Author Obfuscation by Automatically Rewriting Texts

Martijn Oele
ANR: 139805

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN COMMUNICATION AND INFORMATION SCIENCES,
MASTER TRACK DATA SCIENCE: BUSINESS & GOVERNANCE,
AT THE SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
OF TILBURG UNIVERSITY

Thesis committee:

C.D. Emmery MSc
dr. M.M. van Zaanen

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science & Artificial Intelligence
Tilburg, The Netherlands
July 2018

Preface

First of all I would like to acknowledge my thesis supervisor Mr. C.D. Emmery MSc., of the Tilburg School of Humanities and Digital Sciences at Tilburg University. Mr. Emmery was always willing to help me whenever I had a question during the process of researching and writing this thesis. Throughout this research project Mr. Emmery had a steering function and he permitted this thesis to be my own work. The feedback that he provided before I submitted this thesis was very useful. Secondly, I would like to thank the Kaggle team for making data sets freely available and maintaining the community that offers tips and tricks.

Furthermore, I would like to express my thanks to dr. M.M. van Zaanen, of the Tilburg School of Humanities and Digital Sciences at Tilburg University, for his time, expertise, knowledge and feedback as second reader of this thesis. In particular his tips and vision during the proposal presentation were extremely useful for the research of this project.

Lastly, I want to speak out my deepest appreciation to my dearest parents and to my girlfriend for offering me infallible support and endless encouragement during the full length of this master and through the process of researching and writing this thesis. This achievement would not have been possible without these people. Thank you.

Martijn Oele BSc. Artificial Intelligence

The code belonging to this project are stored in a online repository: *https://github.com/martijnoele/MSc-thesis-author-obfuscation*

Author Obfuscation by Automatically Rewriting Texts

Martijn Oele

Author identification is a practice where text mining techniques are applied to text aiming to identify the author of a text. This practice potentially exposes private information. To counter this, in this thesis we investigate author obfuscation by rewriting the original texts in an automatic manner - here referred to as revision models.

To test the level of obfuscation, an adversary is trained on stylometric features and term frequencies that are extracted from the raw data. Revision models rewrite texts using information that is obtained by determining the importance of features, manipulating features with more predictive value first, and are evaluated compared to round-trip baseline models and on the level of semantic preservation. The experiments lead to a revision model that obfuscates the author more effectively compared to the baseline models, but also show that rewriting sentences that preserve semantics is tough.

1. Introduction

Text mining is a field in data science that has been studied for a long time, and which is used often to create practical applications for industry purposes. The goal in this field is often to derive information from text, which can be information about the writing style (extracting so-called stylometric features) or information about the content. Examples of text mining research are cybercrime prevention in the forensics industry (De Vel et al. 2001), fraud detection by classifying written claims in the financial industry (Afroz, Brennan, and Greenstadt 2012) and customer care improvement in all kinds of industries (Lochter et al. 2016). One specific category of information extraction from written text is the identification of the author of a text, which can be divided in two types: 1) inferring author attributes (e.g. gender, age and political interest) (Volkova, Coppersmith, and Van Durme 2014) and 2) identifying the author itself. This study focuses on the latter.

In a classification task where the goal is to identify the author of a new unseen text, an algorithm has to be fitted on texts of a fixed set of authors, because a model can never identify an author from which it has never seen other texts; it can only infer attributes, such as gender or language variety (Rangel et al. 2017; Martinc et al. 2017), rather than authors' identities, from unseen authors. An author identification model can be useful in various practical applications, such as verifying if a person that claims having written a text (for instance a journalist) is indeed the author of the text, or verifying if the sender of cyberbullying messages is the person that belongs to the identity of the sender. Although author identification can be quite useful, it also invokes a major concern about privacy. By using existing identification techniques, it is possible to identify the author of an anonymous text (only if the author belongs to a known group). Examples of such scenarios are employees writing anonymous reviews about their supervisors or anonymous internal job applications. Åslund and Skans (2012) showed in their research

that anonymous application procedures increase the chances of minorities (women or non-Western people) to advance to the interview stage. These examples demand for a system that preserves the anonymity of the writer and thus show the practical relevance of this thesis.

To create these systems, two elements are needed. Firstly, a classifier is necessary to evaluate the performance of the revision models that are built in the second phase. A classifier is trained on stylometric features describing the writing style (e.g. fraction of unique words or the fraction of punctuation) and term frequencies. The classifier searches for patterns in the extracted features and based on these patterns the probability that a text belongs to a particular author is calculated. The stylometric features used to train classifiers are proposed in previous work in the field of author identification. After a classifier is trained, it is evaluated with different parameter settings and tested on the original data, after which the model with the best performance is applied to a rewritten data set.

The second element is a model that makes changes to the original text in order to obfuscate the identity of the author - these models will be called revision models in the remainder of this thesis. Revision models can either make small changes to the text (e.g. only concerning stylometric features) or they can paraphrase entire texts. The revision models manipulate (some of the) stylometric features that are used in the first part, but the level of manipulation does not depend on the classification results. In the end, the revision models are to be compared using so-called baseline revision models in order to make conclusions about the effectiveness. A general practice to assess machine translation quality is to use a round-trip translation as baseline (Almishari, Oguz, and Tsudik 2014). Revision models work well when their level of obfuscation is higher than the level of obfuscation of baseline models, but evaluation of the semantic similarity is needed to investigate if the revision models are useful in practical applications.

Many researchers have studied obfuscation of authors' identities. However, the revision models that are proposed in this thesis minimize the amount of edits, but guarantee the semantic similarity of the rewritten texts to some extent. This brings up the scientific relevance of this thesis. To evaluate the quality and effectiveness of the revision models, the models are evaluated on the drop in performance they cause, as well as on the semantic similarity with the original texts.

Since this research consists of multiple parts, the research question also describes multiple parts. Firstly, the features making classification possible need to be known and secondly, revision models are evaluated on the semantic similarity between rewritten and original texts. Therefore, the research question is formulated as follows:

Which features need to be manipulated, without changing the semantics, in order to make author identification impossible?

An answer to this research question will be found by firstly training an adversary that classifies which author has written a text. The performance of this adversary should be as high as possible. Secondly, the predictive value of a predefined set of stylometric features is computed for two different purposes: 1) try to increase the performance of the classifier by leaving out the features with very little predictive value, and 2) investigate which features should be manipulated, because they have much predictive value, to have much effect on obfuscation. Thirdly, multiple revision models are built by manipulating different features, which then will be evaluated on the level of obfuscation and the level of semantic preservation.

2. Related Work

Previous studies that are relevant for this study can be divided into three parts: author identification, author obfuscation and evaluation of obfuscation models. This section gives an overview of relevant research in these fields.

2.1 Author Identification

Researchers have used many different experimental setups to identify the author of written text. More than thirty years ago, Thisted and Efron (1987) used Bayes' theorem to determine if a new unseen poem was written by Shakespeare, which led to the conclusion that the poem was indeed written by Shakespeare. These researchers did not know in advance if Bayes' theorem would hold on the data they used, but the outcome of their experiment fits the more recent assertion that Naive Bayes works well even if Bayes' theorem does not hold (Friedman, Geiger, and Goldszmidt 1997; Domingos and Pazzani 1997).

