To the Most Gracious Highness, from Your Humble Servant: Analysing Swedish 18th Century Petitions Using Text Classification

Ellinor Lindqvist[1] **Eva Pettersson**[1] **Joakim Nivre**[1,2]

[1]Uppsala University
Dept. of Linguistics and Philology
firstname.lastname@lingfil.uu.se

[2]RISE Research Institutes of Sweden
Dept. of Computer Science
joakim.nivre@ri.se

Abstract

Petitions are a rich historical source, yet they have been relatively little used in historical research. In this paper, we aim to analyse Swedish texts from around the 18th century, and petitions in particular, using automatic means of text classification. We also test how text pre-processing and different feature representations affect the result, and we examine feature importance for our main class of interest – petitions. Our experiments show that the statistical algorithms NB, RF, SVM, and kNN are indeed very able to classify different genres of historical text. Further, we find that normalisation has a positive impact on classification, and that content words are particularly informative for the traditional models. A fine-tuned BERT model, fed with normalised data, outperforms all other classification experiments with a macro average F1 score at 98.8. However, using less computationally expensive methods, including feature representation with word2vec, fastText embeddings or even TF-IDF values, with a SVM classifier also show good results for both unnormalised and normalised data. In the feature importance analysis, where we obtain the features most decisive for the classification models, we find highly relevant characteristics of the petitions, namely words expressing signs of someone inferior addressing someone superior.

1 Introduction

In many pre-modern and pre-democratic societies, ordinary people had the right to address those in power through written petitions in order to ask for help or confirmation of existing rights. Petitions usually addressed a social and economic superior, for example a court of law, a parliament, a landlord, or even the monarch (Houston, 2014). In other words – petitions allowed the powerless to speak to the powerful. Petitions are a rich historical source that could answer questions about the everyday life of ordinary people in the past. Even so, petitions have been relatively little used in historiography. For this reason, we are involved in an interdisciplinary research project at Uppsala University, funded by the Swedish Research Council, with the goal of enhancing accessibility to and knowledge of Swedish 18th century petitions, and using this source to answer questions about people's ways of supporting themselves and claiming rights in the past.[1] This project, titled "Speaking to One's Superiors: Petitions as cultural heritage and sources of knowledge", is coordinated by the Gender and Work (GaW) research project, conducted at the Department of History, Uppsala University. The GaW project studies how women and men sustained and provided for themselves in Sweden in the period from 1550 to 1800. As part of the project, thousands of historical sources have been gathered, classified and stored in a unique database that has been made accessible for researchers, students, and the general public (Fiebranz et al., 2011).

The computational linguistic part of the project aims to contribute to the field of digital philology and the development of automatised historical text analysis. In this paper, we explore computational approaches, more specifically text classification and feature importance, as means to study petitions and other historical documents. Firstly, we examine the possibility to distinguish petitions from other historical texts using different automatic classification methods. If possible, we also want to see what sort of features that characterise different genres of historical texts, and petitions in particular. Due to the noisy nature of historical data, as well as generally limited resources, we are also interested in studying how much is gained when using different variants of pre-processing methods, and how to best represent our data for a classification task. We show that the different text genres in our data set are certainly possible to classify, using both more state-of-the art and traditional methods. We also

[1]https://gaw.hist.uu.se/petitions/

Proceedings of the 6th LaTeCH-CLfL Workshop, pages 53–64

find that our approach to feature importance analysis, where we obtain the features most decisive for some of the classification models, indeed finds highly relevant and interpretable characteristics of the petitions. As a third step, which we plan to proceed with in future work, we want to examine the possibility to distinguish different parts of the petitions. Research suggests that petitions follow a certain structure, based on a classical rhetorical division (Houston, 2014). It would be interesting to investigate how informative specific parts of the petitions are to a classification task, or where the most relevant features are placed. We hope that our work can facilitate the task of information extraction for historians and other scholars interested in studying petitions further.

2 Related Work

Text Classification (TC), the task of assigning text documents to one or more predefined categories, has traditionally been solved by using supervised learning algorithms such as Naive Bayes (NB) (McCallum et al., 1998), Random Forest (RF) (Xu et al., 2012), Support Vector Machines (SVM) (Joachims, 1998) and K-Nearest Neighbor (kNN) (Yang and Liu, 1999). TC could be implemented either topic-based, paying attention to *what* the text is about, or stylistic, being more concerned with *how* a text is written. While topic-based categorisation often uses models based on "bags of content words", style is somewhat more elusive and can include, but is not limited to, the use of function words and syntactic structures (Argamon et al., 2007). For historical texts, a common application for TC is automatic dating of documents (Niculae et al., 2014; Boldsen and Wahlberg, 2021).

