Effects of Evolutionary Linguistics in Text Classification

Julia Efremova[1(✉)], Alejandro Montes García[1], Jianpeng Zhang[1], and Toon Calders[1,2]

[1] Eindhoven University of Technology, Eindhoven, The Netherlands
{i.efremova,a.montes.garcia,j.zhang.4}@tue.nl
[2] Université Libre de Bruxelles, Brussels, Belgium
toon.calders@ulb.ac.be

Abstract. We perform an empirical study to explore the role of evolutionary linguistics on the text classification problem. We conduct experiments on a real-world collection with more than 100.000 Dutch historical notary acts. The document collection spans over six centuries. During such a large time period some lexical terms modified significantly. Person names, professions and other information changed over time as well. Standard text classification techniques which ignore temporal information of the documents might not produce the most optimal results in our case. Therefore, we analyse the temporal aspects of the corpus. We explore the effect of training and testing the model on different time periods. We use time periods that correspond to the main historical events and also apply clustering techniques in order to create time periods in a data driven way. All experiments show a strong time-dependency of our corpus. Exploiting this dependence, we extend standard classification techniques by combining different models trained on particular time periods and achieve overall accuracy above 90 % and macro-average indicators above 63 %.

1 Introduction

Text classification is a popular machine learning problem with many applications, such as: classification of news into groups, classification of fairy tales according to their genres, filtering emails into spam and not, mining opinions... [1,9].

Research on text classification has mainly focused on topic identification, keywords extraction, sparsity reduction and ignored the aspects of language evolution across different time periods. Research that investigates temporal characteristics of documents and their role in text classification has been scarce so far.

Evolutionary linguistics (EL) is the study of the origin and evolution of languages. It plays an important role in text classification. As a result of the evolution of language vocabulary changes. New words appear and other become outdated. Person names also vary. As an example, more than 100 variants of the first name *Jan* are known, (e. g. *Johan*, *Johannes*, *Janis*...) [7]. These modifications change the characteristics of the text, the weights of terms and therefore can

A.-H. Dediu et al. (Eds.): SLSP 2015, LNAI 9449, pp. 50–61, 2015.
DOI: 10.1007/978-3-319-25789-1_6

affect the classification results. Standard classification methods do not consider a time period of the documents to which they belong to [8,16]. They typically use supervised machine learning methods and compute weights of words in the collection.

In this paper, we investigate the role of EL from various perspectives. We make an extensive empirical study to identify robustness of a classifier in the case when the training and test data belong to different time periods. We analyse an impact of EL on the class distribution, correlation between term frequency and time periods, change in the vocabulary across several time periods. In the next part we design a simple framework that enhances existing techniques. The framework incorporates EL aspects and trains the model on relevant examples.

We carry out our experiments on the collection of Dutch historical notary acts from the 15^{th} to the 20^{th} century. Available data spans large time period and we analyse temporal aspects under the context of historical events. To identify main time periods in the history of the Netherlands we consider a time-line proposed by the Rijksmuseum in Amsterdam and split the data into several time periods. Moreover, we identify optimal time periods in a date-driven way and apply year clustering. We present results that show strong term-category-time period dependencies.

The contribution of this paper is summarised as follows:

1. We make an empirical study of the aspects of evolutionary linguistics applied to a large collection of historical data.
2. We develop a framework that incorporates temporal dependencies and, as a result, improve the quality of text classification.

2 Related Work

There is some work available regarding time aspects and empirical studies in text classification. Mourao et al. present an empirical analysis of temporal data on news classification [11]. The impact of empirical methods in information extraction is described by Cardie in [3]. Salles et al. analyse the impact of temporal effects in automatic document classification [14]. Dalli and Wilks [4] investigate the opposite research question. They predict the date of document using the distribution of word frequencies over time. Mihalcea and Nastase [10] identify time period by analysing changes in word usage over time. They make word disambiguation in order to predict time period. In our work we predict a category of a document and assume that a certain category is more common in a certain time period.

The main contribution of our work as compared to previous efforts can be summarise as follow: obtaining an insight of the role of EL in document classification, an empirical study of temporal characteristics and the improvement of classifier performance by integrating temporal information.