Because author identification has practical applications, it is a popular topic among researchers (Iqbal et al. 2008; De Vel et al. 2001; Afroz, Brennan, and Greenstadt 2012; Fissette 2010). PAN, a network of digital text forensics, organizes yearly scientific tasks in the field of author identification. One of the goals of the 2017 task[1] is to identify authorship links (author clustering). The overview of the 2017 task describes each task with the corresponding data sets and evaluation methods (Potthast et al. 2017). The type of data that are used is similar as the type of data used in this thesis: all documents belong to a single author, differ in length and topic, but belong to the same genre and are written in the same language. The results of clustering authors were evaluated using B^3F_1 score, B^3 precision and B^3F_1 recall. From all submissions, the highest scores (for all evaluation metrics) were obtained by the model of Gómez-Adorno, Aleman, Vilariño, Sanchez-Perez, Pinto, and Sidorov (2017).

2.2 Author Obfuscation

Many previous studies are similar in the task framing (i.e. identifying authors), but they differ a lot in approach. Besides, most studies only focus on author identification, rather than identification and obfuscation of authors. More related is the work by Caliskan and Greenstadt (2012), who proposed the Translation Feature Set and who used these features (e.g. average characters per word, fraction of punctuation and fraction of special characters) to identify the author. Subsequently, they applied translation models to the original texts to obfuscate the authors. They created a classifier that was able to predict which author wrote a text with an accuracy of 91.54%. The global idea is the same as the idea of this thesis. There is a substantial difference, however, the classifier they used was also based on stylometric features as well as on term frequencies, but they did not use their own revision models but machine translation performed by different translators (Google Translate and Bing Translator) instead. Their main research task was therefore to create a model that could distinguish between different translators, whereas the research task in this thesis is to make author identification less possible by applying revision models to the original text.

1 https://pan.webis.de/clef17/pan17-web/author-identification.html

Brennan and Greenstadt (2009) used a different approach to obfuscate the author of a text. They used a classifier that predicted the author of a text and asked 12 laymen to rewrite a number of texts. The results of this study showed that the classifier did not perform better than chance level when it was applied to the rewritten texts. Later, this study was extended with a bigger group of human translators (Brennan, Afroz, and Greenstadt 2012). The performance of the classifier was 95% on the original texts and it dropped to 55% on rewritten texts. In this study, they used a common authorship identification model, using word unigrams, word bigrams, character bigrams and character trigrams.

The work by Shetty, Schiele, and Fritz (2017) uses a system that learns to perform obfuscation completely from the data. This method corresponds to the method used in this thesis. There are no human translators involved and the obfuscation method is completely automatic. In this study, the researchers use a more complex model, compared to the model that is used in this thesis, that firstly identifies the writing style of the text and secondly tries to obfuscate this style: the A^4NT network. In this thesis, a revision model is also completely automatic, but it does not learn and improve from the rewriting process and it minimizes the amount of edits that are needed to obfuscate the author.

2.3 Obfuscation Evalation

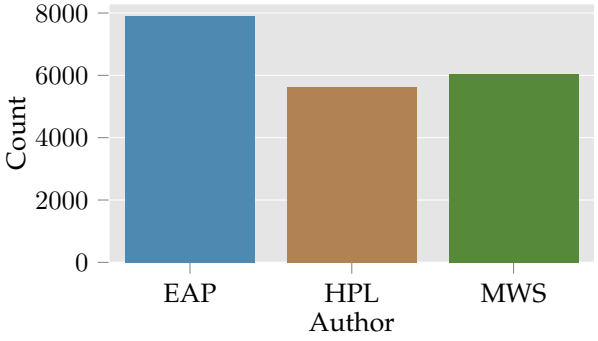
Mansoorizadeh, Rahgooy, Aminiyan, and Eskandari (2016) introduced three metrics to evaluate semantic similarity: safety (the original author should not be revealed from the rewritten text), soundness (the rewritten text should be textually entailed with the original text) and sensibility (the rewritten text should be inconspicuous). Based on this work, Shetty, Schiele, and Fritz (2017) introduced another metric to compute the semantic loss and used this metric to compare semantic similarity. The results of their experiments showed some minor offenders, which means that their model was not always accurate in rewriting the text (e.g. replacing wife by crush and replacing dad by husband). Emmery, Manjavacas, and Chrupała (2018) propose a trade-off between obfuscation, using a model to rewrite style invariant to improve semantic preservation, which they evaluated using humans.

The youngest branch of the PAN tasks is author obfuscation. The evaluation of submissions in this branch focuses on semantic similarity, for which they also use the metrics safe, sound and sensible (Potthast et al. 2017; Hagen, Potthast, and Stein 2017). The submitted result indicate that obfuscation models often produce texts with significantly changed semantics, and thus are not sensible and sound. However, one of the submitted models produces texts that fools the author identification classifier in 42% of the cases. In the overview they conclude that there are currently two types of obfuscation models: 1) models that produce text that are quite safe (i.e. obfuscate the author) but not sound and sensible (i.e. produce text with significantly changed meanings), and 2) models that are quite sound and sensible but not safe.

3. Experimental Setup

This section describes the data and the methods that are used for finding answers to the research question.

Figure 1
Number of instances per author in the original data set.



3.1 Data

3.1.1 Raw Data. The data that are used in this study belong to the Kaggle Spooky classification task[2]. The data consist of 19,579 rows, which all have 3 columns: `id`, which stores a unique id for each row, `text`, which contains sentences from different horror stories, and `author`, which stores a three-letter abbreviation of the author that wrote the horror story. The authors are Edgar Allan Poe (EAP), Mary Shelley (MWS) and HP Lovecraft (HPL). Figure 1 shows the numbers of sentences for each author. The data do not have missing values and the texts are of different lengths, of which the shortest contains 2 words and the longest contains 861 words. Table 1 shows some examples of the raw data set (see Appendix A for more examples).

2 https://www.kaggle.com/c/spooky-author-identification

Table 1
Raw data samples.

ID	Text	Author
id22530	Ellison was remarkable in the continuous profusion of good gifts lavished upon him by fortune.	EAP
id10154	Ahead lay sparse grass and scrub blueberry bushes, and beyond them the naked rock of the crag and the thin peak of the dreaded grey cottage.	HPL
id09646	Then came the frenzied tones again: "Carter, it's terrible monstrous unbelievable" This time my voice did not fail me, and I poured into the transmitter a flood of excited questions.	HPL
id07639	"Adrian, I am about to return to Greece, to become again a soldier, perhaps a conqueror."	MWS
id27335	Todder day he gib me slip fore de sun up and was gone de whole ob de blessed day.	EAP

The phase of preprocessing written text can be complex, because written text does not have a fixed and structured form (parts of a sentence can be presented in different orders) and it often contains misspelled words or abbreviations. Besides, texts can capture feelings or other meanings (such as sarcasm) that can not be explicitly expressed by (a combination of) words. Because we are in particular interested in style variation, preprocessing is applied to a minimal extent in this study. For example, the observation that an author consistently uses the word *tryin'* instead of *trying* adds relevant information to the classification model, and because the entire classification process is about variation in writing style, one should not withhold this from the models.

3.1.2 Features. Firstly, 8 stylometric features are extracted, which are characteristics of the literary style. These features are less relevant when the content of the text is to be analyzed.

- `words`: number of words in a text
- `chars`: number of characters (including punctuation) in a text
- `punctuations`: fraction of punctuations on total number of characters
- `unique_words`: fraction of unique words on total number of words
- `stopwords`: fraction of stopwords on total number of words
- `nouns`: fraction of nouns on total number of words
- `verbs`: fraction of verbs on total number of words
- `adjectives`: fraction of adjectives on total number of words

The stylometric features are used because in the rewriting phase they can be manipulated easily. For instance, long words can be replaced by multiple short words or vice versa. Manual exploration of the features will reveal which manipulation is needed to obfuscate the author. Appendix B shows the distribution plots of all stylometric features for each author.

