Low-Resource Techniques for Analysing the Rhetorical Structure of Swedish Historical Petitions

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

Natural language processing techniques can be valuable for improving and facilitating historical research. This is also true for the analysis of petitions, a source which has been relatively little used in historical research. However, limited data resources pose challenges for mainstream natural language processing approaches based on machine learning. In this paper, we explore methods for automatically segmenting petitions according to their rhetorical structure. We find that the use of rules, word embeddings, and especially keywords can give promising results for this task.

1 Introduction

Digitisation of historical sources supports the aim of preserving cultural heritage, makes the sources more accessible, and enables the use of computational techniques to study them further. Yet, the often limited data resources pose challenges for standard natural language processing methods based on machine learning. In this paper, we study the problem of automatically segmenting a specific type of historical text — petitions — using natural language processing techniques, while dealing with extreme data limitations.

Petitions have been a means for ordinary people in many pre-modern and pre-democratic societies to seek assistance from those in power. Petitions were typically directed towards a social or economic superior, such as a court of law, parliament, landlord, or even the monarch (Houston, 2014). As a valuable historical source, petitions could shed light on the daily lives of common people in the past. Despite this, petitions have rarely been used in historical research. Therefore, we are participating in an interdisciplinary research project, initiated at Uppsala University and funded by the Swedish Research Council, with the aim of increasing access to and knowledge of Swedish 18th century petitions.[1] The goal of the project is to use this source of information to gain insights into how people supported themselves and asserted their rights in the past. The project is entitled *Speaking to One's Superiors: Petitions as Cultural Heritage and Sources of Knowledge*, and is led by the *Gender and Work* (GaW) research project at the Department of History, Uppsala University. The GaW project examines how women and men provided for themselves in the Early Modern Swedish society (approximately 1550–1880). As part of the project, thousands of historical sources have been collected, classified, and stored in a unique database, which is now accessible to researchers, students, and the general public.

The structure of petitions were in many parts of Europe based on a classical rhetorical division. With some variations, most petitions seem to be comprised of five sections: *Salutatio*, *Exordium*, *Narratio* (including *Argumentatio*), *Petitio*, and *Conclusio* (Dodd, 2011; Israelsson, 2016). This paper delves into the use of computational approaches to segment petitions by automatic means, with the main goal of facilitating and enhancing the task of information extraction for historians and other scholars interested in studying petitions. We here target two specified sections of the petitions: the introduction (Salutatio) and the ending (Conclusio). By categorising different sections of the petitions, this segmentation approach could help researchers target only the relevant portions and search for information more efficiently, as the type of information varies in the different parts of the petitions. However, due to significant limitations in available data resources, careful considerations are needed in the development of such ap-

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

Proceedings of the Second Workshop on Resources and Representations for Under-Resourced Languages and Domains (RESOURCEFUL-2023), pages 132–139, May 22, 2023

proaches, where our toolbox includes a combination of rules, keywords and word embeddings.

Previous work related to text or rhetorical structure segmentation for modern text often make use of large pre-trained language models, like in Lukasik et al. (2020), or the creation of a domain specific corpus, for example as a list of predefined Rhetorical Roles applied to legal documents (Kalamkar et al., 2022). However, no previous work has, to the best of our knowledge, focused on the specific task of automatically segmenting petitions, with the challenges of working with historical text and (in our case) a very limited set of available data resources.

2 The Petition Data Sets

In our project, we make use of a transcribed collection of 18th century petitions submitted to the regional administration in Örebro, Sweden, divided into two subsets dating from 1719 and 1782, respectively. To test the generalisability of our methods, we also include a small additional set of petitions from another region in Sweden, since there might exist some regional differences in how petitions were constructed. The latter data set, which we from now on describe as "out-of-domain" (OOD), is a small collection of manually transcribed petitions from Västmanland, Sweden. All data sets, together with some statistics, are described in Table 1. The text pre-processing procedures, including normalisation and part-of-speech (POS) tagging, are described in Section 3.