As with most NLP applications, the raw data used for TC typically undergoes several steps of text pre-processing, though the best pre-processing strategy might differ depending on the data set and the TC algorithm at hand (HaCohen-Kerner et al., 2020). Fewer pre-processing steps and less need of annotation could be particularly advantageous for historical text, since its spelling variations, possible OCR-errors and limited resources of (annotated) data pose challenges for NLP tools. A common approach to tackle spelling variations is to view it as a translation task, where character-based statistical machine translation (SMT) (Pettersson et al., 2014a) and corresponding neural methods (NMT) (Tang et al., 2018) have proven to work well. Bollmann (2019) points out that while neural approaches have become popular for a variety of NLP tasks, there is no clear consensus about the state-of-the-art for the task of normalisation. To the best of our knowledge, no method yet has substantially outperformed a character SMT-based approach for historical Swedish.

An important question in TC is how to represent the documents of interest as input to the machine learning algorithms, where common techniques include bag-of-words (BOW) representation in the form of term frequencies or TF-IDF values, or distributed representations of words in the form of word embeddings (Kowsari et al., 2019), such as word2vec (Mikolov et al., 2013a,b) or fastText (Bojanowski et al., 2017). More recent, deep neural language models such as BERT (Devlin et al., 2019) produce contextualised word vectors that are sensitive to the context in which they appear. Such a pre-trained model is commonly fine-tuned to perform a specific task, such as text classification, simply by changing the final output layer. However, due to memory limitations, the maximum length for the input sequence is limited, which is problematic for long documents, although (Sun et al., 2019) have shown that state-of-the-art results can be obtained with 512 tokens, by concatenating text from the head and tail of a document. Another potential challenge when using large language models such as BERT, especially relevant for historical data, are observed instabilities when fine-tuning with small data sets (Zhang et al., 2020).

Instead of applying techniques to standardise variations in orthography, one could also develop tools that are trained on text more similar to the target data. Hengchen and Tahmasebi (2021) have released a collection of Swedish diachronic WE models trained on historical newspaper data. Their models include word2vec and fastText models, trained on 20-year time bins from 1740 to 1880, with two temporal alignment strategies: independently-trained models for post-hoc alignment, and incremental training.

As described in Section 1, the petition project seeks to use historical sources to study how ordinary people claimed their rights and what they did for a living. The latter has been approached by computational manners within the coordinating GaW project, using historical court records and church documents as a source, by implementing a verb-oriented approach to find text passages de-

scribing work activities (Pettersson et al., 2014b). This has also resulted in a web-based tool for automatic information extraction from historical text (Pettersson et al., 2). Though, to use petitions as a source of information has, to the best of our knowledge, not yet been approached by computational manners.

3 Data Collection

As part of our project, we make use of a transcribed collection of 18th century petitions submitted to the regional administration in Örebro, Sweden. In order to compare the petitions to other relevant historical documents, we select data from other genres based on the following criteria: (a) each genre should to be fairly easy to divide into smaller documents in an automatic or semi-automatic manner, (b) the selected genres, and each document within them, should be reasonably similar to the petitions in terms of size (number of tokens) and time period, and (c) the selected genres should vary in terms of similarity in content to the petitions (to the best of our knowledge), with the purpose of having some variety in challenge for the classification models. Given that transcribed historical documents are a limited resource, it is not possible to meet all criteria for every genre, though we strive to come as close as possible. Our selected genres, which will be referred to as classes from now on, can be viewed in Table 1, and is further described in the following sections. The text pre-processing procedures, including tokenisation, normalisation, lemmatisation and part-of-speech (POS) tagging, are described in Section 4.2.

3.1 Petitions

Through the project, a large volume of handwritten 18th century petitions has been scanned and made publicly accessible, and a smaller subset of the petitions have been manually transcribed by historians for refined analysis. We use this transcribed subset in our data set, which consists of petitions written in 1719 and 1782. Also, another set of petitions from another region in Sweden is used in our test set only, with the purpose of evaluating the generalisability of our classification models. This data set, which we from now on describe as ”out-of-domain”, is a small collection of manually transcribed petitions from Västmanland, Sweden.

3.2 Letters

The data subset of letters was collected from the Swedish Diachronic Corpus (Pettersson and Borin, 2022), available online.[2] This class contains texts written by several authors, digitised through OCR-scanning with manual post-correction. Included are the letters of military Jon Stålhammar, who wrote to his wife Sofia Drake, Pehr Wahlström's letters to a friend during a trip in the countryside, the letters of princess Anna Vasa, and a fictional letter conversation written by Karl August Tavaststjerna. Lastly, we have Sophie von Knorring's letters to her home, during a summer trip in 1846, which we use as an out-of-domain test set.

3.3 Laws

The digitised law documents are manual transcriptions provided by Fornsvenska textbanken,[3] also collected from the Swedish Diachronic Corpus. The first data subset, Sveriges Rikes Lag (Law of Swedish Kingdom) consists of two legislations: 'Giftermåls balk' (Marriged Legislation) and 'Missgiernings Balk' (Misdeed Legislation), both from 1734. The second part of the law subset is 'Regeringsformen' (The Instrument of Government) from 1809.