Secondly, the importance of features is determined using a random forest algorithm (Geurts, Ernst, and Wehenkel 2006), which combines multiple decision trees to get more accurate results (Appendix C). We use a separate model, because using information from the classification model might result in overfitting. The results are combined with the plots of all features to obtain information about the distribution of the features, which can be used to improve the classifier by fitting it on subsets of stylometric features, or to determine which features can be manipulated in the rewriting phase.

Finally, to analyze which specific words are more often used by one particular author requires features that store information about the content of the text. A common practice in the field of text mining is vectorization of the text using word frequencies or tf*idf weighting there of - as e.g. implemented by the CountVectorizer[3] or TF-IDF Vectorizer[4] (Ullman 2011) modules in `scikit-learn` (Pedregosa et al. 2011). These modules create a vector with the corresponding values for all known words, based on words in a single text. Instances with similar vectors are often more similar in their

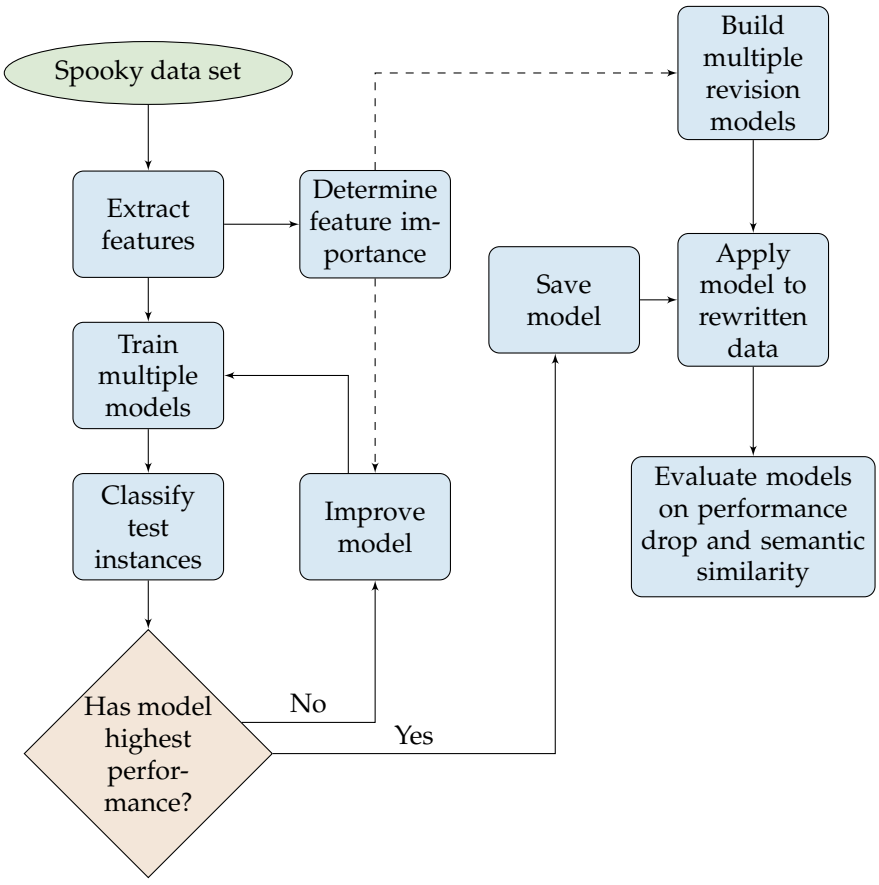
3 http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html
4 http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

contents. The vectors that are produced become very large quickly (more than thousands of values), and therefore some form of dimensionality reduction is desired. Latent Semantic Analysis (LSA) (Deerwester et al. 1990) is a form of dimensionality reduction that can be applied written text, because it does not require the data to be centered and it can work with term frequencies. In this thesis, LSA is applied with the following settings: `algorithm = "randomized"`, `n_iter = 5` and `n_components` $\in$ `{No LSA, 300, 1000, 3000, 6000}`.

3.2 Method / Models

The experimental setup of this thesis consists of two parts. Firstly, an adversary needs to be trained, evaluated and tested. The best adversary is used for the second part. Secondly, the texts need to be rewritten by multiple revision models, which accordingly need to be evaluated on the level of author obfuscation as well as on the level of semantic preservation. The two parts are developed in a way that they do not depend on each other. The flowchart in Figure 2 shows how the two parts are related.

Figure 2
Flowchart of the author identification and author obfuscation processes.



3.2.1 Classification Model. The data are split up in a train set (75%) and a test set (25%). The test set is used for evaluation of the adversary, for building revision models and for comparing semantic similarity. The train data are fitted using three different algorithms. The first model that is trained is a Multinomial Naive Bayes (NB), which works well with different types of data (integers, fractions) and which is highly scalable. Naive Bayes models are known to work quite well, even if the Naive Bayes assumption does not hold. The second model is a Decision Tree (DT), which is flexible and has a good interpretability, because the acquired information is returned in a readable form, which helps understanding the importance of the features. The final model is a K-Nearest Neighbors (KNN). K-Nearest Neighbor models are slow when there are too many data points. However, the idea of KNN (assign the label of the k nearest instances to a new instance) suits the idea of this thesis (assign the author of the most similar texts to a new text).

Using 5-fold cross validation and grid search each algorithm is fitted with different parameter settings (Appendix D). The models are evaluated using the F1-score, which combines recall (i.e. the part of all relevant instances that are classified as relevant) and precision (i.e. the part of all instances that are classified as relevant that are indeed relevant). In a multi-class classification problem, the macro F_1 score, which reaches its best value at 1 and its worst value at 0, is a weighted average of three F_1 scores for each author (a 'relevant' instance is a text that belongs to one particular author).

In the next step the optimal feature set is determined using the best estimator from the previous step. It might be that some features add noise instead of useful information, and therefore we have investigated if leaving out features improves the performance of the classifier. Rather than leaving out features at random, the predictive value of features are used to form a top-8, top-7 and top-6 of stylometric features. Theoretically, leaving out the least important stylometric features will not improve the classifier, because just one out of thousands of features is removed, but the results will confirm this hypothesis. In addition, both vectorizers that are described above are used with different parameter settings: `stop_words` $\in$ `{None, "english"}` and `min_df` $\in$ `{1,2,3}`. To determine the best classification model, the Naive Bayes is fitted on each combination of top-n stylometric features and a vectorizer.

3.2.2 Revision Model. Firstly, the original test set is used to create two baseline test sets, which are both Google Translate round trips. The first baseline is a round trip English $\rightarrow$ Dutch $\rightarrow$ English revision model. The second baseline is a round trip English $\rightarrow$ Finnish $\rightarrow$ English. The second baseline is expected to cause a bigger decrease in performance since Finnish is relatively more distant from Germanic languages.

Secondly, the revision models are built using the information that is obtained by exploring the features in Section 3.1.2 (note that this information does not depend on the classification models). To manipulate simple stylometric features such as the fraction of punctuation, the revision models use functions that are especially built for this study. The feature with the higest predictive value is the fraction of punctuations. This feature can easily be manipulated in a number of ways. The punctuation of a sentence can be completely removed (model 1: no_punctuation), the punctuation can be reduced (e.g. remove only commas - model 2: less_punctuation) or punctuation can be added (e.g. use '...' instead of '.' - model 3: more_punctuation). The chars feature can be manipulated by replacing words with synonyms that have more or less characters than the original word. To manipulate more complex features such as the fraction of unique words or replacing words that are most often used by one specific author, the revision models use synonyms.

Table 2
Different rewritten data sets that are used to obfuscate the identity of the author of a text.