As stated in the introduction, petitions were often written in five or six sections (Dodd, 2011; Israelsson, 2016): (i) Salutatio: formal greeting to the addressee; (ii) *Exordium*: introduction of the petitioner(s); (iii) *Narratio*: narration of the circumstances leading to the petition (often mixed with arguments (iv) *Argumentatio*); (v) *Petitio*: presentation of the request; and (vi) *Conclusio*: final part, with ending and plea.

We are interested in testing computational methods for segmenting the petitions automatically, with the main purpose of facilitating and enhancing the task of information extraction for historians and other scholars. As a start, we focus on the introduction (Salutatio) and the ending (Conclusio), since these parts mostly consist of greetings and courtesy markers, and would typically not be of much interest for the information extraction task. Thus, being able to automatically remove these parts before analysing the text, would mean less noise in the information extraction process. We divide the Örebro data set into a training set (70%) and a test set (30%). In the training set, 10 petitions have been manually annotated by a historian (including all of the petition parts described above). We use this part of the training set as a validation set when needed during the development of our methods. The OOD data set is included in the test set only. The test set of our collection was manually segmented (only the Salutatio and Conclusio parts) by two of the authors independently. The segmentation showed a high inter-annotator agreement with 36/41 petitions being segmented identically. The 5 remaining cases showed minor discrepancies, which were resolved after discussion.

3 Methodology: Finding Salutatio and Conclusio

Working with historical data poses challenges for computational methods due to the often limited data resources. In addition to this, historical data is often noisy with variations in orthography, which makes pre-processing important. Below we first describe the different pre-processing steps used and then move on to the different segmentation methods explored, as well as the evaluation procedure.

3.1 Pre-Processing Techniques

In historical text, inconsistent spelling and spelling different from the standard spelling of contemporary text, can negatively impact the performance of natural language processing techniques, since these tools are normally trained on contemporary language. Therefore, spelling normalisation is an important pre-processing step for many natural language processing tasks when applied to historical text. In the spelling normalisation process, the spelling in the historical text is automatically transformed to a more standard (typically more modern) spelling, before the natural language processing tools are applied. Our data set is normalised using the SMT-based approach of Pettersson et al. (2014), which is available as an online tool.[2] To the best of our knowledge, no method yet has substantially outperformed a character-based SMT approach for historical Swedish. However, we also want to apply our methods to an un-

[2]https://cl.lingfil.uu.se/histcorp/tools.html

Petitions	Period	# Docs	Train/Test	# Tokens
Örebro earlier	1719–1720	51	70/30	16,285
Örebro later	1782–1800	60	70/30	15,814
Västmanland	1758	7	0/100	2,187
All	1719–1800	118	66/34	34,286

Table 1: Overview of the data sets with information about time period, number of documents, proportions of training and test data, and number of tokens (in the un-normalised data set).

normalised version of our data set, to see how much is gained when using pre-processing methods (since tools might not always be available for the intended data, especially with limitations in data resources).

Historical texts may not only have spelling variation. Punctuation may also be used in inconsistent ways, or not used at all. This is also true for the petition data sets. It is not uncommon that a petition from our data set contains segments with more than 100 successive tokens without any sentence boundary marker (in the traditional sense), and in the Örebro dataset, there are around 20 such segments that exceed a length of 250 tokens. Instead, the text usually contains phrase boundaries marked with a comma, semicolon, colon or a new line. For this reason, we perform our own sentence segmentation using a regular expression, where the characters `[.!?,:;\n]` are treated as marking sentence boundaries. This approach leads to shortening of excessively long sentences, although it may also result in some sentences being reduced to very short phrases.

In one of our approaches (see Section 3.3), we utilise part-of-speech (POS) tags. Here, we obtain the annotation from Språkbanken's Sparv annotation pipeline version 4.1.1 (Borin et al., 2016), which uses the Stanza tagger (Qi et al., 2020) for POS tagging, trained on SUC3[3] with Talbanken_SBX_dev[4] as development set.

In two of our approaches (see Section 3.4 and 3.5), we make use of the Swedish pre-trained word embeddings (WE) by Hengchen and Tahmasebi (2021), trained on historical Swedish newspaper material. We try both their Word2vec and fastText models,[5] using the incrementally 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.