3 Data Description and General Approach

We use a dataset of historical notary acts provided by Brabant Historical Information Center[1]. The documents are available in Dutch. Archive notary acts correspond to a wide rage of topics such as sale agreements, inheritance acts, resolutions, etc. It was identified 88 different categories in total.

Volunteers manually digitised notary acts and to some of them assigned a single category that briefly describes document type. However a large number of documents still has to be classified. The overall collection consists of 234,325 notary acts and 115,673 of them contain an assigned category and also a specified date of the document.

Another important characteristic is that the dataset is not balanced regarding the number of documents per each category. The largest categories are *transport (property transfer)*, *verkoop (sale agreement)* and *testament (inheritance act)* and they contain around 20 %, 15 % and 11 % of classified documents respectively. However there are a lot of very small categories that have a support value less than 1 %.

We start by pre-processing the documents and remove punctuation marks and non-alphabetical symbols. In the next step we transform the text to the lower case. AFter that, we create a special feature for each remaining token and apply *term frequency inverse document frequency* (TF-IDF) [8] to compute feature vector. The output of the feature extraction step is a feature set with numerical values. Initially we try a number of classifiers to predict a category of the documents and continue to use the one that has the highest performance on our dataset. We use classifiers from the scikit-learn python tool[2] [13].

4 Empirical Study

In this section, we describe time partitions and show an impact of the training set on the classifier accuracy. Afterwards, we present an analysis of different factors such as the sampling effect within the given time frame, category distribution over time, time-specific vocabulary and correlation between term frequency and time periods.

Identifying Time Frames. We split a large time period into smaller pieces. We define a set of time partitions as $\mathcal{T}$. Each $\mathcal{T}_i$ is described by two time points t_i and t_{i+1} that are the beginning and the ending of a time frame respectively. A document $\mathcal{D}_i$ belongs to the $\mathcal{T}_i$ when $t_i \leq date(\mathcal{D}_i) \leq t_{i+1}$. First, we consider major historical events and follow the time line proposed by the Rijksmuseum[3], later we present an approach to obtain optimal time periods in a data-driven way. We identify seven major periods in Dutch history presented in Table 1.

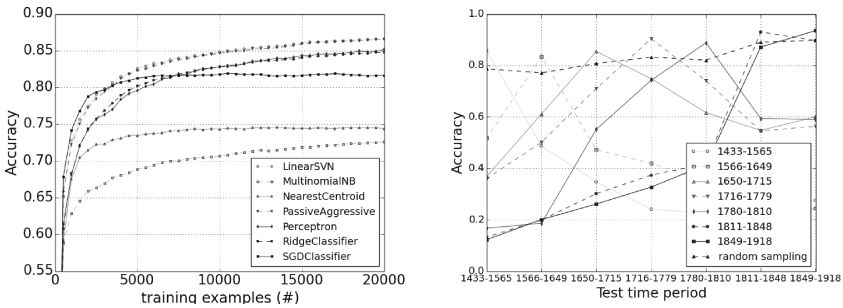
We do not consider periods after 1918 since they are relatively recent and notary acts are not publicly available yet.

[1] http://www.bhic.nl/.
[2] http://scikit-learn.org/.
[3] http://goo.gl/YZvP9q.

Table 1. The timeline of Dutch history obtained from Rijksmuseum.

1433 - 1565	Burgundian and Habsburg period	1
1566 - 1649	Revolt and the Rise of the Dutch Republic	2
1650 - 1715	Republic at war with its neighbours	3
1716 - 1779	Dutch republic	4
1780 - 1810	Patriots, Batavian Republic and the French	5
1811 - 1848	Kingdom of the Netherlands	6
1849 - 1918	Modernisation	7

Size of the Training Set. We also illustrate the impact of the size of the training set on the classifier accuracy. It allows to clarify the effect of the training size and to distinguish it from the time effect. The difference in a number of training examples in each time period affects the classifier accuracy. We use a number of classifiers, namely: *Support Vector Machines* classifier with a linear basis kernel function, *multinomial naive Bayes*, *nearest centroid*, *passive aggressive, perceptron, ridge regression* and *stochastic gradient descent* [15]. We divide data into fixed subsets (training and test) and vary a size of the training data from 1.000 to 20.000 examples. Figure 1(a) demonstrates a clear dependency between the overall classifier accuracy and the number of training examples. The more training examples we use, the better accuracy a classifier achieves. We compared a



(a) Classification accuracy as a function of a training set size.