3.4 Parish Protocols

The Gender and Work (GaW) research project, conducted at the Department of History, Uppsala University, studies how women and men sustained and provided for themselves in Sweden in the period from 1550 to 1800 (Fiebranz et al., 2011). We use a smaller set of the GaW corpus, namely a subset of Stora Malm, which are OCR-scanned protocols of parish meetings between the years 1728 and 1812. We select protocols from the years 1728-1741 and 1784-1812 in order to better match the petition data set in terms of time period and numbers of documents. During these parish meetings, the parish's residents met to discuss common matters under the pastor's leadership. These meetings could also include some administration of justice. [4]

3.5 Court Records

We also collect a subset of manually transcribed court records from the GaW corpus. Courts in Sweden in older times dealt with a number of different

[2] https://cl.lingfil.uu.se/svediakorp/

[3] https://project2.sol.lu.se/fornsvenska

[4] https://gaw.hist.uu.se/vad-kan-jag-hitta-i-gaw/kallunderlag/stora-malm—sockenstamman/

Classes and Subsets	Period	# Docs	Train–Test	# Tokens$_{raw}$	# Tokens$_{norm}$
Petitions all	1719–1800	**119**	75/25	**34,286**	**34,302**
Petitions Örebro earlier	1719–1720	51	80/20	16,285	16,286
Petitions Örebro later	1782–1800	60	80/20	15,814	15,812
Petitions Västmanland	1758	8	0/100	2,187	2204
Letters all	1591–1893	**178**	**66/34**	**196,980**	**198,975**
Written by Anna Vasa	1591–1612	23	80/20	7,714	7.690
Written by Jon Stålhammar	1700–1708	84	80/20	49,562	51,142
Written by Pehr Wahlström	1800	17	80/20	29,538	29,536
Written by Karl August Tavaststjerna	1893	23	80/20	87,550	87,983
Written by Sophie von Knorring	1846	31	0/100	22,616	22,624
Laws all	1734–1809	**196**	80/20	**34,798**	**34,792**
Sveriges Rikes Lag	1734	76	80/20	22,709	22,703
Regeringsformen	1809	120	80/20	12,089	12,089
Parish protocols all	1728–1812	**131**	80/20	**158,585**	**160,415**
Stora Malm earlier	1728–1741	46	80/20	59,688	59,910
Stora Malm later	1784–1812	85	80/20	98,897	100,505
Court records all	1691–1771	**137**	72/28	**251,351**	**263,899**
Underåker	1691–1700	22	80/20	119,330	124,622
Åsbo	1707–1716	15	80/20	7,750	7,754
Linköping	1709–1710	86	80/20	79,869	86,794
Skellefteå	1771	14	0/100	44,402	44,729
All		**761**	**74/26**	**676,000**	**692,383**

Table 1: Overview of the data sets with information about period, number of documents, proportions of training and test data, and number of tokens: unnormalised (raw) vs. normalised.

types of cases. The court records therefore contain various types of text files, including court documents from criminal cases, accounts of and the settlement of civil disputes, as well as the handling of various administrative cases.[5] These court records are from different locations in Sweden; Underåker, Åsbo, Linköping and Skellefteå, where we use the documents from Skellefteå as our out-of-domain test set.

4 Method

4.1 Text Classification Models

We first make use of the traditional statistical algorithms NB, RF, SVM, and kNN through Scikit Learn's implementations (Pedregosa et al., 2011): MultinomialNB, RandomForestClassifier, LinearSVC, and KNeighborsClassifier. A grid-search is performed to find the optimal hyperparameter setting for each algorithm, where we run a 5-fold cross validation on a unnormalise version of our training data vectorized with TF-IDF (using a rather narrow combination of parameters to limit the search). We refer to this unnormalised data set as a raw version of our corpus. The selected hyperparameter settings can be found in Appendix A. To further examine different manners to represent our data, we also fine-tune a pre-trained BERT model for a later experiment, described in Section 4.4.

We perform a multiclass classification with each algorithm. Even though a binary classification would be sufficient enough to explore whether the models can distinguish petitions from other historical texts, we find it interesting to also study how well other historical genres of texts are separable by automatic means.

4.2 Data Pre-Processing

Our TC experiments are run on several versions of our data set, using different amounts of preprocessing. As a baseline, we use a raw version of our data set. We experiment by adding the preprocessing steps of spelling normalisation, lemmatisation and selection of certain POS tags. The latter is done with the aim of capturing terms that are more informative. Our data set is normalised

[5] https://gaw.hist.uu.se/what-can-i-find-in-gaw/sources-in-gaw/dombocker-i-gaw/

using the SMT-based approach of Pettersson et al. (2014a), which is available as an online tool[6] (the normalised version of our data set, compared to the raw version, differs a bit in number of tokens since some non-alphanumeric characters are treated and separated differently). The annotation is done with Språkbanken's Sparv pipeline version 4.1.1 (Borin et al., 2016), including tokenisation, POS tagging using the Stanza tagger (Qi et al., 2020), trained on SUC3[7] with Talbanken_SBX_dev[8] as development set, and lemmatisation using the Saldo lexicon (Borin et al., 2013).