3.2 Baseline Method: Cut by Length

As a start, we establish a baseline method for extracting the Salutatio and Conclusio from a petition by simply extracting the first and last sections of the text, respectively. To determine the length of these sections, we analyse the validation set and calculate the average number of sentences for each part. We here define a "sentence" as a chunk of text between two separators, as explained in the previous section. Based on our validation set, the average length of the Salutatio is 4 sentences, while the Conclusio is 10 sentences.

To extract the Salutatio in new documents, the baseline method simply extracts the first 4 sentences returned by the sentence segmenter. The process is repeated in reverse order to extract the Conclusio, starting at the end of the petition and extracting the last 10 sentences. Once the Salutatio and Conclusio have been extracted in this way, we remove any leading or trailing white space before returning the final results.

3.3 Rule-based Method

Our first real method for extracting the Salutatio and Conclusio uses a set of simple rules. To establish these rules, we analyse the petitions in our training set and identify common patterns. Our observations lead us to identify a set of typical words that frequently appear in the Salutatio and Conclusio parts of the petitions. We also conclude that both the beginning and (especially) the end of the petitions have short sentences or phrases with only names of people and of geographical places, often as a part of a greeting or a formal farewell. While the initial idea was to include a rule to capture sentences that contain proper names, we inspected the POS tagging in the training files and

[3]https://spraakbanken.gu.se/en/resources/suc3
[4]https://spraakbanken.gu.se/resurser/talbanken
[5]https://zenodo.org/record/4301658 (June, 2022)

concluded that the tagging result often was inaccurate for names, which were tagged as nouns, adjectives or other parts of speech. To still capture these phrases, we instead implement a rule that targets sentences without a verb, which is often also true for these short phrases.

For the initial section, which is likely to be (part of) the Salutatio, we find the words *högvälborne* (high-born), *kunglig* (royal), *landshövding* (governor), *herre* (lord), *nådige* (gracious), *riddare* (knight), *orden* (order), and *baron* (baron), as well as their various spelling variations in both the normalised petitions and the raw dataset. Similarly, for the final section, we find the words *tjänare/ tjänarinna* (servant, masculine and feminine), *vördnad* (homage), *nådes* (grace), *ödmjukaste* (most humble), *djupaste* (deepest), and *undersåte* (subject).

To extract the Salutatio and Conclusio, we iterate through the initial and final sentences of the petition, respectively. We include only those sentences that meet at least one of two criteria: either they lack verbs or they contain at least one of the frequently used words identified in our analysis. To determine the presence of verbs, we use the POS-tagging method described in Section 3. To capture sentences that contain the identified highly frequent words, we employ a regular expression that detects the words and various common spelling variations of them.

3.4 Keyword Method

For our second method, we make use of statistically defined keywords. These keywords are calculated for all the different petition parts of the validation set, including not only Salutatio and Conclusio, but also Exordium, Narratio, Argumentatio and Petitio, when available. We also make use of the Swedish historical word embeddings, introduced in Section 3.1.

We use two methods to obtain the keywords. For the first one, we take the top 10 keywords, by calculating and ranking TF-IDF scores of all tokens, through scikit-learn's implementations (Pedregosa et al., 2011). We also expand the keyword list by finding the (up to) 10 most similar words to each of the keywords by the word2vec word embeddings, and the (up to) 10 most similar words to each of the keywords by the fastText word embeddings by the `most_similar` function from Gensim library (Rehurek and Sojka, 2011). The full list of keywords based on TF-IDF scores combined with word embeddings can therefore be up to 100 words. However, not all words exist in the historical Swedish word embeddings models, which lead to a shorter keyword list.