(b) Accuracy results as a function of different training and test time periods.

Fig. 1. Analysis of classification accuracy. In the case (a) each line on the graph represents applied classifiers such as: Support Vector Machines, multinomial naive Bayes, nearest centroid, passive aggressive, perceptron, ridge regression and stochastic gradient descent. In the case (b) the lines in the graph indicate the performance of SVM classifiers trained on one specific time period, applied on all the different time periods. It can be seen clearly that a classifier trained on period $\mathcal{T}_i$ when tested on period $\mathcal{T}_i$ (cross-validated performance figures) outperforms all other classifiers

number of classifiers and a SVM constantly showed the highest performance. Therefore we choose the SVM as a classifier for further experiments.

Sampling Effect within Given Time Frame. To demonstrate the effect of sampling within a particular time period on the text classification results we associate each document $d_i \in \mathcal{D}$ to the appropriate time period $\mathcal{T}_i$. The number of documents in each period varies significantly. The total number of unique categories in every $\mathcal{T}_i$ is also different. Table 2 shows the statistics about splitting the dataset into time periods.

Table 2. The number of documents and categories in each time period

	1433-1565	1566-1649	1650-1715	1716-1779	1780-1810	1811-1848	1849-1918
Categories	45	46	70	78	75	52	34
Documents	6166	3594	11550	25914	17301	26087	25538

The idea of this experiment is to construct training and test sets using documents from non-overlapping time frames. More specifically, we use the partition $\mathcal{T}_i$ to train a classifier and test it consequentially on all $\mathcal{T}_j$ with $i \neq j$. We evaluate a change in the classification results when the average time difference between training and test documents gradually increases.

We divide the data collection into partitions according to the identified timeline. Then we train a classifier on one partition and evaluate on all the others. When training and test belong to the same time period we apply 10-fold cross validation. Figure 1(b) demonstrates the overall performance accuracy. Each division on the X axis on the plot represents a fixed time period which was used for training a classifier and dots show results on test sets from different time periods. Clearly we see that all the peaks on the graph occur when training and test time partitions are equal $\mathcal{T}_{train} = \mathcal{T}_{test}$.

In order to compare our results we used random sampling to construct a training set. To avoid the fact that classifiers are sensitive to the number of training examples, we randomly select from every $\mathcal{T}_i$ equal number of documents.

Category Distribution over Time. We analyse category distribution over the time. Figure 2 represents a percentage of each type of documents in different time periods. We denote as *other* the categories that are not in the list of top-10 the most frequent categories. We see that the proportion of other categories gradually decreases over time leaving space for the top-10 categories.

Dealing with a large number of small categories requires additional efforts. They do not always have a sufficient number of training examples and can easily be confused with larger neighbours. In our previous work [6] we clustered categories according to their frequencies and identified small categories in two steps. In this work we analyse how time segmentation affects the classification results for both large and small categories.

Category distributions also confirm the existence of time dependencies in a dataset.

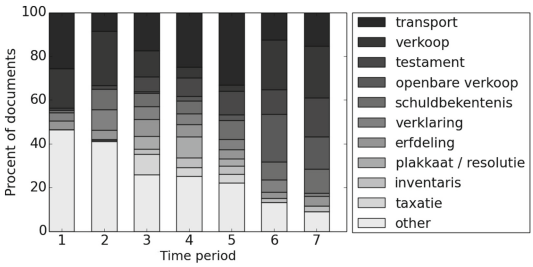


Fig. 2. The distribution of top 10 categories in each time period

Temporal Term Weighting. In this section we test if the occurrence of the term is independent of the time period. To do so, we use χ^2 statistic [12]. We compute χ^2 for each unique term in the corpus. We do not consider terms which occur less than 0.5 % in a collection.

Table 3 shows a number of terms and their corresponding p-value across the overall collection (not the balanced subset). The larger the p-value is, the more terms meet these requirements. The probability of 0.1 is the maximum bound when the observed deviation from the null hypothesis is significant.