4.3 Topic-based vs Stylistic Classification

In order to see what types of features are the most informative to our models and how stable our predictions are, we perform both a variant of a topic-based approach and a more stylistic-like classification. A topic-based classification is covered by our approach to select only certain, more content-like POS tags, including nouns, proper nouns, adjectives, verbs, and adverbs. To perform a stylistic classification, we instead target the complement of those tags (though removing foreign words, delimiters, and cardinal and ordinal numbers) to use function words as stylometric features.

4.4 Data Representations

We also try different approaches to represent our historical texts. As a baseline, we vectorise the texts using term frequencies. First, we compare the use of term frequencies with TF-IDF scores. Second, we use a raw, unnormalised version of our data to try language models implemented for historical Swedish texts. Here, we make use of the Swedish pre-trained WEs by Hengchen and Tahmasebi (2021). We try both their Word2vec and fastText models,[9] using the incremental trained embeddings from 1740 up to the year of 1800 in order to best match our data. For these experiments, we follow the cleaning procedure described in Hengchen and Tahmasebi (2021) by lowercasing the text, removing all characters not belonging to the Swedish alphabet (including digits and punctuation marks), and removing tokens with the length of two characters or smaller. For simplicity, all out-of-vocabulary (OOV) words get a plain zero embedding. To obtain one vector for each text in our data set, we use the pre-trained word2vec and fastText to look up individual words, and average all word embeddings for each text.

As a third approach, we use language models implemented for modern Swedish texts in combination with a normalised version of our data. We work with a word2vec model[10] by Kutuzov et al. (2017), trained on the Swedish CoNLL17 corpus. We also make use of a Swedish fastText model[11] by Grave et al. (2018), trained on Wikipedia data. As before, all OOV words get a plain zero embedding, and we average the word vectors for each text to get one vector per document.

As a final text classification approach, we use a pre-trained Swedish BERT model created by KBLab (Malmsten et al., 2020), and fine-tune the model on the classification task with our (relatively small) data. We carry out the experiment in Google Colaboratory with one NVIDIA Tesla T4 GPU, and load the BERT model using HuggingFace's Transformers library (Wolf et al., 2019). Like Holmer and Jönsson (2020), we use the default PyTorch cross-entropy loss function utilised by HuggingFace's Transformers together with the hyperparameters learning rate=2e-5, and epochs=4, with an exception of the batch size, in which we use 16. The input sequence is limited to 512 tokens, so we include the 510 first tokens of each document, together with the required [CLS] and [SEP] tokens.

4.5 Feature Importance

In our second task, we aim to study the characteristics of our main class of interest - the petitions. Here, we will move in the other direction and use the method of text classification in order to extract the most important features for our class. We make use of the MultinomialNB classifier and the LinearSVC with the same settings and models that we use in Section 4.2. Through their implementation in SciKit Learn, the importance of each feature for each class is calculated and easily accessible. The feature importance scores that we use are calculated in different manners for the different classifiers. The MultinomialNB classifier uses the empirical log probability of features given a class, $P(x_i|y)$, to score the importance of each feature. The LinearSVC has the attribute coef_attribute, which assigns weights to the features for each class versus all other classes (coefficients in the primal prob-

[6] https://cl.lingfil.uu.se/histcorp/tools.html
[7] https://spraakbanken.gu.se/en/resources/suc3
[8] https://spraakbanken.gu.se/resurser/talbanken
[9] https://zenodo.org/record/4301658 (June, 2022)
[10] http://vectors.nlpl.eu/repository/ (July, 2022)
[11] https://github.com/facebookresearch/fastText/blob/master/docs/crawl-vectors.md (July, 2022)

lem). We are interested to see which features get high rankings consistently. Therefore, as a final step, we do exactly this by merging both of our ranked results. We select the top important features by examining which ones that often appear in the top ranked results throughout all approaches. We merge the rank of each feature in three steps: (1) extract the top 100 features for each approach to a sorted list, (2) calculate the average rank for each feature that appears in at least one of the sorted lists (features without a rank in a specific list will get a ranking score of 101 for that list), and (3) rank the features by their average score.

4.6 Evaluation Procedure

To evaluate the classification model performance for each class, we use precision, recall and F1 metrics. When looking at the overall performance for each classifier, including all classes, and each data representation, we use the metrics macro average F1-scores and accuracy. Macro average F1 metric computes a simple average of F1-score over classes, with equal weight to each class (Manning et al., 2008). We also perform an error analysis, where we look more closely at what types of errors the models produce. Furthermore, we evaluate how well our models are able to generalise by computing recall scores for the data sets not included in the training set (which we refer to as "out-of-domain", see Section 3). It is noteworthy that due to the limited amount of data points in these new data sets, it is difficult to draw any certain general conclusions for this type of experiment. Even so, we still include this experiment to get an indication of our models' generalisation capacity.