Name of revision model			
1	no_punctuation	8	nouns
2	less_punctuation	9	nouns_no_punctuation
3	more_punctuation	10	nouns_less_punctuation
4	verbs	11	nouns_more_punctuation
5	verbs_no_punctuation	12	nouns_verbs
6	verbs_less_punctuation	13	nouns_verbs_no_punctuation
7	verbs_more_punctuation	14	nouns_verbs_less_punctuation
		15	nouns_verbs_more_punctuation

The `spacy` package[5] (Honnibal and Montani 2017) tokenizes all texts and assigns a POS (part-of-speech) tag to each token. For in-text replacements, the `KeyedVectors` module from the `gensim` package[6], which uses a corpus to find similar words (`GoogleNews-vectors-negative300`[7]), is used. One of the models, replaces all tokens with a `VERB` POS tag with the most similar synonym (model 4: verbs) and another model, replaces all tokens with a POS tag $\in$ {`NOUN`, `NNP`, `NN`} with the most similar synonym (model 8: nouns). A third model replaces both nouns and verbs (model 12: nouns_verbs). The choice to replace nouns and verbs, but not to replace adjectives, is based on the distribution of the features (Appendix B): the fractions of verbs and nouns is much higher than the fraction of adjectives, which means that replacing verbs or nouns has more impact on the term frequencies. In addition, model 4 and model 8 are combined with different manipulations of the `punctuation` feature (models 1-3). The complete set of revision models that are used for author obfuscation is shown in Table 2.

Thirdly, all rewritten data sets are classified with the stored adversary from Section 3.2.1. The performance of classification is measured with F_1 score and is compared to the F_1 score of the original test set. The difference in performance is called the *performance drop*. The goal is to build a revision model that has a higher performance drop than the baseline revision models. Besides, all rewritten data sets are evaluated on semantic similarity. Semantic similarity is measured with cosine similarity using the cortical.io API[8], which is inspired by the way the human brain processes textual information. The API creates semantic fingerprints of a text using over 16,000 features and uses these fingerprints to measure similarity between two texts (Ibriyamova et al. 2017). This study states that semantic fingerprinting works well, without human intervention, but to confirm these findings a small post-hoc task is set up to test the API. As additional metrics, the average number of manipulations and the IBM's BLEU score (Papineni et al. 2002) are computed and used to evaluate semantic similarity.

5 https://spacy.io
6 https://radimrehurek.com/gensim/models/word2vec.html
7 https://code.google.com/archive/p/word2vec/
8 https://www.cortical.io

4. Results

The goal of this thesis is to build a revision model that obfuscates the identify of an author. Therefore, we first need an adversary that classifies new texts. After it is trained and tuned, this adversary needs to be fooled with automatically rewritten texts. The revision model that causes the biggest drop in performance of the adversary is the optimal revision model. Revision models are also evaluated on preserving semantics by computing the semantic similarity.

Table 3 shows the scores of training different models model with LSA and without LSA. The data that are used to train these models consist of all eight stylometric features and term frequencies that are computed with a default TF-IDF Vectorizer (stop_words = None and min_df = 1). The scores of all other models, as well as the algorithm parameters that are manipulated, are shown in Appendix E. The parameters column shows the parameter setting that led to the best model. All models are trained with different values for n_components of LSA as dimensionality reduction. All algorithms performed best when n_components was set to 300. However, the scores were higher when LSA is not applied at all in most experiments. The performance of models is measured with F_1 scores. The highest F_1 score is printed in a bold font.

The model with the highest F_1 score (i.e. Naive Bayes, alpha = 1.0, no LSA) is trained again using different subsets (top-8, top-7 and top-6) of the stylometric features (based on feature importance) and different settings for the vectorizers. Table 4 shows the results of fitting the tuned Naive Bayes model. Each cell in this table shows the F_1 score for a combination of different vectorizer settings and a different number of stylometric features. For instance: training a Naive Bayes (alpha = 1.0) on TF-IDF(stop_words=None, min_df=1) and 6 best stylometric features results in F_1 score = 0.841. Again, the models are compared on F_1 score. The model with the highest score is printed in a bold font. This model (Naive Bayes, alpha = 1.0, 8 stylometric features, TF-IDF vectorizer, stop_words = None, min_df = 2) is used to test author obfuscation in the second part of this thesis.

The best classifier from the previous steps is applied to all rewritten data sets (See Appendix H for example sentences as produced by revision models). Table 5 shows the F_1 score of classifying these test sets. The reference data set is the original test set. For all other data sets, the drop in performance is computed by subtracting the F_1 score for a rewritten set from the F_1 score of the original set (the obtained F_1 scores of all models

Table 3
Results of training different classification models. This table shows the highest F_1 score with LSA and without LSA for each model.

Model	Parameters	LSA (n terms)	F_1 score
Naive Bayes	alpha = 1.0	-	**0.843**
	alpha = 0.01	300	0.232
Decision Trees	max_depth = 10	-	0.545
	max_depth = 5	300	0.539
K-Nearest Neighbours	-	-	-
	n_neighbours = 5	300	0.476

Table 4
F_1 scores of optimized Naive Bayes model (alpha = 1.0), trained on different feature sets.

		Top n stylo features		
Vectorizer	Parameters	8	7	6
TF-IDF	stop_words=None, min_df=1	0.843	0.843	0.841
	stop_words=None, min_df=2	**0.845**	0.845	0.845
	stop_words=None, min_df=3	0.844	0.844	0.844
	stop_words="english", min_df=1	0.824	0.824	0.824
	stop_words="english", min_df=2	0.823	0.823	0.823
	stop_words="english", min_df=3	0.823	0.823	0.822
CountVectorizer	stop_words=None, min_df=1	0.835	0.835	0.833
	stop_words=None, min_df=2	0.838	0.838	0.837
	stop_words=None, min_df=3	0.839	0.839	0.839
	stop_words="english", min_df=1	0.825	0.825	0.825
	stop_words="english", min_df=2	0.828	0.828	0.829
	stop_words="english", min_df=3	0.827	0.828	0.827

Table 5
Performance drop and semantic similarity computed with cosine similarity (cortical.io), BLEU score and average number of changes.

Revision model	Perf. drop	Cosine similarity	BLEU	Avg. number of changes
no_punctuation	0.003	0.988 ± 0.048	**1.000**	3.797 ± 3.112
less_punctuation	0.000	1.000 ± 0.006	**1.000**	$\mathbf{1.956 \pm 2.074}$
more_punctuation	0.001	$\mathbf{1.000 \pm 0.000}$	**1.000**	2.093 ± 2.455
verbs	0.078	0.655 ± 0.153	0.107	12.658 ± 8.978
verbs_no_punctuation	0.078	0.655 ± 0.153	0.107	16.455 ± 10.994
verbs_less_punctuation	0.078	0.655 ± 0.153	0.107	14.614 ± 10.342
verbs_more_punctuation	0.078	0.655 ± 0.153	0.107	14.751 ± 9.289
nouns	0.117	0.561 ± 0.134	0.054	14.602 ± 10.572
nouns_no_punctuation	0.118	0.561 ± 0.133	0.054	18.399 ± 12.539
nouns_less_punctuation	0.118	0.561 ± 0.133	0.054	16.558 ± 11.941
nouns_more_punctuation	0.117	0.561 ± 0.133	0.054	16.695 ± 10.841
nouns_verbs_replaced	**0.154**	0.460 ± 0.132	0.103	12.934 ± 10.045
nouns_verbs_no_punctuation	0.137	0.462 ± 0.136	0.103	16.730 ± 11.990
nouns_verbs_less_punctuation	0.153	0.460 ± 0.132	0.103	14.890 ± 11.360
nouns_verbs_more_punctuation	0.154	0.460 ± 0.132	0.103	15.027 ± 10.329
Baseline: Google Translate (NL)	0.081	0.690 ± 0.158	0.418	6.025 ± 12.463
Baseline: Google Translate (FI)	0.114	0.592 ± 0.163	0.208	8.252 ± 5.931
Reference: original	-	1.000 ± 0.000	1.000	0.000 ± 0.000

are shown in Appendix F). There are two baselines: English → Dutch Google Translate round-trip and English → Finnish Google Translate round-trip, which also cause a drop in performance. The highest performance drop is printed in a bold font. The next three columns show the semantic similarity between the rewritten data set and the original test set, measured with the cosine similarity (avg ± std), BLEU score and number of manipulations (avg ± std). The highest average cosine similarity is printed in a bold font. Appendix G shows all results of the computations of the cosine similarities.