Table 3. Time period analysis. Number of terms bounded by p-value

p-value	0.25	0.2	0.15	0.1	0.05	0.025	0.02	0.01	0.005	0.0025	0.001	0.0005
Number of terms	837	713	598	471	306	212	187	136	98	77	57	44

From that data it is possible to identify time-dependent named-entity groups of words such as: general terms, person names, professions. Table 4 shows p-values of time-dependent variations of the person name *Hendrik*, typical professions, their absolute frequencies and p-values.

Table 4. Example of time dependent names and professions and their p-values

Word	p-value	Freq	Word	Translation	p-value	Freq
Henrick	0.0002	4821	Arbeider	*Worker*	0.0000	2404
Hendricx	0.0003	1123	Bouwmans	*Builders*	0.0000	557
Henricks	0.0254	1023	Raaijmaker	*Wheelmaker*	0.0147	636
Hendricus	0.0488	3848	Biggelaar	-	0.0102	1071

We see that the official form of the name *Hendrik* has time-dependent variations such as: *Henrick, Hendricx, Henricks, Hendricus*, etc.

5 EL-Framework for Text Classification

General Framework. We have already seen that historical data contains time dependencies. We aim to improve the classification results by combining several models that are trained on different time partitions. The classification task can be done by any appropriate model. The idea is described in the pseudo-code from Algorithm 1.

The original data set is split into two parts: training set $\mathcal{D}$ and test set $\mathcal{R}$. A set of identified time periods is denoted by $\mathcal{T}$. For every $\mathcal{T}_i$ in $\mathcal{T}$ (line 1) the algorithm constructs corresponding subsets $\mathcal{D}^{'} \in \mathcal{D}$ such that $\{d_i \in \mathcal{D}^{'} | date(d_i) \in \mathcal{T}_i\}$ with the corresponding target categories $\mathcal{C}^{'} \in \mathcal{C}$ such that$\{d_i \in \mathcal{D}^{'}$, $c_i \in \mathcal{C}^{'} | category(d_i) = c_i$ and $\mathcal{R}^{'} \in \mathcal{R}$ such that $\{r_i \in \mathcal{R}^{'} | date(r_i) \in \mathcal{T}_i\}$ (lines 2-4). On the next step (line 5) we learn a prediction model $\mathcal{M}_i$ for the time partition $\mathcal{T}_i$ on the identified training subset: $(\mathcal{D}^{'}, \mathcal{C}^{'})$ that has only the documents from partition $\mathcal{T}_i$. We use a model M_i to predict a category only for the documents from the same time partition t_i (line 6). As a result we have a number of models $\{\mathcal{M}_1, \ldots \mathcal{M}_n\}$, one model for each time period. We choose a model depending on the date of a document that we need to classify.

Input: Training set $\mathcal{D} = \{d_1, ..., d_n\}$ with category-labels $\{c_1, ..., c_k\}$.
Test set $\mathcal{R} = \{r_1, ..., r_h\}$.
Set of categories $\mathcal{C} = \{c_1, ..., c_k\}$ and set of time periods $\mathcal{T}$.
Output: Predicted labels $\mathcal{N}$ for all test instances $\mathcal{R}$

1: **for** each time period $\mathcal{T}_i$ in $\mathcal{T}$ **do**
2: $\mathcal{D}^{'} \in \mathcal{D}$: $\{d_i \in \mathcal{D}^{'} | date(d_i) \in \mathcal{T}_i\}$
3: $\mathcal{C}^{'} \in \mathcal{C}$: $\{c_i \in \mathcal{C}^{'}, d_i \in \mathcal{D}^{'} | category(d_i) = c_i\}$
4: $\mathcal{R}^{'} \in \mathcal{R}$: $\{r_i \in \mathcal{R}^{'} | date(r_i) \in \mathcal{T}_i\}$
5: $\mathcal{M}_i \leftarrow TrainModel(\mathcal{D}^{'}, \mathcal{C}^{'})$ # Learn a model on a specific time period
6: $\mathcal{N}_i \leftarrow Classify(\mathcal{R}^{'}, \mathcal{M}_i)$ # Classify data
7: $\mathcal{N} \leftarrow \mathcal{N} \cup \mathcal{N}_i$
8: **end for**
9: **return** $\mathcal{N}$

Algorithm 1. EL-framework

Optimal Time Frame Identification. One of the benefits of the described approach is that it can be used as a framework of any text classification algorithm. It requires already predefined time periods. In Sect. 4 we identified time periods based on historical events. However, historical events give arbitrary time periods and may correspond to linguistic changes in the language. Another approach is to cluster years.