For the other keyword approach, we rank keywords based on their feature importance in a classification task. This is done in order to compare different approaches to obtain keywords for our very small data set. Here, we make use of the LinearSVC, also through scikit-learn. The LinearSVC has the attribute coef_attribute, which assigns weights to the features for each class versus all other classes (coefficients in the primal problem). To perform the classification task, we train the classification model on our validation set using the default settings. We collect the top 10 keywords of each approach in a separate list, and we expand the keyword list with similar words based on the historical word embeddings, as explained above. Each sentence of the petition receives a score per petition part, where one point is given per corresponding keyword it contains. The Salutatio part is extracted by looping through the sentences from the beginning of the petition, until it encounters a sentence that does not have Salutatio as a top candidate. The first sentence is by default treated as Salutatio. The same procedure is repeated for the Conclusio part, though instead starting the loop from the end of the petition, where the last ending sentence is treated as Conclusio by default.

3.5 Window Embedding Method

In our third approach, we again utilise the Swedish historical word embeddings, which were introduced in Section 3.1. To achieve this, we iterate through various chunk sizes of the petition text (measured by the number of tokens), and obtain a vector for each text chunk. We use the pre-trained word2vec and fastText embeddings to look up individual words, and then compute the average of all word embeddings for each text. Any out-of-vocabulary (OOV) words are assigned a plain zero embedding.

To compare the embeddings with a gold standard, we vectorise all the Salutatio and Conclusio segments from the validation set using the same approach. Each extracted and vectorised text chunk is compared with each of the vectorised Salutations or Conclusions, and we com-

pare their similarities by computing a similarity score for each comparison. All these comparisons are summed, and the text chunk which receives the highest similarity score is chosen as the winner. We do the same procedure for both Salutatio and Conclusio. If several text chunk candidates get the same top score, we choose the text chunk with the longest string.

We experiment with different window sizes of text chunks, which are retrieved by counting the number of tokens per Salutatio and Conclusio part in the validation set. We use the smallest window size as a starting point, and add one more token in each iteration until we reach the largest window size.

3.6 Evaluation Procedure

To evaluate our methods, we use precision and recall, where we compare the suggested Salutatio and Conclusio to their gold counterparts. We calculate these scores both at the word level and at the sentence level. In addition, we also look at how much the candidates differ from the gold standard in the start and the ending. We do this for each candidate by counting the number of tokens before or beyond the gold start token ($+$ or $-$), and the number of tokens before or beyond the gold end token ($+$ or $-$).

4 Results and Discussion

The results for our methods can be viewed in Table 2 (normalised data), Table 3 (un-normalised data) and Table 4 (OOD data). Some general trends can be spotted in the results for all of the data, even though the high standard deviation indicates that we must be cautious when interpreting the results.

For a start, we conclude that even though we work with extremely reduced resources in terms of data size (and noisy data on top of that), the results indicate that several of our approaches can be potentially useful for segmenting petitions. We see that the baseline method performs strongly when extracting the Salutatio part, which indicates that Salutatio is easier to correctly catch in comparison to Conclusio. This is not surprising, since the Conclusio contains a formal farewell and is often signed with names of the petitioner(s), and the number of these may differ. Since the names are often written with line separation, they are treated as several sentences by our sentence segmentation approach, which lowers the performance of the baseline method. And indeed, some of our methods are able to beat the baseline method for extracting the Conclusio part, though high standard deviations also suggest that the performance for Conclusio is more varied both between and within the methods. Overall, the Keyword Method generally performs well, and gets the most consistent high results for the in-domain data set.

Another observation is that the gain of using spelling normalisation varies between the methods. The Window Embedding approach performs considerably better at extracting Salutatio on the normalised data set, in comparison of their results for the un-normalised data set. Also the Rule-based Method is helped by including spelling normalisation, both when extracting Salutatio and when extracting Conclusio. However, the Keyword Method works well both on the normalised and un-normalised data set, and in some cases the results are even better when not including spelling normalisation.