The result from the feature importance analysis is not quantitatively evaluated. Instead, the result is qualitatively interpreted and discussed.

5 Results and Discussion

5.1 Text Classification

The results of the text classification task for the different classes of our data set can be viewed in Table 2. As a baseline, we have here used a raw (unnormalised) version of our data set, vectorised with TF-IDF values. Though the result differs somewhat between the classes and the different TC models, we can see that all models are able to distinguish between these different classes quite well. Generally, the classes of letters, laws and petitions are easier for the models to differentiate, while parish protocols and law documents get lower scores. The differences between classes is further discussed in Section 5.1.2. Out of all the models, kNN has the overall lowest performance for this raw version of our data set, while the SVC model gets the strongest result.

Class	NB	RF	SVC	kNN
Petitions	94.7	93.8	**98.4**	76.4
Letters	**100.0**	94.8	99.2	96.7
Laws	**100.0**	92.0	**100.0**	97.5
Parish	75.4	**88.1**	82.5	76.9
Court	78.1	78.1	**83.6**	76.5
Macro avg F1	89.6	89.4	**92.7**	84.8
Accuracy all	91.3	90.3	**93.8**	87.2

Table 2: F1 scores, macro avg F1 scores and overall accuracy for raw (unnormalised) data, vectorised with TF-IDF values.

5.1.1 The Impact of Pre-Processing

To study the impact of different amounts of pre-processing, we compare the performance when feeding our classification models with different versions of our data: raw tokens, normalised tokens, normalised lemmas, and finally normalised content words (nouns, proper nouns, adjectives, verbs, and adverbs). The result in Table 3 shows that normalisation has a positive effect for the SVC and the kNN models, a modest positive impact for the NB model, while the RF model instead decreases in performance. By contrast, using normalised lemmas seems to harm the performance of all models. It may well be that errors in lemmatisation lead to these results (we did not evaluate the lemmatisation quality). The results indicate that content words are important features for all classifiers that essentially increase the models' performance, something we investigate further in Section 5.1.3. Even so, we can also conclude that the baseline results, in which we use all tokens, are quite high, and therefore imply that the classes have characteristics that sets them apart from each other. Finally, when comparing the classifiers, we see that even if kNN has the highest performance when using normalised content words, the SVC has the most consistently high results no matter the amount of pre-processing.

5.1.2 Error Analysis and Class Comparison

To study the differences between classes, we look at the recall, precision and F1-score for all classes when using one of the best performing models (the

Pre-processing	NB	RF	SVC	kNN
Raw tokens	89.6	89.4	92.7	84.8
Norm tokens	90.8	88.1	95.0	89.3
Norm lemmas	88.4	83.7	92.8	84.9
Norm content words	**91.4**	**93.7**	**96.5**	**97.0**

Table 3: Macro average F1-scores for the TC models when using different amounts of pre-processing. Normalisation is performed using an SMT-based approach described in Section 4.2.

Class	Prec	Rec	F1
Petitions	100.0	100.0	100.0
Letters	100.0	98.3	99.2
Laws	97.6	100.0	98.8
Parish	83.9	100.0	91.2
Court	100.0	87.2	93.2

Table 4: Precision, recall and F1-scores for all classes when using one of the best performing models (SVC and TF-IDF values of normalised content words)

consistently high performing SVC together with TF-IDF values of normalised content words). As can be seen in Table 4, the model makes few or no mistakes regarding petitions, letters and law documents. Parish protocols and court records are harder for the model to separate. As we can see in the precision and recall scores for these classes, the most common error for the model is to label parish protocols as court records. This is not surprising, since the parish meetings of this time period also could contain testimonies and administration of justice cases, which we write about in Section 3.4.

When it comes to the models' abilities to generalise to new data sets of petitions, law documents and court records, we use recall scores for the in-domain and out-of-domain data sets, presented in Table 5. As described in Section 3, the out-of-domain data are petitions and court records from other geographical areas, and letters written by other authors, than those seen in the training data.

We can see that the models generally are doing well for the class letters, while the results for petitions and especially court records vary considerably between the models. It is difficult to draw any safe conclusions due to very few data points in our new data sets, but the results suggest that letters of various authors have more features in common than petitions and court documents from different regions, at least for our chosen time periods.

TC	**Petitions**		**Letters**		**Court**	
model	ID	OOD	ID	OOD	ID	OOD
NB	100.0	62.5	100.0	100.0	100.0	21.4
RF	100.0	87.5	99.3	96.8	95.9	78.6
SVC	100.0	100.0	100.0	96.8	100.0	64.3
kNN	100.0	62.5	100.0	96.8	100.0	92.9

Table 5: Generalisability of the TC models: recall for petitions, letters and court records when using out-of-domain (OOD) and in-domain (ID) data.