In the first step we merge all of the documents from the same year together. As a result we have one large single document per each year. Then we construct a feature vector using the TF-IDF vectorizer as described in Sect. 3. TF-IDF feature extraction is more appropriate for this task than term-frequency because

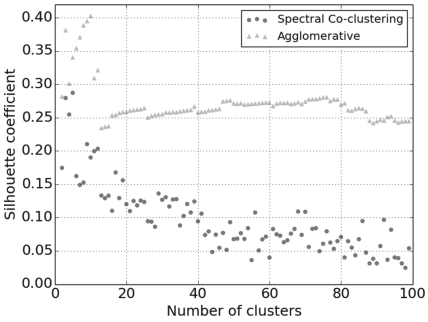


Fig. 3. Silhouette coefficient

it assigns a higher weight to infrequent words. After that we apply clustering. In this paper we compare two clustering techniques: *Spectral Co-clustering algorithm* [5] and *Agglomerative Hierarchical Cluster* [17]. Before apply clustering technique we remove from the original dataset all numbers, years, category names and non-alphabetical characters. This step is necessary in order to avoid biases in clustering.

6 Experiments and Results

We evaluate the designed EL-framework with Rijksmuseum time division and EL-framework with time frames according to year clustering and compared the results to two baselines. As the first baseline, we use the standard text classification method and as the second baseline, we use a sliding window (+/- decades). In the second case a classifier is trained on a decade before and a decade after a classifying year. We apply 10-fold cross-validation when training and testing examples belong to the same data partition. We evaluated the performance of the applied algorithms in standard metrics such as: overall accuracy and the macro-average indicators (precision, recall and f-score).

Cluster Evaluation. In order to evaluate the year clustering technique and find an appropriate number of clusters we compute the Silhouette coefficient [2] and vary the number of clusters from 2 to 100. The Silhouette coefficient is an unsupervised clustering evaluation measure when the ground truth labels are unknown. The higher the Silhouette coefficient is, the better clusters are found. Figure 3 shows the Silhouette coefficient of Spectral Co-clustering and Agglomerative Hierarchical Cluster for different number of clusters. We use *cosine similarity* to calculate the intra-cluster distance and nearest-cluster distance for each sample. We see that the Silhouette coefficient achieves the maximum value when the number of clusters $k = 5$ for Spectral Co-clustering and the number of clusters $k = 10$ for Agglomerative Hierarchical Clustering.

We consider the number of clusters $\{5, 7, 10\}$ for experiments. Number of clusters equals to 5 or 10 yields the maximum Silhouette coefficient, number

of clusters equal to 7 corresponds to the number of identified main historical events.

Figure 4 shows year partitioning according to the two described clustering approaches. Most of the clusters have relatively homogeneous structure without forcing temporal cluster constrains. Years from early periods and recent periods never occur in the same cluster. It shows that the clustering is not random and confirms the existence of linguistic changes over time. In early periods the structure is less homogeneous, because of the lack of documentation standards. However, we clearly see the clusters starting from the beginning of 18^{th} century and from 1811 onwards.

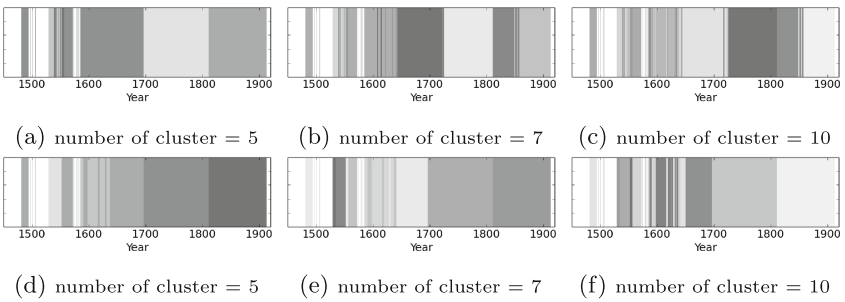


Fig. 4. Comparison of different year clusters: (a)-(c) after applying the Spectral Co-clustering algorithm [5], (d)-(f) after applying Agglomerative Hierarchical Cluster algorithm. The white space on the graph indicates that there are no documents in some years.