As a post-hoc analysis, the cortical.io API demands for some manual analysis. Therefore, the API is tested with word combinations as well as sentence combinations. Appendix I shows the texts that are used as input of the API. The middle column shows the semantic similarity (based on the cosine similarity measure) that was returned by the API. The results of the API and whether or not the API shows divergent behaviour is discussed in the next section.

5. Discussion

The results that are shown above are discussed and interpreted with regards to the research question. Results that are not necessarily relevant to answer the research question but which are remarkable however, are also discussed here. The goal of this thesis was to build a revision model, in order to find out which features need to be manipulated to obfuscate the identity of the original author, while maintaining good semantic similarity.

In the first part, an adversary is trained using three classification algorithms. In the results section, the highest outcomes are presented (Table 3). Appendix E shows all results of the training phase. The highest score is obtained by a trained Multinomial Naive Bayes (NB) model (F_1 = 0.843). It appears that adding LSA heavily drops the performance of the NB model. The number of terms that are captured with this type of dimensionality reduction has no impact on the score of the model. When LSA is added to the NB model, all instances are classified to one particular class. This observation indicates that there is important information captured by the term frequencies when these are not reduced. Apparently, this information is used by Naive Bayes to make a good model.

The second model that is trained is a Decision Tree (DT). The results show that a DT model performs worse than NB models (F_1 = 0.545 vs 0.843). The difference between a DT model without LSA and a DT model with LSA (F_1 = 0.545 and 0.539, respectively) is much smaller compared to the difference of NB model scores without and with LSA (F_1 = 0.843 and 0.232, respectively). However, when LSA is added to a DT model, the complexity of the model (i.e. the maximum depth of the tree) decreases with a factor 2, which is good for interpretability and the computation time of the decision tree.

The final model is a K-Nearest Neighbors (KNN) model. This algorithm could only fit a model with LSA for n_components equals 300 and 1000. Without LSA, or with a higher n_components, the dimensions of the data were too large for the model to fit (i.e. the computation took more than several hours). A KNN model tries to find instances that are most similar to a new instance (note that this is not clustering). Without dimensionality reduction (e.g. LSA), a KNN model has to match instances on more than 15,000 features, which is a very time and memory intensive task.

The model with the highest score from the first step is trained again on 36 different feature sets (Table 4). The differences between the F_1 scores of all those models are very small. The lowest score is 0.822 and the highest score is 0.845. It appears that removing stylometric features with the least predictive value does not improve the performance of

a model, which is possibly caused by the fact that just one out of thousands of features is removed. Changing the settings of the vectorizers also makes little difference. For all models, it can be concluded that vectorizers that do not take into account stopwords perform better. A possible cause for this observation is that stopwords are spelled differently in the raw data, and that they are therefore not counted as stopwords (see Appendix A for examples of texts with misspelled words). This brings up a limitation as well as a possible improvement of this research. If the original texts were preprocessed in a professional manner (as performed by linguists), the vectors with term frequencies will become smaller when, for example, lemmatization is applied, which leads to more predictive value of the stylometric features. Besides, dimensionality reduction by LSA might then also improve the performance of the models. The downside, however, is that style variance may be removed when any form of preprocessing is applied.

There are three models with F_1 score = 0.845, which brings up the discussion about which model should be used. The stylometric features are chosen based on previous research in the linguistic field and therefore there are good reasons to keep them in a model. The computation speed of the models does not vary much, giving no reason to choose another model.

As described in the methods section, the first revision model is built by manipulating relatively simple stylometric features. The punctuation feature has the most predictive value and it is easy to manipulate (e.g. adding extra commas or removing all punctuation). However, the performance of classifying a data set with the punctuation manipulated barely drops. All three models with manipulated punctuation cause performance drops smaller than 0.003 (Table 5). The main reason for these small performance drops is that just one feature is adjusted, because the vectorizers do not count punctuation: the fraction of punctuation can decrease from 0.5 to 0.0, but the impact on the entire vector with feature values is too small. These observations lead to the conclusion that manipulation of a stylometric feature with the most predictive value has little impact on obfuscation. However, manipulation of this feature does barely change the semantics, as measured with all methods described in this work. For all three models with the punctuation manipulated, the cosine similarity measured with corticol.io is around 1.000, the BLEU scores, representing a modified precision score, are much higher compared to all other models and the average number of changes is much smaller. A BLEU score close to 1.0 indicates that the rewritten text is quite similar to the original text. There are no previous studies where the researchers were able to rewrite sentences with a higher cosine similarity, or anything barely close.

In all other revision models, the term frequency vectors are affected by replacing special types of words with synonyms. These models cause bigger performance drops, because a bigger part of the features are affected. Not only term frequencies change, but also some stylometric features (such as unique_words) can change. The results show that when one type of words (e.g. all nouns or all verbs) are replaced, manipulating another stylometric feature does not improve the revision model (i.e. increase the performance drop). This observation also holds for the semantic similarity, as measured with cosine similarity and BLEU score. These scores do not change when an additional stylometric feature is manipulated. The average number of changes changes, but this is a logical result of the fact that the number of changes depends on whether punctuation is added, punctuation is sometimes removed, or punctuation is removed entirely. The biggest drop in performance is obtained when both verbs and nouns are replaced (performance drop = 0.154, average cosine similarity = 0.460). This drop is higher than the drop that is caused by both baseline models (Dutch round-trip: 0.081; Finnish round-trip: 0.114). The hypothesis that the drop caused by the Finnish round-trip is bigger than

the drop caused by the Dutch round-trip is confirmed. The latter also have a higher semantic similarity compared to a Finnish round-trip.

A better revision model can be obtained by adding knowledge about term frequencies per author. The revision models that are built replace all words of a certain type (e.g. POS tag = VERB), without knowing which verbs in a sentence are relevant for making classifications. Adding this knowledge results in models with better semantic similarity scores. The cosine similarity will be higher because original and rewritten texts are more similar; texts with less changed words have a higher precision with regards to the original text (i.e. higher BLEU score); the amount of changes made is smaller so the average number of changes will also be lower. The downside, however, is that these revision models need to be trained on the original texts, because without training the revision models do not know which words/terms in a text are relevant for the classification, whereas the revision models built in this thesis can be applied out-of-the-box. Possible future work is to build models that improve themselves while rewriting texts, so that they gain knowledge about relevant features. This approach is more author-specific and will need more computation time. When an author-specific model knows which terms are relevant, the number of changes can be reduced, so this future improvement fits the minimalist approach.