Features	NB	RF	SVC	kNN
All words	88.4	83.7	92.8	84.9
Content words	**91.4**	**93.7**	**96.5**	**97.0**
Function words	68.4	76.6	83.3	69.5

Table 6: Macro average F1-scores for the TC models when using lemmas for the whole corpus, only content words, and only function words (all data normalised and lemmatised).

5.1.3 Topic-Based vs. Stylistic Classification

For this experiment, we use a normalised and lemmatised version of our data set, represented with TF-IDF values. We compare the results when using all lemmas in our data set, using only the lemmas of content words, and using only the lemmas of function words (see more in Section 4.3). As can be seen in Table 6, the best results are reached when using only content words as features. In contrast, the classification does not benefit from a stylistic approach, as using only function words harm the models.

5.1.4 The Effect of Different Data Representations

As a final experiment for our TC task, we test the performance of our models when using different types of data representation. Here, we use term frequencies as baseline, and compare it with the use of TF-IDF values, and Swedish pre-trained word2vec and fastText word embeddings. We run experiments on both a raw version and a normalised version of our data. For the raw version of our data, we use the word embedding models trained on historical Swedish texts, and for the normalised data, we use the corresponding pre-trained word embeddings trained on contemporary Swedish text (cf. Section 4.4). To reduce the comparisons, we here show the results for different data representation when using SVC, since this classifier provides the most consistently high results result. For the nor-

Using raw (unnormalised) data				
	Acc	Prec	Rec	F1
Term freq + SVC	89.2	89.4	89.9	87.9
TF-IDF + SVC	93.8	93.4	94.0	92.7
hist w2v + SVC	90.3	89.8	89.6	88.5
hist ft + SVC	94.9	93.6	94.5	93.9
Using normalised data				
Term freq + SVC	86.7	86.6	87.7	84.9
TF-IDF + SVC	95.9	95.3	95.9	95.0
modern w2v + SVC	95.9	95.1	95.1	95.1
modern ft + SVC	88.2	88.5	87.6	85.6
BERT classifier	**99.0**	**99.0**	**98.6**	**98.8**

Table 7: Comparing different data representations ran with SVC, and a fine-tuned BERT classifier. We use term frequencies, TF-IDF scores, word2vec vectors and fastText vectors for either historic or modern Swedish text, respectively. The results are presented as accuracy, and macro averaged precision, recall and F1 scores.

malised data set, we also include the results when using a pre-trained BERT model, fine-tuned for our classification task.

Even though we have a very limited amount of data, the BERT model is able to learn well from the fine-tuning, and outperforms all other data representations classified with SVC. Also, for the normalised data, both TF-IDF and word2vec representations get reasonably high scores. The use of fastText word embeddings gets one of the lowest results for the normalised data set, which is presumably explained by the fact this model is trained on a domain (Wikipedia data) relatively far from our historical data set. It is worth mentioning, though, that the BERT model may have had an advantage compared to the other models by only seeing the beginning of each text. It is possible that the first part of the documents provides the most beneficial information for a classification task, and this is a question we mean to follow up in future work.

For the raw data set, the use of historical fastText word embeddings performs the best with a F1 score at 93.9, though the use of TF-IDF values is not far behind with a F1 score at 92.7. The use of historical word2vec embeddings gets a rather low result, which is most likely explained by the number of OOV words for our data set matched with those embeddings (162,999 OOV words of the 676,000 tokens in our data set).

Top 30 features petitions
vy, vi, för, eder, nå, ödmjuk, nådig, höga, baron, nådes, riddare, ûti, herr, hemman, ock, landshövding, tjänare, nåd, högvälborne, allra, jag, hög, herre, ed, kongl, ûnder, högvälborne, är, nû, anhålla
[*we, we, for, your, grace/reach, humble, gracious, high, baron, grace, knight, in/within, mister, home, and, governor, servant, grace, "highness", the most, I, high, mister, oath, royal, under, "highness", is, now, request*]

Table 8: Top features for the petition class in Swedish (top) and with English translations (bottom).

5.2 Feature Importance

For this analysis, we will focus on the results for the class of petitions (the results for the other classes are displayed in Appendix B). We use a normalised and lemmatised version of our data set in order to get less inflected word forms and a more interpretable result. Even though the TC task benefits from only including content words (see Table 3), we here include all tokens in our data set so as not to exclude any part of speech.

As described in Section 1, writing petitions was a means for ordinary people to ask a social and economic superior for help or make complaints. This is also quite salient when inspecting the results from our feature importance experiment. Table 8 shows many tokens that express signs of someone inferior addressing someone superior (e.g. "grace", "humble", "servant", "highness"). Some of the features are redundant, since these are spelling variations of the same word (e.g. *vy/vi* 'we', *högvälborne/högvälborne* 'highness') that failed to be normalised. Overall, we find that our chosen method for feature importance reveals highly relevant and interpretable characteristics of the petitions.