Cluster Analysis. We apply the χ^2 statistic to analyse the reasons of cluster homogeneity. All of the visualised clusters are similar, therefore we use in this experiment Spectral Co-clustering with the number of clusters equal to 7. Table 5 presents the number of terms and their corresponding p-value.

The number of terms that are cluster dependent is much higher than the number of terms that are dependent on arbitrary time partitioning, compare Table 5 with Table 3 respectively. It means that the current time partitioning is more optimal. There are different groups of terms with low p-values. Among of them occur general terms including verbs, professions, names, etc. For instance, words such as: *gulden (guilder), schepenen (alderman), beroep (profession), pastoor (pastor), burgemeester (mayor), goederen (goeds), verklaren (to declare)* have a large correlation with the clusters.

6.1 EL-Framework Evaluation

We compare the results of EL framework with two other approaches: standard text classification method and a sliding window decade based approach (see

Table 5. Cluster analysis. Number of terms bounded by p-value

p-value	0.25	0.2	0.15	0.1	0.05	0.025	0.02	0.01	0.005	0.0025	0.001	0.0005
Number of terms	9083	8619	8079	7476	6698	6144	6000	5581	5248	4961	4631	4423

Table 6). The EL-framework demonstrates an improvement in the main evaluation metrics. Overall, the classification accuracy increases almost 1 %, the three macro-average indicators (precision, recall and f-score) increase up to 2 %. The standard approach and sliding window that we take as a baseline already produces very high results, that is why it is very difficult to achieve contrasting difference. Improving the results from 90 % to 91 % means that we remove 10 % of the errors. It is easier to improve 10 % if the performance is only, for instance, around 40 % than when the performance is already 90 %. The EL-framework achieves the maximum improvements using Spectral Co-clustering for year partitioning with the number of clusters equal to 7. We exclude years and class labels to make clustering.

We see a large difference in the performance between overall accuracy and macro-average indicators in all experiments. The original dataset is not balanced: 20 % of the data belongs to the largest category and there are several very small categories that do not have enough examples for training the classifier.

Table 6. Overall accuracy and macro average indicators. TC stands for text classification

	Overall accuracy	Precision	Recall	f-score
Baseline 1: Standard TC	90.01 %	73.93 %	54.84 %	0.6297
Baseline 2: Sliding window	89.53 %	74.89 %	53.64 %	0.6238
EL + Spectral, $k = 5$	90.67 %	**75.87**%	55.90 %	0.6437
EL + Spectral, $k = 7$	**90.83**%	74.45 %	55.71 %	0.6373
EL + Spectral, $k = 10$	90.69 %	74.62 %	55.43 %	0.6361
EL + Aggl., $k = 5$	90.67 %	75.83 %	55.83 %	0.6431
EL + Aggl., $k = 7$	90.65 %	75.65 %	**56.03**%	**0.6438**
EL + Aggl., $k = 10$	90.59 %	75.80 %	55.81 %	0.6429

We also evaluate the performance of applied techniques per every as an average per century as it shown on Fig. 5. The standard text classification uses more training examples, however it never achieves the maximum performance compared to the EL-framework with an optimal time partitioning. The difference in performance between the EL-framework and the standard technique is positive in many centuries but the former depends on the selected time partitioning strategy.

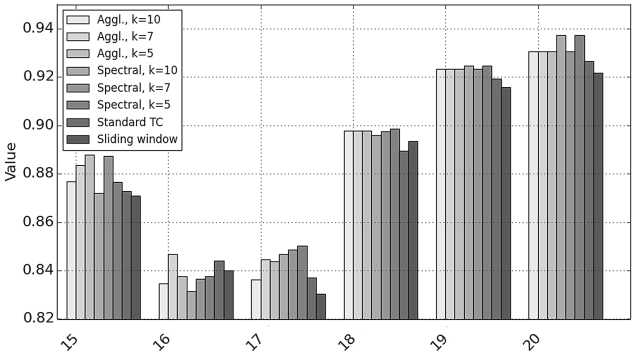


Fig. 5. Overall accuracy averaged per century that corresponds to EL-framework with different time partitioning and standard text classification.