In conclusion, it can be said that the packages that are used (`spacy` for POS-tagging and `gensim` for replacing words with similar words) are good resources to do these tasks. The results also show that it is hard to preserve semantics, which is in line with the two types of author obfuscation described in the overview of Potthast, Rangel, Tschuggnall, Stamatatos, Rosso, and Stein (2017). The authors are more obfuscated compared to the baselines when either the verbs are replaced, or the nouns and verbs are replaced, which shows that replacing words with synonyms is a better method for author obfuscation compared to manipulating a single stylometric features. However, the latter is a better method when preserving semantics has the highest priority. The average cosine similarity measured with the cortical.io API is between 0.46 and 0.66 for all revision models that affect term frequencies, whereas the cosine similarity is around 1.00 if the punctuation fraction is manipulated. The BLEU score is much higher for the models that manipulate the punctuation feature compared to the models that manipulate term frequencies. Both baseline models do not have high cosine similarity as well (Dutch round-trip: 0.690; Finnish round-trip: 0.592), but the BLEU scores of the baseline models (Dutch round-trip: 0.418; Finnish round-trip: 0.208) are a bit higher than the BLEU scores of the revision models that affect term frequencies. The average number of changes in the Dutch round-trip is smaller compared to the Finnish round-trip, which again confirms the assumption that the Finnish language is quite more distant from English compared to the Dutch language.

Looking at the results, an answer to the research question can be formulated by saying that replacing nouns with synonyms and removing some punctuation leads to a model that has a higher performance drop and just a slightly lower cosine similarity with regards to both baseline models. However, the BLEU score is 2-4 times smaller than the baselines and the average number almost two times larger. The approach that is adopted in this work is quite new (i.e. minimize the amount of edits), however, the results are not outstanding. The obfuscation methods work well, but they change the sentences too heavily. As future improvement, the methods used here can be combined with methods from other work (e.g. the work by Emmery, Manjavacas, and Chrupała (2018)) to create models that have a higher level of semantic preservation. These new models should focus on semantic similarity primarily, rather than on the level of obfuscation.

A limitation is that there is few existing work that shows the accuracy of the corticol.io API and therefore the post-hoc analysis of the API is added to this thesis. Appendix I shows a table with sample texts that are evaluated on semantic similarity using the corticol.io API. The returned cosine similarities are quite divergent. The first example (*tree* vs *forest*) is expected to have a quite high similarity score, which is not true: cosine similarity = 0.30. However, the next example (*trees* vs *forest*) has a score that is around 1.5 times as high, which makes sense because a forest always consists of multiple trees. The results of the post-hoc analysis also show that misspelled words cause a drop in similarity with regards to the same text without misspelled words (0.19 and 0.50, respectively). The API seems to return quite low similarity scores if both inputs are not completely identical, which is probably caused by the fact that the fingerprints differ too much when just some words in a sentence have changed. In this thesis, it is in particular the semantic similarity of sentences with only a few words different that needs to be computed, so therefore this API might not be the right tool to use.

6. Conclusion

The goal of this thesis was to build a revision model that makes small changes to texts (e.g. replacing words with synonyms, manipulating punctuation or adjusting the length of texts) in such a way that a pre-trained classifier will not be able anymore to predict the author of the manipulated text. In the first step, a classifier that fits a model on training data that consist of 8 stylometric features (words, chars, punctuations, unique_words, stopwords, nouns, verbs, adjectives) and term frequency vectors (CountVectorizer and TF-IDF vectorizer, with and without LSA), was trained. Adding LSA makes the fitted models less complex in time and space, but the performance does not increase.

In the second phase, the classifier is tuned on different (sub)sets of features. The settings of the vectorizer are varied and stylometric features with least predictive value, which are computed using a Random Forest model, are left out. The latter had no effect on the performance of the classifier. Varying settings of the vectorizer had little effect on the performance. The optimal model was trained on all stylometric features and TF-IDF vectorizer with stopwords = None, min_df = 1. The observations have led to the conclusion that advanced preprocessing practices can improve the classification model, since texts that are preprocessed can be used to apply lemmatization and dimensionality reduction (e.g. LSA) has more impact, however, preprocessing might result in important style variance to be removed from the data. The classifier with the highest performance, as measured with F_1 score is used in the obfsuscation part.

In the obfuscation phase, several revision models were built. These models manipulate (a combination of) the punctuation feature and term frequencies, by replacing all nouns and/or verbs. The texts that were produced by the revision models are classified with the previous trained model. The results that are obtained show that manipulating a stylometric feature that has most predictive value (i.e. fraction of punctuations) has no impact on the level of obfuscation. Adjusting other parts of the texts (e.g. replacing nouns or verbs with synonyms) has more impact, however, the semantic similarity (measured with cosine similarity using the cortical.io API, BLEU score and the average number of changes) of texts that have undergone these manipulations is very low (cosine similarity = $\pm$ 0.5, BLEU score = $\pm$ 0.1, avg. number of changes = $\pm$ 15). The revision model with the biggest performance drop replaces all nouns and all verbs of the text with a synonym (performance drop = 0.154). This drop is compared to the drop caused by two baseline models: English $\rightarrow$ Dutch $\rightarrow$ English and English $\rightarrow$ Finnish $\rightarrow$ English Google Translate round-trips (0.081 and 0.114, respectively). With respect to

the performance drop, manipulating all nouns and/or verbs are a good method for author obfuscation, but it fails in preserving semantics. The revision models can be divided in two types: revision models with 1) high semantic similarity but a low level of obfuscation, and 2) a high level of obfuscation, but low semantic similarity. Suggested improvements are to use methods that train a revision model in order to add knowledge about the relevant style variance. This study can also be extended by experimenting if more generalizable models, that apply preprocessing to the raw data, remove important style variance.

References

Afroz, Sadia, Michael Brennan, and Rachel Greenstadt. 2012. Detecting hoaxes, frauds, and deception in writing style online. In *Security and Privacy (SP), 2012 IEEE Symposium on*, pages 461–475, IEEE.

Almishari, Mishari, Ekin Oguz, and Gene Tsudik. 2014. Fighting authorship linkability with crowdsourcing. In *Proceedings of the second ACM conference on Online social networks*, pages 69–82, ACM.

Åslund, Olof and Oskar Nordströum Skans. 2012. Do anonymous job application procedures level the playing field? *ILR Review*, 65(1):82–107.

Brennan, Michael, Sadia Afroz, and Rachel Greenstadt. 2012. Adversarial stylometry: Circumventing authorship recognition to preserve privacy and anonymity. *ACM Transactions on Information and System Security (TISSEC)*, 15(3):12.

Brennan, Michael Robert and Rachel Greenstadt. 2009. Practical attacks against authorship recognition techniques. In *IAAI*.

Caliskan, Aylin and Rachel Greenstadt. 2012. Translate once, translate twice, translate thrice and attribute: Identifying authors and machine translation tools in translated text. In *Semantic Computing (ICSC), 2012 IEEE Sixth International Conference on*, pages 121–125, IEEE.

De Vel, Olivier, Alison Anderson, Malcolm Corney, and George Mohay. 2001. Mining e-mail content for author identification forensics. *ACM Sigmod Record*, 30(4):55–64.

Deerwester, Scott, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.

Domingos, Pedro and Michael Pazzani. 1997. On the optimality of the simple bayesian classifier under zero-one loss. *Machine learning*, 29(2-3):103–130.

Emmery, Chris, Enrique Manjavacas, and Grzegorz Chrupała. 2018. Style obfuscation by invariance. *arXiv preprint arXiv:1805.07143*.

Fissette, MVM. 2010. Author identifcation in short texts.

Friedman, Nir, Dan Geiger, and Moises Goldszmidt. 1997. Bayesian network classifiers. *Machine learning*, 29(2-3):131–163.

Geurts, Pierre, Damien Ernst, and Louis Wehenkel. 2006. Extremely randomized trees. *Machine learning*, 63(1):3–42.

Gómez-Adorno, Helena, Yuridiana Aleman, Darnes Vilariño, Miguel A Sanchez-Perez, David Pinto, and Grigori Sidorov. 2017. Author clustering using hierarchical clustering analysis. CLEF.

