `https://en.wikipedia.org/wiki/Precision_and_recall`. [Online; accessed 1-June-2017].

Svante Wold, Kim Esbensen, i Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

Mario Zechner, Markus Muhr, Roman Kern, i Michael Granitzer. External and intrinsic plagiarism detection using vector space models. U *Proc. SEPLN*, svezak 32, stranice 47–55, 2009.

Sven Meyer Zu Eissen i Benno Stein. Intrinsic plagiarism detection. U *European Conference on Information Retrieval*, stranice 565–569. Springer, 2006.

Intrinsic Plagiarism Detection in Student Theses

Abstract

The goal of intrinsic plagiarism detection is to uncover theft without the aid of external references by analyzing the discrepancies within a single corpus. In this thesis, two approaches are proposed. One focuses one outlier detection using the one-class SVM, and the other performs classification using several classifiers. An artificial dataset was created for the task and the documents are analyzed on a sentence level using a sliding window. While there have been some positive results, the system alone is not satisfactory enough for confident plagiarism detection on the sentence level. The first model scored an F1 score of 0.3 on the plagiarism class and the second model achieved an F1 score of 0.25.

Keywords: Natural language processing, machine learning, plagiarism detection, intrinsic plagiarism detection, Croatian language, SVM, one-class SVM.

Intrinzično otkrivanje plagijata u studentskim radovima

Sažetak

Cilj intrinzičnog otkrivanja plagijata je prepoznavanje krađe unutar dokumenta bez pomoći referentnih tekstova. U ovom radu su predložena dva pristupa. Prvi koristi one-class SVM u prepoznavanju stršećih vrijednosti, dok drugi radi klasifikaciju koristeći niz klasifikatora. Za potrebe zadatka stvoren je umjetni skup podataka gdje su dokumenti analizirani na razini rečenice koristeći tehniku klizećeg prozora. Iako su postignuti pozitivni rezultati, sustav nije dovoljno učinkovit za samostalnu detekciju plagijata. F1 rezultat prvog modela je 0.3, dok je F1 rezultat drugog modela 0.25 za klasu plagijata.

Ključne riječi: Obrada prirodnog jezika, strojno učenje, detekcija plagijata, intrinzična detekcija plagijata, hrvatski jezik, SVM, one-class SVM.