The number of documents per year also varies a lot and the number of documents in some periods is less than in others. In many cases the available amount of training data is sufficient to make a high quality prediction within a time period. However we leave the identification of optimal size constrained time periods to future work.

7 Conclusions and Future Work

In this paper, we demonstrated temporal characteristics of the data applied to a collection of historical notary acts. We analysed dependency between time periods and correlated terms, class distributions and sampling effect. Then we designed a framework to incorporate temporal dependencies into the overall text classification process. We used main historical events to determine time periods. Moreover, we applied clustering techniques in order to obtain optimal time partitions automatically. This is a novel view of the text classification problem which demonstrated improvements in the results.

The presented empirical study of the temporal data aspects and the designed EL-framework make a significant contribution into the text classification area.

Acknowledgments. Mining Social Structures from Genealogical Data (project no. 640.005.003) project, part of the CATCH program funded by the Netherlands Organization for Scientific Research (NWO).

References

1. Almeida, T.A., Almeida, J., Yamakami, A.: Spam filtering: how the dimensionality reduction affects the accuracy of naive bayes classifiers. J. Internet Serv. Appl. **1**(3), 183–200 (2011)

2. Aranganayagi, S., Thangavel, K.: Clustering categorical data using silhouette coefficient as a relocating measure. In: Proceedings of the International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007), vol. 2 pp. 13–17, IEEE Computer Society, Washington, DC (2007)
3. Cardie, C.: Empirical methods in information extraction. AI Mag. **18**, 65–79 (1997)
4. Dalli, A., Wilks, Y.: Automatic dating of documents and temporal text classification. In: Proceedings of the Workshop on Annotating and Reasoning About Time and Events. ARTE 2006, pp. 17–22. Association for Computational Linguistics (2006)
5. Dhillon, I.S.: Co-clustering documents and words using bipartite spectral graph partitioning. In: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD 2001, pp. 269–274. ACM (2001)
6. Efremova, J., Montes García, A., Calders, T.: Classification of historical notary acts with noisy labels. In: Hanbury, A., Kazai, G., Rauber, A., Fuhr, N. (eds.) ECIR 2015. LNCS, vol. 9022, pp. 49–54. Springer, Heidelberg (2015)
7. Efremova, J., Ranjbar-Sahraei, B., Calders, T.: A hybrid disambiguation measure for inaccurate cultural heritage data. In: The 8th Workshop on LaTeCH, pp. 47–55 (2014)
8. Ikonomakis, M., Kotsiantis, S., Tampakas, V.: Text classification using machine learning techniques (2005)
9. Leong, C.K., Lee, Y.H., Mak, W.K.: Mining sentiments in sms texts for teaching evaluation. Expert Syst. Appl. **39**(3), 2584–2589 (2012)
10. Mihalcea, R., Nastase, V.: Word epoch disambiguation: finding how words change over time. In: ACL (2), pp. 259–263. The Association for Computer Linguistics (2012)
11. Mourão, F., Rocha, L., Araújo, R., Couto, T., Gonçalves, M., Meira Jr., W.: Understanding temporal aspects in document classification. In: Proceedings of the 2008 International Conference on Web Search and Data Mining. WSDM 2008, pp. 159–170. ACM, USA (2008)
12. Pearson, K.: On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that can be reasonably supposed to have arisen from random sampling. Phil. Mag. **50**, 157–175 (1900)
13. Pedregosa, F., et al.: Scikit-learn: machine learning in python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
14. Salles, T., da Rocha, L.C., Mourão, F., Pappa, G.L., Cunha, L., Gonçalves Jr, M.A., Wrigley Jr, W.: Automatic document classification temporally robust. JIDM **1**(2), 199–212 (2010)
15. Sammut, C., Webb, G.I.: Encyclopedia of Machine Learning. Springer, Berlin Heidelberg (2010)
16. Sebastiani, F.: Machine learning in automated text categorization. ACM Comput. Surv. **34**(1), 1–47 (2002)
17. Zhao, Y., Karypis, G., Fayyad, U.: Hierarchical clustering algorithms for document datasets. Data Min. Knowl. Discov. **10**, 141–168 (2005)