Hagen, Matthias, Martin Potthast, and Benno Stein. 2017. Overview of the author obfuscation task at pan 2017: safety evaluation revisited. *Working Notes Papers of the CLEF*, pages 33–64.

Honnibal, Matthew and Ines Montani. 2017. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*.

Ibriyamova, Feriha, Samuel Kogan, Galla Salganik-Shoshan, and David Stolin. 2017. Using semantic fingerprinting in finance. *Applied Economics*, 49(28):2719–2735.

Iqbal, Farkhund, Rachid Hadjidj, Benjamin CM Fung, and Mourad Debbabi. 2008. A novel approach of mining write-prints for authorship attribution in e-mail forensics. *digital investigation*, 5:S42–S51.

Lochter, Johannes V, Rafael F Zanetti, Dominik Reller, and Tiago A Almeida. 2016. Short text opinion detection using ensemble of classifiers and semantic indexing. *Expert Systems with Applications*, 62:243–249.

Mansoorizadeh, Muharram, Taher Rahgooy, Mohammad Aminiyan, and Mahdy Eskandari. 2016. Author obfuscation using wordnet and language modelsâĂŤnotebook for pan at clef 2016. In *CLEF 2016 Evaluation Labs and Workshop–Working Notes Papers. CEUR-WS. org*.

Martinc, Matej, Iza Škrjanec, Katja Zupan, and Senja Pollak. 2017. Pan 2017: Author profiling-gender and language variety prediction. *Cappellato et al.[13]*.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318, Association for Computational Linguistics.

Pedregosa, F., 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, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Potthast, Martin, Francisco Rangel, Michael Tschuggnall, Efstathios Stamatatos, Paolo Rosso, and Benno Stein. 2017. Overview of panâĂŹ17. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 275–290, Springer.

Rangel, Francisco, Paolo Rosso, Martin Potthast, and Benno Stein. 2017. Overview of the 5th author profiling task at pan 2017: Gender and language variety identification in twitter. *Working Notes Papers of the CLEF*.

Shetty, Rakshith, Bernt Schiele, and Mario Fritz. 2017. A4nt: Author attribute anonymity by adversarial training of neural machine translation.

Thisted, Ronald and Bradley Efron. 1987. Did shakespeare write a newly-discovered poem? *Biometrika*, 74(3):445–455.

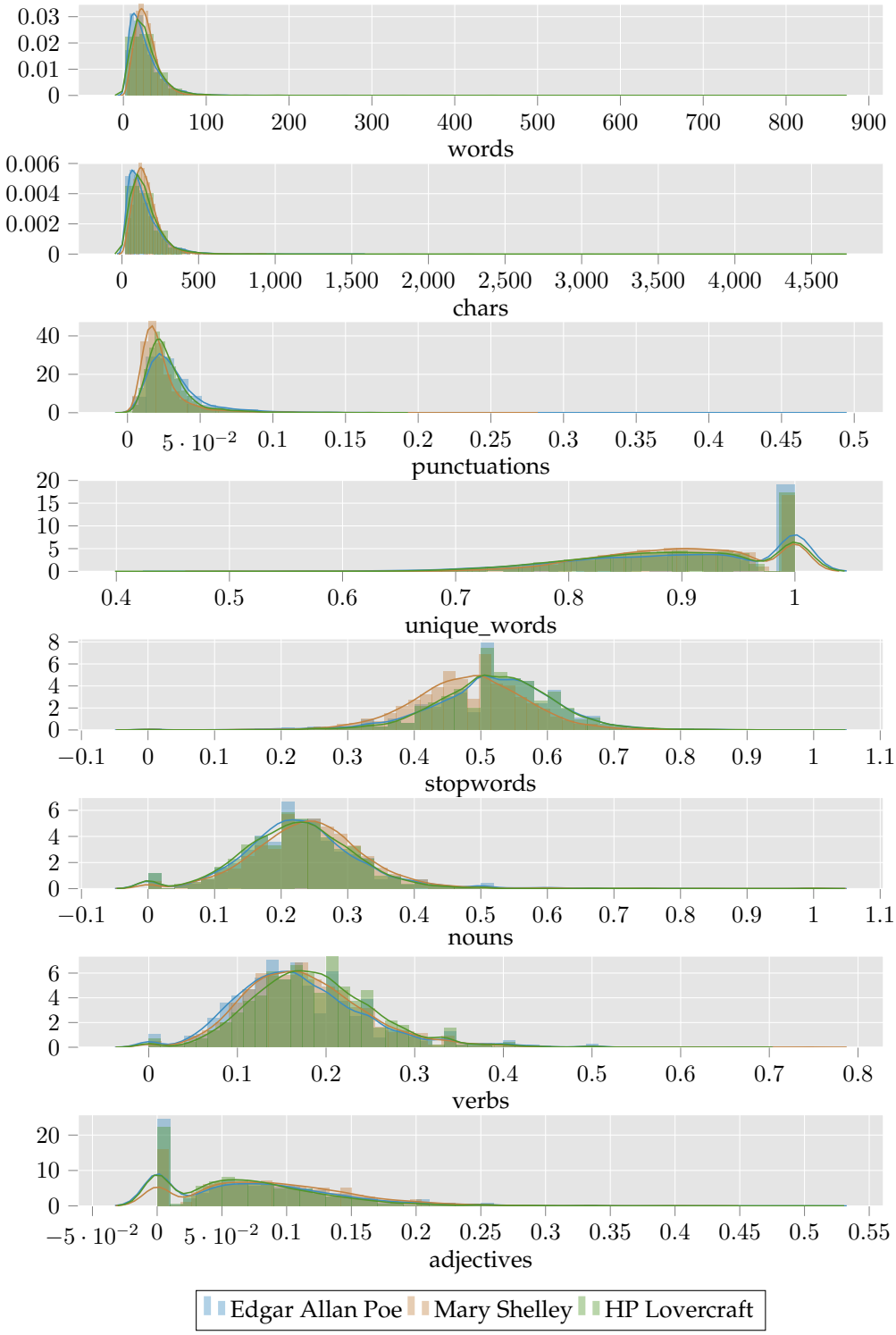
Ullman, Jeffrey David. 2011. *Mining of massive datasets*. Cambridge University Press.

Volkova, Svitlana, Glen Coppersmith, and Benjamin Van Durme. 2014. Inferring user political preferences from streaming communications. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 186–196.

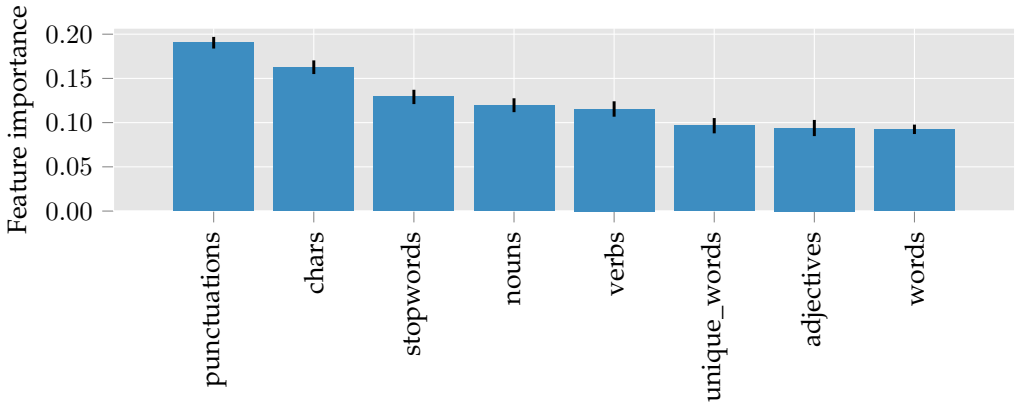
Appendix A: Raw data samples.