6 Conclusions

In this paper, we present a study of text classification and feature importance applied to historical Swedish text, with a special focus on petitions. We test the performance of both traditional and newer classification algorithms, and we examine how text pre-processing and different types of feature representation affect the result. We also analyze feature importance for our main class of interest: petitions.

The text classification results show that the statistical classification algorithms NB, RF, SVM, and

kNN are indeed very able to distinguish between our different classes of historical text. We also find that pre-processing in the form of normalisation has a positive impact on the classification models, and that content words are particularly informative. Using a normalised and lemmatised version of our data set classified with an SVM classifier achieves a macro-averaged F1-score at 96.5, and only targeting content words with a kNN model pushes the score up to 97.0.

We also test how to best represent our data for a classification task. Using a more state-of-the-art method, a pre-trained BERT model, fine-tuned for our classification task and fed with a normalised version of our data set outperforms all other classification experiments with a macro average F1 score at 98.8. However, using much less computationally expensive methods with an SVM classifier also show quite good results for both a raw and a normalised version of our data set. For the raw data, using fastText embeddings, trained on historical Swedish texts, gave the best F1 score at 93.9. For the normalised data, fastText embeddings trained on contemporary Swedish resulted in a F1 score at 95.1. Even using such a simple approach as TF-IDF values in combination with an SVM classifier gave quite good results, both for the raw and normalised data set, with F1 scores at 92.7 and 95.0, respectively. We believe that this could be explained by the small amount of data used, and also that the classes in our data set have characteristics that the classifiers are able to differentiate quite effectively.

In the feature importance analysis, we make use of our text classification task and obtain the features most decisive for some of the classification models. We find that this method reveals features that are highly relevant and interpretable characteristics of the petitions, namely tokens that express signs of someone inferior addressing someone superior.

For future work, we are interested in further exploring text classification and feature importance as methods to analyse petitions. As mentioned in Section 1, research indicate that petitions follow a certain disposition. With this in mind, and given our results in this paper, we plan to investigate if petitions could be segmented by automatic means. If possible, we also want to examine where the most relevant features typically are placed. The main goal with these steps would be to facilitate and improve the task of information extraction for historians and other scholars interested in studying petitions further. Another area of improvement would be to test if other methods of classification would work better for a small, historical data set such as ours, in particular additional deep learning techniques. It would be interesting to make use of new generations of language models adapted to historical texts, if available.

References

Shlomo Argamon, Casey Whitelaw, Paul Chase, Sobhan Raj Hota, Navendu Garg, and Shlomo Levitan. 2007. Stylistic text classification using functional lexical features. *Journal of the American Society for Information Science and Technology*, 58(6):802–822.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146.

Sidsel Boldsen and Fredrik Wahlberg. 2021. Survey and reproduction of computational approaches to dating of historical texts. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 145–156.

Marcel Bollmann. 2019. A large-scale comparison of historical text normalization systems. *arXiv preprint arXiv:1904.02036*.

Lars Borin, Markus Forsberg, Martin Hammarstedt, Dan Rosén, Roland Schäfer, and Anne Schumacher. 2016. Sparv: Språkbanken's corpus annotation pipeline infrastructure. In *SLTC 2016. The Sixth Swedish Language Technology Conference, Umeå University, 17-18 November, 2016*.

Lars Borin, Markus Forsberg, and Lennart Lönngren. 2013. Saldo: a touch of yin to wordnet's yang. *Language resources and evaluation*, 47(4):1191–1211.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Rosemarie Fiebranz, Erik Lindberg, Jonas Lindström, and Maria Ågren. 2011. Making verbs count: the research project 'gender and work' and its methodology. *Scandinavian Economic History Review*, 59(3):273–293.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Yaakov HaCohen-Kerner, Daniel Miller, and Yair Yigal. 2020. The influence of preprocessing on text classification using a bag-of-words representation. *PloS one*, 15(5):e0232525.

Simon Hengchen and Nina Tahmasebi. 2021. A collection of Swedish diachronic word embedding models trained on historical newspaper data. *Journal of Open Humanities Data*, 7(2):1–7.

Daniel Holmer and Arne Jönsson. 2020. Comparing the performance of various Swedish BERT models for classification. In *Eighth Swedish Language Technology Conference (SLTC2020). Organised by University of Gothenburgå, Sweden*.

Rab Houston. 2014. *Peasant petitions: social relations and economic life on landed estates, 1600-1850*. Springer.

Thorsten Joachims. 1998. Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning*, pages 137–142. Springer.

Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. 2019. Text classification algorithms: A survey. *Information*, 10(4):150.

Andrei Kutuzov, Murhaf Fares, Stephan Oepen, and Erik Velldal. 2017. Word vectors, reuse, and replicability: Towards a community repository of large-text resources. In *Proceedings of the 58th Conference on Simulation and Modelling*, pages 271–276. Linköping University Electronic Press.