To see how well our methods can generalise to unseen data, we look at the results for the OOD set. However, since there are very few data points in the OOD set, it is difficult to draw any certain conclusions, though we can spot some potential trends. The results for all methods are generally low when handling un-normalised data. Here, normalisation seems to help quite substantially. The Window Embedding approach gets high results for normalised Salutatio, while the performance of the Keyword Method drops significantly, especially for the task of extracting Conclusio. The baseline method works surprisingly well for Salutatio and gets the highest result for the un-normalised OOD data, suggesting that the length of Salutatio may be somewhat consistent between regional places. For extracting Conclusio, the Keword Method and the Rule-based method seem to perform the best, at least on the normalised data set. However, the performance for extracting Conclusio is low for all methods, indicating that regional differences may affect the composition of this petition part.

When performing a qualitative analysis of the extracted Salutatio and Conclusio parts, we can identify some areas of improvement for our methods. By inspecting the results obtained using the Rule Method, we found several phrases where errors in the POS tagging led to the incorrect inclusion or exclusion of these phrases. It is important to consider these findings when developing

Tested	Baseline	Rules	kw TF-IDF	kw SVC	Window w2v	Window ft
Sal mean prec (words)	93.2 ± 21.8	75.9 ± 30.3	**93.8** ± 17.1	84.8 ± 25.5	91.4 ± 12.5	93.3 ± 16.5
Sal mean rec (words)	98.3 ± 6.8	**100** ± 0.0	94.2 ± 18.6	98.1 ± 10.9	92.4 ± 17.3	96.9 ± 13.2
Sal mean prec (sents)	**91.4** ± 25.3	75.5 ± 33.5	89.4 ± 26.8	84.1 ± 27.8	65.7 ± 35.3	79.3 ± 36.2
Sal mean rec (sents)	90.4 ± 25.6	86.4 ± 30.8	89.9 ± 27.8	**91.9** ± 22.9	76.5 ± 37.1	83.8 ± 35.9
Sal mean diff start (words)	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0-0	0.0 ± 0.0	1.0 ± 2.2	6.6 ± 35.5
Sal mean diff end (words)	6.0 ± 28.6	11.9 ± 30.7	0.8 ± 5.2	4.3 ± 8.6	1.1 ± 3.1	7.1 ± 36.1
Con mean prec (words)	67.5 ± 32.6	90.5 ± 19.9	89.0 ± 20.3	**98.1** ± 7.7	42.4 ± 29.5	60.0 ± 34.4
Con mean rec (words)	**81.1** ± 29.0	68.8 ± 29.4	78.3 ± 27.3	55.7 ± 34.7	66.5 ± 35.6	68.6 ± 31.8
Con mean prec (sents)	78.0 ± 27.2	96.0 ± 8.3	94.8 ± 10.4	**99.1** ± 3.7	43.5 ± 34.3	52.4 ± 32.8
Con mean rec (sents)	82.0 ±26.6	73.4 ± 23.9	**83.7** ± 22.6	63.4 ± 31.1	53.0 ± 42.5	53.6 ± 35.9
Con mean diff start (words)	-15.8 ± 39.7	2.3 ± 34.5	2.8 ± 16.6	13.9 ± 17.0	-114.5 ± 159.7	-73.4 ± 140.7
Con mean diff end (words)	-7.2 ± 33.7	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	-94.0 ± 155.3	-65.8 ± 128.0

Table 2: For normalised data: mean and standard deviation of Precision and Recall, both on the word and sentence level. Also mean difference in start token and end token between extracted petition parts and the gold parts (measured in number of tokens). Sal = Salutatio. Con = Conclusio.