ID	Text	Author
id22530	Ellison was remarkable in the continuous profusion of good gifts lavished upon him by fortune.	EAP
id10154	Ahead lay sparse grass and scrub blueberry bushes, and beyond them the naked rock of the crag and the thin peak of the dreaded grey cottage.	HPL
id03072	"I vaow afur Gawd, I dun't know what he wants nor what he's a tryin' to dew."	HPL
id10542	I must collect my thoughts.	MWS
id27149	Trade fallin' off, mills losin' business even the new ones an' the best of our menfolks kilt a privateerin' in the War of or lost with the Elizy brig an' the Ranger snow both of 'em Gilman venters.	HPL
id09646	Then came the frenzied tones again: "Carter, it's terrible monstrous unbelievable" This time my voice did not fail me, and I poured into the transmitter a flood of excited questions.	HPL
id07639	"Adrian, I am about to return to Greece, to become again a soldier, perhaps a conqueror."	MWS
id16041	After great trouble, occasioned by the intractable ferocity of his captive during the home voyage, he at length succeeded in lodging it safely at his own residence in Paris, where, not to attract toward himself the unpleasant curiosity of his neighbors, he kept it carefully secluded, until such time as it should recover from a wound in the foot, received from a splinter on board ship.	EAP
id16815	'Oppodeldoc,' whoever he is, is entirely devoid of imagination and imagination, in our humble opinion, is not only the soul of sy, but also its very heart.	EAP
id26550	We now resolved to let off enough gas to bring our guide rope, with the buoys affixed, into the water.	EAP
id01304	Many changes also now occurred in these spontaneous regal elections: depositions and abdications were frequent, while, in the place of the old and prudent, the ardent youth would step forward, eager for action, regardless of danger.	MWS
id27335	Todder day he gib me slip fore de sun up and was gone de whole ob de blessed day.	EAP
id16909	The rudder was a light frame of cane covered with silk, shaped somewhat like a battle door, and was about three feet long, and at the widest, one foot.	EAP
id19165	There are horrors beyond horrors, and this was one of those nuclei of all dreamable hideousness which the cosmos saves to blast an accursed and unhappy few.	HPL

Appendix B: Distribution plots of all features, separated by author.



Appendix C: Feature importance computed with a Random Forest model.



Appendix D: Parameter settings that manipulated using a grid search 5-fold cross validation.

Model	Parameter	Values
Multinomial Naive Bayes	alpha	{0.01, 0.1, 1.0}
Decision Tree	max_depth	{3, 5, 10, 20, 25}
K-Neares Neighbors	n_neighbors	{5, 10}

Appendix E: Classification results of all different models.

Model	Parameters	LSA (n terms)	F_1 score
Naive Bayes	alpha = 1.0	-	**0.843**
	alpha = 0.01	300	0.232
	alpha = 0.01	1000	0.232
	alpha = 0.01	3000	0.232
	alpha = 0.01	6000	0.232
Decision Trees	max_depth = 10	-	0.545
	max_depth = 5	300	0.539
	max_depth = 5	1000	0.534
	max_depth = 5	3000	0.535
	max_depth = 5	6000	0.527
K-Nearest Neighbours	-	-	-
	n_neighbours = 5	300	0.476
	n_neighbours = 5	1000	0.346
	n_neighbours = 5	3000	0.253
	-	6000	-

Revision model	F_1 score	Performance drop
no_punctuation	0.842	0.003
less_punctuation	0.845	0.000
more_punctuation	0.844	0.001
verbs	0.767	0.078
verbs_no_punctuation	0.767	0.078
verbs_less_punctuation	0.767	0.078
verbs_more_punctuation	0.767	0.078
nouns	0.728	0.117
nouns_no_punctuation	0.727	0.118
nouns_less_punctuation	0.727	0.118
nouns_more_punctuation	0.728	0.117
nouns_verbs_replaced	0.691	**0.154**
nouns_verbs_no_punctuation	0.708	0.137
nouns_verbs_less_punctuation	0.692	0.153
nouns_verbs_more_punctuation	0.691	0.154
Baseline: Google Translate (NL)	0.764	0.081
Baseline: Google Translate (FI)	0.731	0.114
Reference: original	0.845	-

Appendix F: Computed F_1 scores and performance drops of all rewritten data sets.

Appendix G: All semantic similarity results: average (avg), standard deviation (std), minimum (min) and maximum (max)

	Cosine similarity			
Revision model	avg	std	min	max
no_punctuation	0.988	0.048	0.045	1.000
less_punctuation	1.000	0.006	0.786	1.000
more_punctuation	1.000	0.000	1.000	1.000
verbs	0.655	0.153	0.028	1.000
verbs_no_punctuation	0.655	0.153	0.028	1.000
verbs_less_punctuation	0.655	0.153	0.028	1.000
verbs_more_punctuation	0.655	0.153	0.028	1.000
nouns	0.561	0.134	0.028	1.000
nouns_no_punctuation	0.561	0.133	0.028	1.000
nouns_less_punctuation	0.561	0.133	0.028	1.000
nouns_more_punctuation	0.561	0.134	0.028	1.000
nouns_verbs	0.460	0.132	0.026	1.000
nouns_verbs_no_punctuation	0.462	0.136	0.026	1.000
nouns_verbs_less_punctuation	0.460	0.132	0.026	1.000
nouns_verbs_more_punctuation	0.460	0.132	0.026	1.000
Baseline: Google Translate (NL)	0.690	0.158	0.000	1.000
Baseline: Google Translate (FI)	0.592	0.163	0.000	1.000
Reference: original	1.000	0.000	1.000	1.000

Appendix H: Example texts as produced by revision models.

Models: 1 - original, 2 - baseline (FI), 3 - baseline (NL), 4 - no_punctuation, 5 - verbs, 6 - nouns, 7 - nouns_verbs

Model	Text	Sim
1	You say to me that there is nothing in the swamp or near it which could form the setting of that frightful episode.	-
2	You tell me that there is nothing in the swamp or near that could form that scary episode.	0.671
3	You tell me that there is nothing in the swamp or near it that might be the environment of that horrible episode.	0.595
4	You say to me that there is nothing in the swamp or near it which could form the setting of that frightful episode	1.000
5	You believe to somebody it there'sa isn'ta anything inthe this swamp either near that The would shape this setting of it frightful episode.	0.489
6	You believe to somebody it there'sa isn'ta anything inthe this swampy either near that The would form this set of it frightful ep.	0.352
7	You say to somebody that it'sa isn'ta nothing in the swampy or near it The could shape the set of that frightful ep.	0.375

Appendix I: Post-hoc analysis of cortical.io API.

* *Rewritten by nouns_verbs_replaced*
** *Rewritten by nouns_verbs_replaced without spelling mistakes*
*** *Rewritten by nouns_verbs_replaced_no_punctuation*

Text 1	Sim.	Text 2
Tree	1.000	Tree
Tree	0.305	Forest
Trees	0.461	Forest
Car	0.366	Vehicle
Two cars have collided.	0.422	A car was driving against another car.
Let us pursue our fancies.	0.190	Can everybody persue yours loves.*
Let us pursue our fancies.	0.497	Can everybody pursue yours loves.**
Let us pursue our fancies.	0.190	Can everybody persue yours loves***
There were also some curious muddy rat tracks leading out of a fresh hole and back into it again.	0.515	There were additionally several curious muddy rodent tracks preeminent up of a fresh fairway and down intothe it again.*