Martin Malmsten, Love Börjeson, and Chris Haffenden. 2020. Playing with words at the National Library of Sweden – making a Swedish BERT.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

Andrew McCallum, Kamal Nigam, et al. 1998. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. Citeseer.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Vlad Niculae, Marcos Zampieri, Liviu P Dinu, and Alina Maria Ciobanu. 2014. Temporal text ranking and automatic dating of texts. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 17–21.

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

Eva Pettersson and Lars Borin. 2022. *Swedish Diachronic Corpus*. Darja Fišer and Andreas Witt, CLARIN, Berlin: deGruyter.

Eva Pettersson, Jonas Lindström, Benny Jacobsson, and Rosemarie Fiebranz. 2. Histsearch-implementation and evaluation of a web-based tool for automatic information extraction from historical text. In *HistoInformatics@ DH*, pages 25–36.

Eva Pettersson, Beáta Megyesi, and Joakim Nivre. 2014a. A multilingual evaluation of three spelling normalisation methods for historical text. In *Proceedings of the 8th workshop on language technology for cultural heritage, social sciences, and humanities (LaTeCH)*, pages 32–41.

Eva Pettersson, Beáta Megyesi, and Joakim Nivre. 2014b. Verb phrase extraction in a historical context. In *The First Swedish National SWE-CLARIN Workshop, Swedish Language Technology Conference, 13th Nov 2014, Uppsala, Sweden*.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune BERT for text classification? In *China national conference on Chinese computational linguistics*, pages 194–206. Springer.

Gongbo Tang, Fabienne Cap, Eva Pettersson, and Joakim Nivre. 2018. An evaluation of neural machine translation models on historical spelling normalization. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1320–1331, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Baoxun Xu, Xiufeng Guo, Yunming Ye, and Jiefeng Cheng. 2012. An improved random forest classifier for text categorization. *J. Comput.*, 7(12):2913–2920.

Yiming Yang and Xin Liu. 1999. A re-examination of text categorization methods. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 42–49.

Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q Weinberger, and Yoav Artzi. 2020. Revisiting few-sample BERT fine-tuning. *arXiv preprint arXiv:2006.05987*.

A Hyperparameter Settings for Text Classification Task

When a random state is used, we set the seed to 11 to enable reproducible output. For all other hyperparameters, not specified here, we use the default settings.

MultinomialNB:	`alpha` = 0.5 `fit_prior` = False
RandomForestClassifier:	`bootstrap` = False `max_features` = 0.3
LinearSVC:	`C` = 5.0
KNeighborsClassifier:	`n_neighbors` = 1

Table 9: Hyperparameter settings for statistical text classification models.

B Top 30 Features for Classes Other than Petitions

Top 30 features letters
jag, du, vara, vi, gud, ha, hälsa, en, den, hjärta, väl, skriva, vÿ, kär, god, om, brev, vän, liv, skola, totus, stållhammar, vilja, fru, och, intet, att, hon, inte, han [*I, you, to be, we, God, to have, greet, one/a, it/that, hart, well, write, we(?), dear/in love, good, about, letter, friend, life, school, Totus, Stålhammar, will, wife, and, nothing/not, to/that, she, not, he*]
Top 30 features laws
eller, stånd, konung, riks, statsråd, rike, ej, man, domstol, böte, då, justitie, riksdag, äga, daler, kap, domare, sån, utskott, varda, lag, stats, miste, bo, särskild, ämbete, sätt, gälla, straffa, och [or, estate, king, national, minister, kingdom, not, one/man, court, fine, then, justice, parliament, own, daler, chapter, judge, such, committee, be/become, law, governmental, lost, live, specific/distinct, office, way/manner, apply/concern, penalise, and]
Top 30 features parish protocols
församling, att, församl, kyrka, på, socken, sexman, herr, st, sockenman, sockenstämma, pastor, uppläsa, icke, ock, person, barn, malm, år, uti, eric, per, maneck, av, gammal, ingen, sig, dr, fidem, man [*parish/assembly, to/that, parish/assembly, church, on, parish, elected representative in a parish, mister, Saint/pieces of, man of the parish/assembly, parish meeting, reverend, read, not, also, person, child, Malm, year, in/within, Eric, Per, Manech, of/off/by, old, no one, oneselfe, dr, (latin) faith, man*]
Top 30 features court records
och, ner, en, han, de, rådstuga, magistrat, där, niels, intet, borgmästare, rådman, rätt, ha, johan, 1709, sak, sal, stad, linköping, ordinarie, klingenberg, hon, 1710, samuel, pyttner, jöns, behm, här, haraldsson [*and, down, one/a, he, they/those, town hall, magistrate, there, Niels, nothing/not, mayor, district court judge, right/just/court, to have, Johan, 1709, think/matter/cause, ward, city, Linköping, ordinary, Klingenberg, she, 1710, Samuel, Pyttner, Jöns, Behm, here, Haraldsson*]

Table 10: Top features for genres other than petitions in Swedish (top) and with English translations (bottom).