Tested	Baseline	Rules	kw TF-IDF	kw SVC	Window w2v	Window ft
Sal mean prec (words)	**93.1** ± 22.1	63.2 ± 29.2	89.0 ± 21.4	89.2 ± 21.2	72.9 ± 19.1	75.4 ± 19.6
Sal mean rec (words)	**98.2** ± 7.5	91.5 ± 8.4	85.9 ± 27.3	85.9 ± 27.3	80.2 ± 24.4	87.5 ± 13.1
Sal mean prec (sents)	**96.2** ± 12.5	45.2 ± 37.6	91.9 ± 16.0	91.1 ± 17.8	34.2 ± 33.7	39.6 ± 38.5
Sal mean rec (sents)	**98.2** ± 7.0	57.6 ± 40.6	87.6 ± 25.0	87.6 ± 25.0	43.4 ± 42.4	46.0 ± 42.8
Sal mean diff start (words)	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	4.5 ± 20.7	6.8 ± 35.4
Sal mean diff end (words)	6.03 ± 28.6	13.3 ± 31.3	0.73 ± 6.9	0.79 ± 7.2	4.7 ± 22.7	7.9 ± 36.2
Con mean prec (words)	64.3 ± 33.6	75.2 ± 25.7	92.7 ± 15.7	**98.4** ± 7.1	48.3 ± 37.6	33.9 ± 30.4
Con mean rec (words)	82.0 ± 29.0 ±	76.4 ± 26.8	**82.8** ± 27.3	56.2 ± 36.6	37.1 ± 28.8	45.3 ± 33.3
Con mean prec (sents)	75.2± 28.1	74.0 ± 24.2	96.4 ± 8.3	**99.3** ± 2.9	37.2 ± 35.6	27.1 ± 31.7
Con mean rec (sents)	82.6 ± 26.6	69.2 ± 26.2	**86.9** ± 22.6	65.1 ±32.9	31.2 ± 33.9	29.3 ± 35.1
Con mean diff start (words)	-20.2 ± 42.0	-17.6 ± 71.1	3.4 ± 14.6	14.4 ± 18.6	-74.4 ± 143.3	-143.3 ± 161.5
Con mean diff end (words)	-7.2 ± 33.7	-11.2 ± 63.1	0.0 ± 0.0	0.0 ± 0.0	-66.7 ± 122.3	-129.1 ± 155.2

Table 3: For un-normaliased data: mean and standard deviation of Precision and Recall, both on the word and sentence level. Also mean difference in start token and end token between extracted petition parts and the gold parts (measured in number of tokens). Sal = Salutatio. Con = Conclusio.

our methods further in future work. In the Keyword Method and Window Embedding Methods, we used WE as part of the approach. However, some of the tokens in our petition dataset were not present in the WE models. Further analysis is needed to determine the extent to which this affected the results for the different parts of the petition. Regarding the Window Embedding Method, we select the winning text chunk with the longest string when several candidates receive the same top score. Although this is the correct decision in some cases, it can decrease performance in other cases. This is particularly true when searching for the Conclusio part, where many of the extracted text chunks in the result are overly long. This method would likely benefit from a more elaborate ranking system when dealing with multiple winning text chunks.

5 Conclusions

In this paper, we try different approaches to automatically segment Swedish 18th-century petitions according to their rhetorical structure. More precisely, we extract the opening and ending parts of petitions: the Salutatio and the Conclusio.

Historical data is challenging for computational methods due to the noisy nature of non-standardised orthography, and due to limited data resources. Still, several of our methods are able to correctly identify the Salutatio and Conclusio parts, even though the precision and recall scores exhibit a high standard deviation. Our Keyword Method, which looks at each sentence and scores the number of keywords related to the target petition part, performed consistently high. However, even the baseline method, where we cut the pe-

Using normalised data						
Tested	**Baseline**	**Rules**	**kw TF-IDF**	**kw SVC**	**Window w2v**	**Window ft**
Sal mean prec (words)	77.9 ± 11.6	73.3 ± 19.4	73.6 ± 10.5	71.9 ± 12.6	**97.4** ± 2.9	95.0 ± 3.6
Sal mean rec (words)	78.5 ± 13.2	**100** ± 0.0	**100** ± 0.0	**100** ± 0.0	98.2 ± 2.7	**100** ± 0.0
Con mean prec (words)	24.7 ± 13.1	68.1 ± 35.2	68.5 ± 32.0	**84.9** ± 29.6	45.0 ± 28.4	37.4 ± 26.2
Con mean rec (words)	**100** ± 0.0	86.2 ± 33.7	84.5 ± 31.9	84.5 ± 31.9	77.9 ± 34.6	69.6 ± 38.4
Using raw (unnormalised) data						
Sal mean prec (words)	**67.2** ± 11.5	51.4 ± 14.3	19.6 ± 35.2	19.6 ± 35.2	63.0 ± 8.6	65.1 ± 10.3
Sal mean rec (words)	**86.6** ± 11.2	81.5 ± 5.6	7.1 ± 11.6	7.1 ± 11.6 ±	73.9 ± 5.9	81.5 ± 5.6
Con mean prec (words)	20.9 ± 12.0	51.2 ± 31.7	63.6 ± 33.7	**82.1** ± 34.6	8.2 ± 12.5	21.3 ± 18.5
Con mean rec (words)	**100** ± 0	78.3 ± 32.9	85.7 ± 35.0	85.7 ± 35.0	31.5± 41.4	48.8± 41.0

Table 4: Results when using out-of-domain (OOD) data: precision and recall for both normalised and un-normalised data (only for words, to save space). Sal = Salutatio. Con = Conclusio.

tition parts according to a defined length, works well for extracting Salutatio, suggesting that this part is easy to extract with more simpler methods.

The Conclusio part seems to have more variations in length and construction, and here our Keyword Method, and our Rule Based Method to some extent, outperforms the baseline. When it comes to text pre-processing, the gain of using spelling normalisation varies between the methods, suggesting that it may not always be a necessary step for tasks such as ours.

The results for a small out-of-domain data set are generally low when handling un-normalised data. Here, normalisation seems to help quite substantially. The baseline method gets the highest result for this data set, indicating that the length of Salutatio may be somewhat consistent between regional places. However, the performance for extracting Conclusio is low for all methods, indicating that regional differences may affect the composition of this petition part.

When inspecting the extracted Salutatio and Conclusio parts, we can detect some weaknesses in our methods that could be improved with further development. The Rule-Based method could be further refined by adding rules more specialised for the target petition part, and perhaps by also including an applicable Named Entity Recognition model to better target the names of people and of geographical places, which is relevant for Salutatio and Conclusio. In the Keyword Method, we could have made use of POS information to target only content words, as these might be more specific for each petition part. For the Window Embedding Method, we believe that it would be beneficial to further elaborate the ranking system when dealing with multiple winning text chunks. For future work, we are interested in applying the suggested improvements, as well as expanding the segmentation task to also include other petition parts. We also want to explore other available approaches. More data would be desirable in order to train a statistical model, yet we might explore other possibilities in the form of few-shot learning or similar methods. Even with restricted resources, we anticipate that we could increase our results even more with suitable approaches.

Acknowledgments

We are grateful for many fruitful discussions with the members of the Petitions and GaW projects: Jezzica Israelsson, Örjan Kardell, Jonas Lindström, Sofia Ling, Maria Ågren, Linda Oja and Fredrik Wahlberg. We also thank the anonymous reviewers for constructive comments. The research reported in this paper was supported by a grant from the Swedish Research Council (grant number 2018-06159).

References

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*.

Gwilym Dodd. 2011. Writing wrongs: the drafting of supplications to the crown in later fourteenth-century england. *Medium Aevum*, 80(2):217–246.

Simon Hengchen and Nina Tahmasebi. 2021. https://doi.org/10.5334/johd.22 A collection of Swedish diachronic word embedding models trained on historical newspaper data. *Journal of Open Humanities Data*, 7(2):1–7.

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

Jezzica Israelsson. 2016. In consideration of my meagre circumstances: The language of poverty as a tool for ordinary people in early modern sweden.

Prathamesh Kalamkar, Aman Tiwari, Astha Agarwal, Saurabh Karn, Smita Gupta, Vivek Raghavan, and Ashutosh Modi. 2022. Corpus for automatic structuring of legal documents. *arXiv preprint arXiv:2201.13125*.

Michal Lukasik, Boris Dadachev, Gonçalo Simoes, and Kishore Papineni. 2020. Text segmentation by cross segment attention. *arXiv preprint arXiv:2004.14535*.

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, Beáta Megyesi, and Joakim Nivre. 2014. 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.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. https://nlp.stanford.edu/pubs/qi2020stanza.pdf 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*.

Radim Rehurek and Petr Sojka. 2011. Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, 3(2).